



**Universitat Autònoma
de Barcelona**

Assistent de formularis Walnut

Memòria del projecte
d'Enginyeria Tècnica en
Informàtica de Gestió
realitzat per

Eduard Sciortino Poch

i dirigit per

Jordi Pons Aróztegui

Escola d'Enginyeria

Sabadell, juny de 2012

El sotasignat, **Jordi Pons Aróztegui**,
professor de l'Escola d'Enginyeria de la UAB,

CERTIFICA:

Que el treball al que correspon la present
memòria ha estat realitzat sota la seva direcció
per

Eduard Sciortino Poch

I per a que consti firma la present.
Sabadell, **juny** de **2012**

Signat: **Jordi Pons Aróztegui**

El sotasignat, **Ezequiel Parra Mestre**,
de **UNIT4 Business Software Ibérica**,

CERTIFICA:

Que el treball al que correspon la present
memòria ha estat realitzat sota la seva
supervisió per

Eduard Sciortino Poch

I per a que consti firma la present.

Sabadell, **juny** de **2012**

Signat: **Ezequiel Parra Mestre**

FULL DE RESUM – PROJECTE FI DE CARRERA DE L'ESCOLA D'ENGINYERIA

Títol del projecte: Assistent de formularis Walnut	
Autor: Eduard Sciortino Poch	Data: Juny 2012
Tutors: Jordi Pons Aróztegui (UAB) / Ezequiel Parra Mestre (Unit4)	
Titulació: Enginyeria Tècnica en Informàtica de Gestió	
Paraules clau (mínim 3) <ul style="list-style-type: none">• Català: formularis, walnut, karat• Castellà: formularios, walnut, karat• Anglès: forms, walnut, karat	
Resum del projecte (extensió màxima 100 paraules) <ul style="list-style-type: none">• <u>Català</u>: El projecte es desenvolupa en el marc d'un conveni entre la UAB i la empresa UNIT4 Business Software Ibérica en les oficines de Barberà del Vallès. L'objectiu del projecte consisteix en crear un assistent de formularis per a la plataforma de treball de l'aplicació de gestió de l'empresa UNIT4 que permeti tant a usuaris com a clients crear els seus propis formularis personalitzats complint sempre les directrius de disseny predefinides.• <u>Castellà</u>: El proyecto se desarrolla en el marco de un convenio entre la UAB y la empresa UNIT4 Business Software Ibérica en las oficinas de Barberà del Vallès. El objetivo del proyecto consiste en crear un asistente de formularios para la plataforma de trabajo de la aplicación de gestión de la empresa UNIT4 que permita tanto a usuarios como a clientes crear sus propios formularios personalizados cumpliendo siempre las directrices de diseño predefinidas.• <u>Anglès</u>: The project is developed under an agreement between the UAB and the company UNIT4 Business Software Ibérica in the offices of Barberà del Vallès. The goal of the project is create a form wizard for the work platform of the application of business management of the company UNIT4 that allows both users and customers to create their own customized forms fulfilling the predefined design guidelines.	

Taula de continguts

1. Introducció	1
1.1. Marc de treball	1
1.1.1. Conveni UNIT4 / UAB	1
1.1.2. L'empresa	1
1.2. Objectius del projecte	2
1.3. Breu introducció a karat i al model Walnut	2
1.3.1. Karat.....	2
1.3.2. Model Walnut.....	4
1.4. Contingut de la memòria.....	6
2. Estudi de viabilitat	9
2.1. Introducció	9
2.1.1. Descripció	9
2.1.2. Objectius del projecte	10
2.1.3. Parts interessades	11
2.1.4. Equip del projecte	11
2.2. Situació actual.....	12
2.2.1. Diagnòstic.....	16
2.3. Requisits del projecte	16
2.3.1. Requisits funcionals del projecte	16
2.3.2. Requisits no funcionals del projecte.....	17
2.3.3. Restriccions del sistema	17
2.3.4. Catalogació i prioritització dels requisits	18
2.4. Solució proposada.....	18
2.5. Planificació del projecte.....	19
2.5.1. Recursos del projecte.....	19
2.5.2. Planificació temporal	20
2.6. Pressupost.....	22
2.6.1. Estimació del cost del personal	22
2.6.2. Estimació del cost dels recursos.....	22
2.6.3. Resum i anàlisi del cost	22
2.7. Conclusions	23

3. Anàlisi de requisits	25
3.1. Perfils d'usuari.....	25
3.2. Requisits funcionals	25
3.3. Requisits no funcionals	27
4. Desenvolupament	29
4.1. Introducció	29
4.2. Eines utilitzades.....	29
4.2.1. Java	29
4.2.2. Eclipse.....	29
4.3. Interfície d'usuari.....	30
4.4. View's	32
4.4.1. Edit (TextBox)	33
4.4.2. Label.....	33
4.4.3. Check.....	34
4.4.4. Combo	34
4.4.5. Picture.....	35
4.4.6. Button.....	35
4.5. Containers	36
4.5.1. Grid	36
4.5.2. Card.....	37
4.5.3. Frame (Collapsible)	37
4.5.4. TreeView	38
4.6. Propietats.....	39
4.6.1. Nom del control	39
4.6.2. Etiqueta.....	40
4.6.3. Tamany	40
4.6.4. Read Only	42
4.6.5. Multilínia	43
4.6.6. Mode de visualització	43
4.7. Events	44
4.7.1. Menú click.....	44
4.7.2. Form Item Select	45
4.7.3. Form Item Validate.....	45
4.7.4. Form Post Load	46

4.7.5. View Click	46
4.7.6. View Validate.....	47
4.7.7. Container Drop	48
4.7.8. Container Create Node	49
4.7.9. Container Node Click.....	49
4.7.10. PreSemantic	50
4.7.11. Altres mètodes.....	51
5. Fase de proves	53
5.1. Control d'errors	53
5.2. Proves	56
5.2.1. Proves unitàries	56
5.2.2. Proves d'integració	57
6. Conclusions	59
6.1. Objectius assolits i no assolits	59
6.2. Desviació de la planificació	60
6.3. Possibles ampliacions i millores	60
6.4. Valoració personal.....	61
Bibliografia	63
Format escrit.....	63
Format Web	63
Contingut del CD	65

Índex de taules

Taula 1: Taula de classificació d'objectius.....	10
Taula 2: Parts interessades.....	11
Taula 3: Equip del projecte	12
Taula 4: Catalogació i prioritització dels requisits funcionals	18
Taula 5: Catalogació i prioritització dels requisits no funcionals.....	18
Taula 6: Recursos humans del projecte.....	19
Taula 7: Tasques del projecte	21
Taula 8: Costos estimats del personal.....	22
Taula 9: Costos estimats dels recursos	22
Taula 10: Resum i anàlisi del cost	22

Índex de figures

Figura 1: Arbre de navegació	4
Figura 2: Ruta de navegació	5
Figura 3: Formulari.....	6
Figura 4: Diagrama del desenvolupament de karat	9
Figura 5: Assistent de <i>karat Studio</i> (pas 1).....	12
Figura 6: Assistent de <i>karat Studio</i> (pas 2).....	13
Figura 7: Assistent de <i>karat Studio</i> (pas 3).....	13
Figura 8: Assistent de <i>karat Studio</i> (pas 4).....	14
Figura 9: Assistent de <i>karat Studio</i> (pas 5).....	14
Figura 10: Definidora de formularis	15
Figura 11: Diagrama de Gantt.....	21
Figura 12: Entorn de desenvolupament Eclipse	30
Figura 13: Proposta de disseny de la interfície gràfica	31
Figura 14: Proposta de disseny escollida	32
Figura 15: Diagrama de view's	32
Figura 16: Quadres de text	33

Figura 17: Etiqueta (Label)	33
Figura 18: Caixa de selecció.....	34
Figura 19: Llista desplegable.....	34
Figura 20: Imatge	35
Figura 21: Botons	35
Figura 22: Diagrama de Containers.....	36
Figura 23: Taula (grid).....	36
Figura 24: Pestanyes agrupades	37
Figura 25: Grups comprimibles.....	37
Figura 26: Grups comprimibles intercanviats	38
Figura 27: Grup comprimit	38
Figura 28: Vista en arbre.....	39
Figura 29: Vista sense control associat	39
Figura 30: Vista amb control associat.....	40
Figura 31: Etiqueta d'una vista.....	40
Figura 32: Diversos tipus d'ample	41
Figura 33: Alçada de les vistes	41
Figura 34: Coordenades top.....	41
Figura 35: Coordenada left	42
Figura 36: Composició de l'amplada d'una vista	42
Figura 37: Quadre de text amb propietat de només lectura	43
Figura 38: Quadre de text multilínia	43
Figura 39: Mode de visualització de les llistes desplegables	43
Figura 40: Menú pop-up corresponent a un grup comprimible.....	44
Figura 41: Menú pop-up corresponent a una pestanya	44
Figura 42: La selecció del tipus de vista activarà l'event	45
Figura 43: La modificació d'una propietat activa l'event.....	46
Figura 44: Funcionament de l'event	47
Figura 45: Validació d'un formulari ja existent	48
Figura 46: Element afegit mitjançant drag&drop amb posterior activació de l'event.....	48
Figura 47: Node creat que activa l'event	49
Figura 48: Cada node seleccionat mostra les seves corresponents propietats	50
Figura 49: Finestra de selecció d'objectes de negoci	50
Figura 50: Load corresponent als quadres de text.....	51

Figura 51: Load corresponent a les llistes desplegable	51
Figura 52: Load corresponent a un grup comprimible	51
Figura 53: Objecte de negoci construït en forma d'arbre	52
Figura 54: Estructura del formulari	52
Figura 55: Missatge d'error al crear un node incorrecte	53
Figura 56: Missatge d'error al borrar un node incorrecte	54
Figura 57: Missatge d'error al borrar un node amb fills	54
Figura 58: Missatge d'error al no tindre nom el formulari	54
Figura 59: Missatge d'error quan s'afegeix un node duplicat	55
Figura 60: Missatge d'error al obrir un formulari que no s'ha creat amb l'assistent	55
Figura 61: Missatge d'error quan no s'indica nom ni objecte de negoci	56

1. Introducció

1.1. Marc de treball

1.1.1. Conveni UNIT4 / UAB

Aquest projecte s'ha desenvolupat dins del marc d'un conveni entre la Universitat Autònoma de Barcelona i l'empresa UNIT4 Business Software Ibérica, en el qual, s'estipulen 560 hores per realitzar un treball de desenvolupament de software. Mitjançant aquest conveni, l'alumne pot adquirir coneixements de com funciona una empresa de desenvolupament de software de gestió i de tecnologies de la informació, a la vegada que aquest desenvolupament serveix com a treball de final de carrera.

1.1.2. L'empresa

UNIT4 té una gran experiència durant més de quaranta anys en el món de la enginyeria del software. Posseeix un ampli ventall de solucions de gestió d'última generació la qual la fa ser una de les primeres empreses espanyoles en tecnologies de la informació i comunicacions.

UNIT4 disposa actualment de 4.320 treballadors i compta amb oficines i distribuïdors en tot el món per garantir un accés fàcil i local a les ventes, serveis i suport. Es troben en: Alemanya, Austràlia, Bèlgica, Canadà, Dinamarca, Espanya, Estats Units, Estònia, França, Holanda, Hongria, Irlanda, Malàisia, Noruega, Portugal, Regne Unit, República Checa, Singapur, Sudàfrica, Suècia i Uganda.

UNIT4 desenvolupa i ofereix la gama de solucions ekon, tant ERP¹ com per la gestió del negoci, que s'adapten especialment a les organitzacions que desenvolupen la seva activitat en un clima de canvis freqüents i dinàmics (especialment les centrades en serveis o les del sector públic).

UNIT4 Ibérica aplica a les seves solucions la tecnologia *karat*, una completa

¹ Els sistemes de Planificació de Recursos Empresarials, (ERP) són Sistemes d'Informació Gerencials que integren i manegen molts dels negocis associats amb les operacions de producció i dels aspectes de distribució d'una companyia en la producció de béns o serveis.

plataforma tecnològica per la gestió de les empreses, que aporta un nou concepte de solucions basades en la independència total i real d'entorns. La seva característica principal és que permet l'execució del software desenvolupat en múltiples plataformes tecnològiques, sistemes operatius (MS Windows, Linux, MAC OS) i bases de dades sense modificar el codi font. A més a més, ofereix eines per la personalització de processos de negoci en les solucions, l'evolució de la qual queda garantida al mateix temps que l'estàndard.

Una de les últimes incorporacions a *karat* és Walnut, que es tracta de la nova interfície d'usuari. Amb el fi d'optimitzar el treball quotidià amb l'aplicació, centra la innovació en aprofitar l'experiència dels usuaris. A més a més és molt més dinàmic, personalitzable i potencia la operativitat.

Per els requisits del projecte, aquest es situa dins del departament de "Research & Development" que és el grup que s'encarrega del desenvolupament i el suport de l'eina *karat*. Aquest departament a més a més s'ocupa del desenvolupament de noves aplicacions i manteniment de les ja existents.

1.2. Objectius del projecte

El projecte té com a objectiu crear un assistent de formularis per a la plataforma *karat* que permeti, tant als propis usuaris de l'empresa com als clients que disposin d'una llicència de desenvolupament, construir els seus formularis personalitzats complint les directrius Walnut. En posteriors capítols s'explicarà detalladament en que consisteixen aquestes normes. L'assistent ha de ser també amigable i fàcil d'utilitzar.

A nivell personal espero obtenir experiència a nivell laboral en una gran empresa i en les seves metodologies de treball, a més a més de poder realitzar un projecte de principi a fi completant tots els seus cicles de desenvolupament.

1.3. Breu introducció a karat i al model Walnut

1.3.1. Karat

Amb la seva avançada plataforma tecnològica *karat*, UNIT4 ofereix un entorn de gestió integral del cicle de vida per a les aplicacions de gestió UNIT4 ekon,

concretament en temes de desenvolupament i instal·lació, suport, manteniment, evolució, personalització, etc.

És la plataforma tecnològica idònia per al desenvolupament, implantació, manteniment i evolució contínua d'aplicacions.

Karat garanteix la màxima independència respecte als elements tecnològics (bases de dades, sistemes operatius, terminals, etc.) i facilita un alt grau d'autonomia en l'automatització dels processos de negoci gràcies al seu complet ventall d'assistents, que simplifiquen a l'usuari tasques que abans realitzaven consultores.

La plataforma tecnològica *karat* ofereix la màxima agilitat d'ús després de la implementació, i a més permet que els canvis es produeixin de forma ràpida, instantània i gairebé sense cost. A sobre, facilita al màxim la personalització de la gestió i aconsegueix reduir a la mínima expressió els costos de propietat.

El resultat final són aplicacions de gestió totalment flexibles i amb una tecnologia tan avançada que poden personalitzar-se sense necessitat de tocar fonts. A més a més, permeten que els upgrades / actualitzacions es realitzin de forma immediata, amb totes les personalitzacions i pràcticament sense costos. *karat* també permet als usuaris la fàcil creació d'indicadors dinàmics (widgets) associats a nombrosos conceptes de negoci.

Aquests indicadors permeten, des d'un dispositiu mòbil o un ordinador, accedir instantàniament i amb un sol clic a una gran quantitat d'informació de UNIT4 ekon o internet, i generen avisos quan es produeixen certes situacions són fàcilment personalitzables i poden ser agrupats en quadres de comandament (workspaces) a la mida de cada usuari. D'aquesta manera s'optimitza la funcionalitat i aparença de les aplicacions.

Les principals característiques que *karat* ofereix als clients:

- ❖ Personalitzar la gestió per diferenciar-se de la competència
- ❖ Reduir els costos de propietat de les aplicacions
- ❖ Reduir els terminis d'adaptació
- ❖ Aprofitar-se al màxim dels avanços en les solucions estàndards
- ❖ Garantir l'evolució instantània de les personalitzacions, essencials per al negoci.
- ❖ Permetre la utilització de les aplicacions en qualsevol moment i en qualsevol lloc.

1.3.2. Model Walnut

El projecte Walnut parteix amb la intenció que la migració tecnològica de Visual Studio a Java suposi una millora clara per als clients de UNIT4 ekon, alhora que s'unifiquen criteris i s'homogeneïtza l'aspecte.

Cal tenir en compte que els objectius del projecte Walnut són:

- ❖ Respondre a les necessitats de negoci dels clients.
- ❖ Homogeneïtzar l'aspecte dels productes UNIT4 ekon amb una aparença atractiva.
- ❖ Optimitzar la usabilitat.
- ❖ Reduir els costos de desenvolupament i implantació.
- ❖ Comunicar internament i externament.

ESTRUCTURA GENERAL (ARBRE DE NAVEGACIÓ)

És l'apartat corresponent a la part esquerra de la finestra i pot ser amagat per augmentar l'espai de treball en l'àrea de formularis.

La funció de l'arbre de navegació consisteix en facilitar la recerca dels formularis pertanyents a cada producte que tingui associat tant el client com el propi usuari de *karat*.

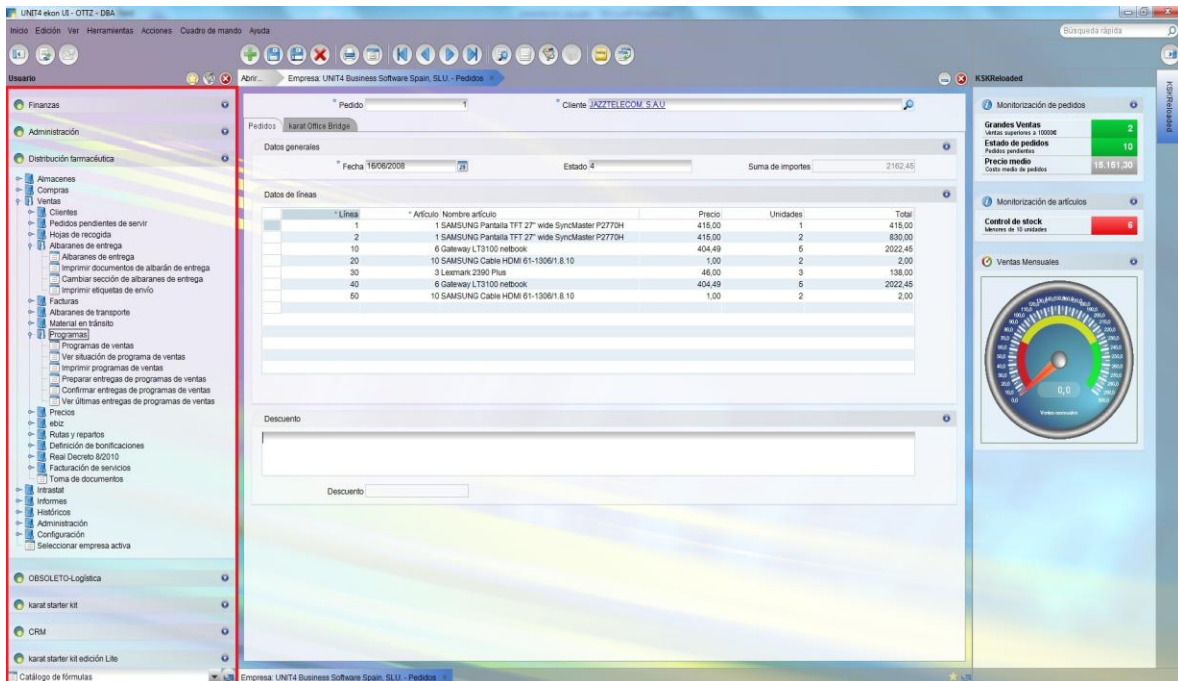


Figura 1: Arbre de navegació

ESTRUCTURA GENERAL (ÀREA DE FORMULARIS)

Ocupa l'espai central de les aplicacions i és l'àrea principal de treball dels usuaris. L'àrea de formularis té a la seva vegada tres parts fàcilment distingibles:

- ❖ **Ruta de navegació:** indica el flux que s'ha seguit navegant entre diferents formularis similar al dels navegadors web.

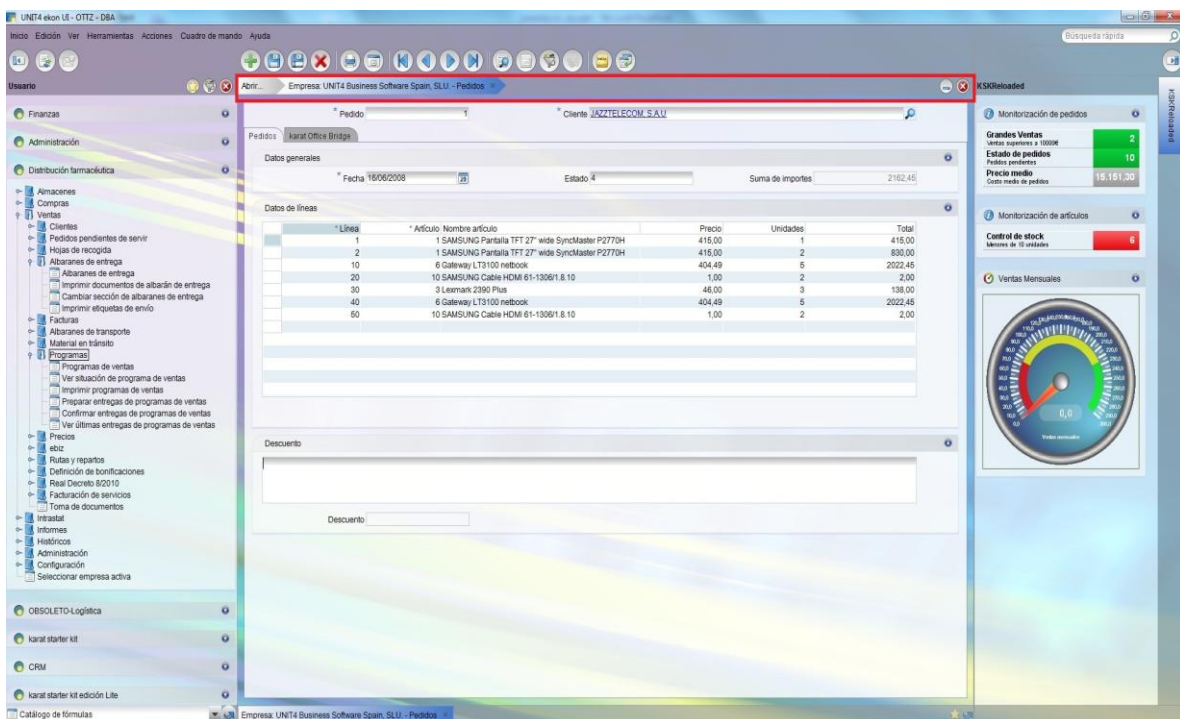


Figura 2: Ruta de navegació

- ❖ **Formulari:** ocupa el major espai d'aquesta àrea i és on es mostrarà gairebé tot el contingut corresponent al formulari. La part del formulari es divideix en les següents seccions:

- Part fixa: l'entitat principal té a la part superior un espai que conté els camps que permeten identificar de forma unívoca el registre i aquells que, per alguna raó, requereixen estar visibles en totes les pestanyes del formulari.
- Grup de pestanyes: acostuma a ser l'espai que ocupa un major

volum del formulari i està destinat a contenir grups de camps repartits en pestanyes. Aquests grups d'informació són comprimibles i es poden moure per a intercanviar la seva posició en la mateixa pestanya o traslladar-se a altres pestanyes.

- Botons d'entitat: conté els botons que realitzaran accions relacionades amb l'entitat. Aquest espai no és obligatori i únicament apareixerà si l'entitat té accions que l'afectin de forma global.

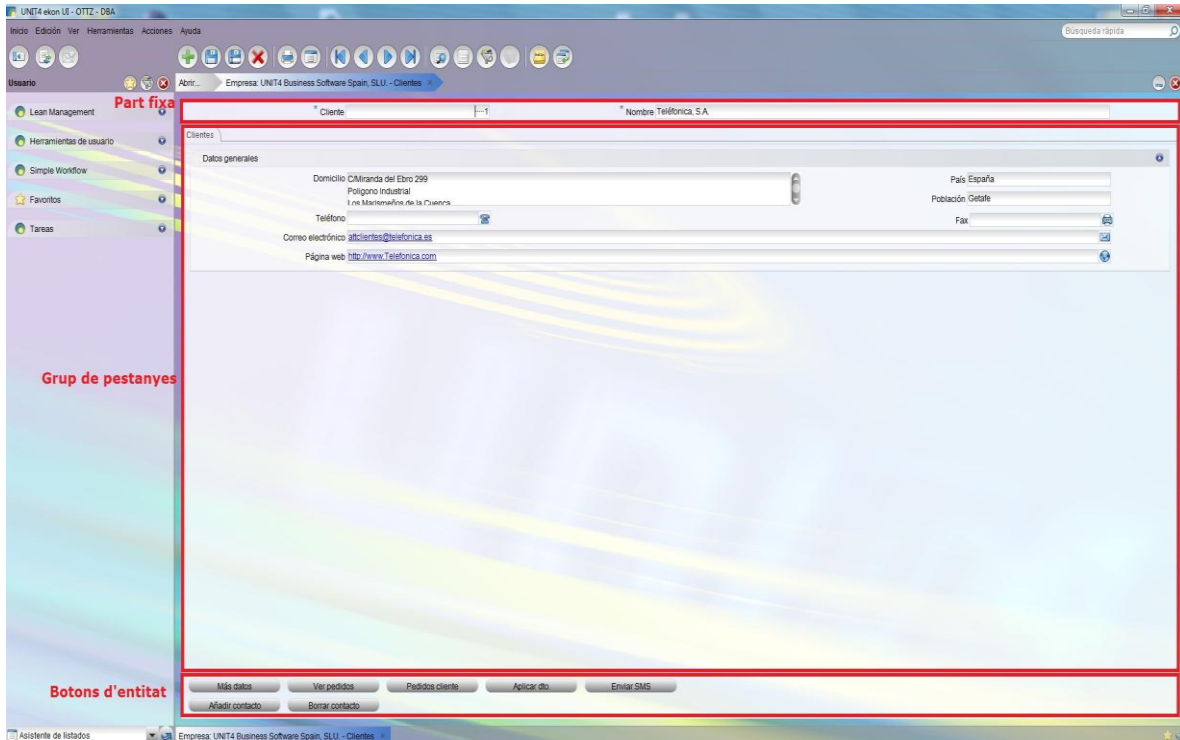


Figura 3: Formulari

1.4. Contingut de la memòria

A continuació, es descriuen els capítols dels quals consta la memòria del projecte:

- ❖ Introducció: Apartat on s'explica breument el marc d'ubicació del projecte, els objectius del projecte, una explicació curta i concisa de l'entorn on es desenvoluparà el projecte i conceptes a conèixer prèviament per tal d'entendre el funcionament de l'assistent en posteriors capítols.
- ❖ Estudi de viabilitat: En aquest capítol es veurà una descripció del sistema actual, una solució proposada per aquest i els beneficis que s'esperen

obtenir.

- ❖ Planificació del projecte: Es mostraran els processos que conformen el projecte i es detallarà com es gestionen i el temps i cost dedicat per a cada un.
- ❖ Anàlisi: Es farà un detallat anàlisi de les funcionalitats del projecte.
- ❖ Desenvolupament: Apartat on s'explicarà com s'han realitzat les funcionalitats comentades anteriorment d'una forma més detallada.
- ❖ Conclusions: Es redactaran els objectius assolits i possibles ampliacions i finalment es farà una valoració personal sobre el projecte.

2. Estudi de viabilitat

2.1. Introducció

Tot seguit s'explicarà de forma resumida el funcionament del desenvolupament de *karat* amb els elements que el componen i la seva funcionalitat.

2.1.1. Descripció

En la realització del nostre projecte ens centrarem en els conceptes d'Objecte de negoci i formularis, ja que no s'accedirà ni es farà cap manteniment sobre la base de dades.

L'Objecte de negoci *karat* és l'objecte que permet gestionar com una entitat única diversos nivells d'informació de la capa de dades. Aquest objecte ha de contenir tota la lògica de negoci de l'entitat que implementa, així com les regles de validació d'aquesta. Per a cada entitat de l'aplicació és necessària la definició d'un objecte de negoci ja que aquest serà el que ens permetrà fer el manteniment de les dades associades a l'entitat que representa.

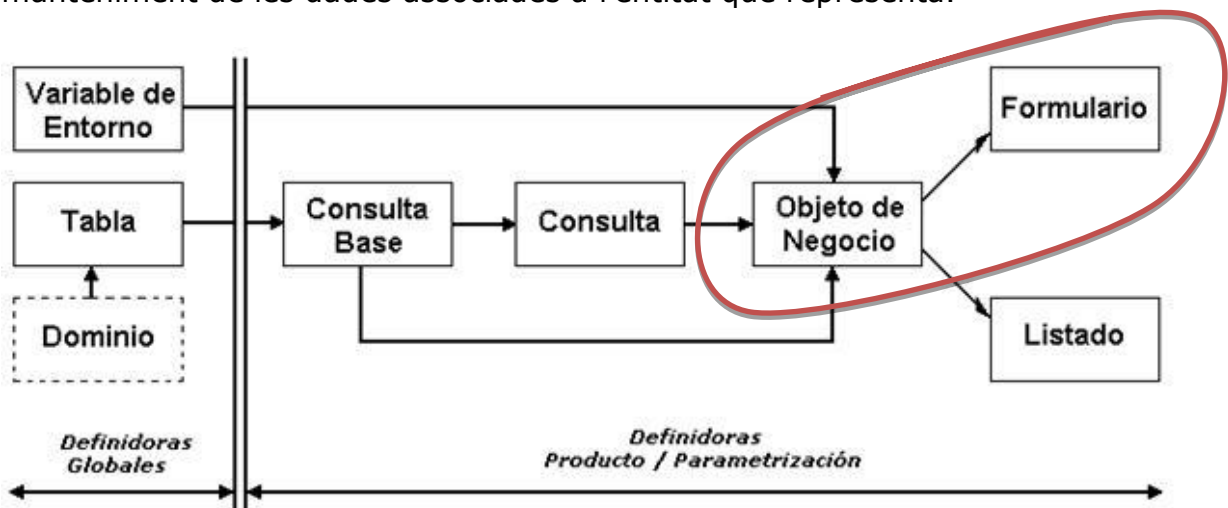


Figura 4: Diagrama del desenvolupament de karat

L'Objecte de negoci és l'objecte clau dins de l'entorn de desenvolupament *karat*, lloc que representa el nucli sobre el qual es basaran la resta d'objectes

com:

- Formularis.
- Llistats.
- Informes Navegables.
- Gestió Documental.
- Quadre de comandament.
- Transferència de Dades.
- Web Services.

La base del desenvolupament en *karat* es centra en l'Objecte de Negoci, on estan definides la major part de funcionalitats de l'aplicació.

2.1.2. Objectius del projecte

Els principals objectius proposats en el projecte, i la seva classificació segons el grau d'importància, són els següents:

1. Permetre la selecció de qualsevol objecte de negoci del producte.
2. Creació dels arbres d'objecte de negoci i formulari amb posterior intercanvi d'elements entre ells.
3. Assignació de propietats per a cada element seleccionat.
4. Generar el formulari.
5. Carregar l'estructura dels formularis prèviament creats.
6. Compliment de les directrius Walnut.
7. Ha de ser fàcil d'entendre i d'utilitzar amb una interfície atractiva per l'usuari.

	Crític	Prioritari	Secundari
Objectiu 1	X		
Objectiu 2	X		
Objectiu 3	X		
Objectiu 4	X		
Objectiu 5		X	
Objectiu 6	X		
Objectiu 7		X	

Taula 1: Taula de classificació d'objectius

2.1.3. Parts interessades

A continuació es mostren les parts interessades del projecte:

Nom	Descripció	Responsabilitat
Dept. "Research & Development" UNIT4	Departament de l'empresa UNIT4 dedicat a la part de desenvolupament i manteniment del software.	Departament encarregat de desenvolupar i mantenir nous productes de software.

Taula 2: Parts interessades

2.1.4. Equip del projecte

Tot seguit, en la següent taula podem veure la composició de l'equip del projecte amb una breu descripció de la responsabilitat de cada un.

Nom	Descripció	Responsabilitat
Jordi Pons Aróztegui	Tutor del projecte (TP)	Supervisa el treball de l'alumne durant el projecte.
Ezequiel Parra Mestre	Director del projecte (DP)	Defineix, gestiona i controla el projecte.
Eduard Sciortino Poch	Analista (A)	Desenvolupa l'estudi de viabilitat i la planificació. Anàlisi de l'aplicació: arquitectura, metodologia, especificació, estàndards... Participa en el disseny i validació.
Albert Bonet Carrera	Analista (A)	Assessorament en l'anàlisi de l'aplicació i assessorament tècnic al llarg del projecte.
Eduard Sciortino Poch	Programador (P)	Desenvolupa l'aplicació d'acord a l'anàlisi i la planificació prevista. Fa la implantació del projecte.
Eduard Sciortino Poch	Dissenyador (D)	Dissenya i desenvolupa l'aspecte de l'aplicació d'acord a l'anàlisi i

		normes de la usabilitat establertes.
Eduard Sciortino Poch	Tècnic de proves (T)	S'encarrega de dur a terme una sèrie de proves i tests de l'aplicació

Taula 3: Equip del projecte

2.2. Situació actual

Per a la creació de formularis encara no es disposa d'una eina adaptada en la nova versió de la plataforma *karat*. A causa d'això, s'utilitzen versions antigues en quant al desenvolupament de formularis. Més concretament, disposem de dues opcions a l'hora de construir el nostre propi formulari.

La primera opció és utilitzar l'assistent de l'entorn de desenvolupament *karat Studio* que consisteix en un assistent on ens anirà demanant pas a pas com volem que sigui el nostre formulari i quina informació volem que es mostri en ell a més del format d'aquest. L'inconvenient d'aquest sistema és que no deixa molta llibertat a l'usuari per organitzar el formulari al seu gust.

A continuació veurem els diferents passos de l'assistent i el tipus d'informació que necessita:

El primer pas de l'assistent és el corresponent al nom que li donarem al formulari junt amb una breu descripció d'aquest.

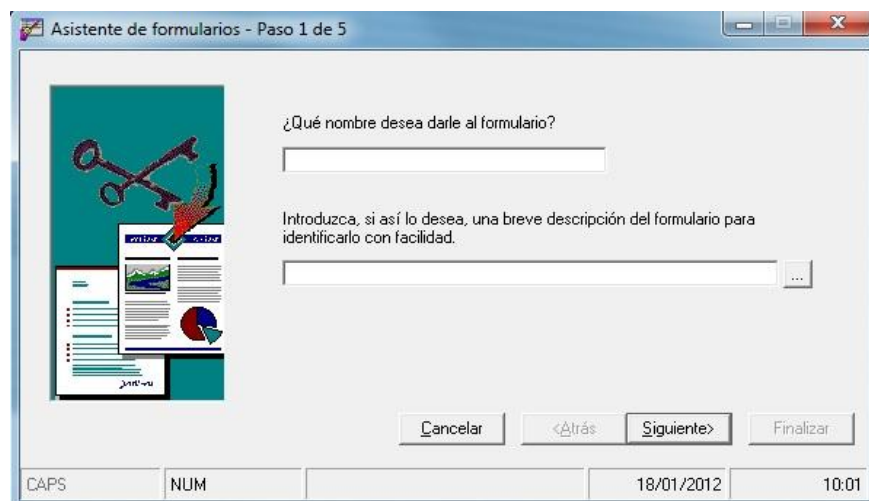


Figura 5: Assistent de *karat Studio* (pas 1)

Seguidament, el segon pas de l'assistent ens porta la primera restricció que és que no es poden escollir objectes de negoci amb més de tres nivells d'informació. Per exemple, un objecte de negoci de clients que pugui tenir un nivell de dades personals, dades de contacte, dades de pagament, etc. no es podria mostrar.

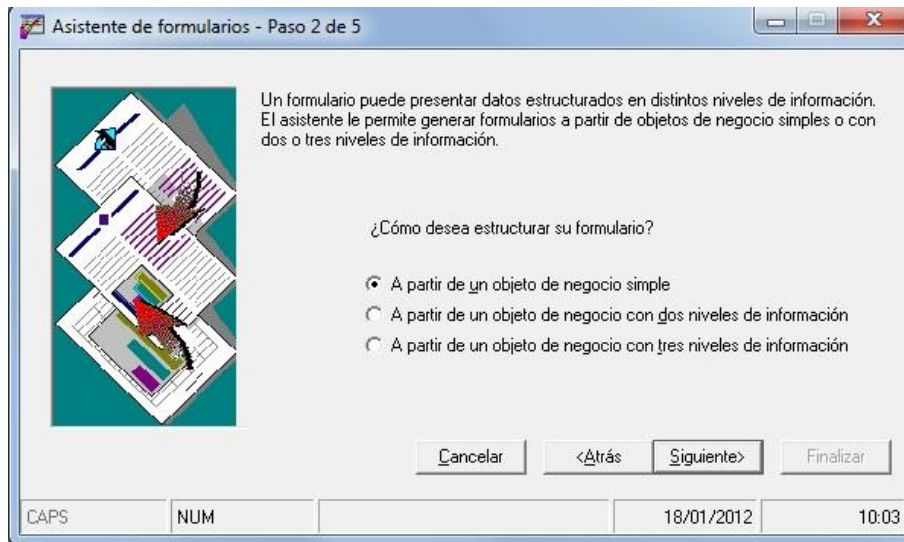


Figura 6: Assistent de karat Studio (pas 2)

Seguint els passos de l'assistent ens trobem amb el tercer pas que consisteix en l'elecció de l'objecte de negoci sobre el qual es basarà el nou formulari i si es vol associar a un altre servidor.



Figura 7: Assistent de karat Studio (pas 3)

Un cop s'arriba al pas quatre, trobem potser l'apartat més important de l'assistent ja que consisteix en la col·locació dels elements del formulari tal com es veuran un cop finalitzat aquest.

El mètode de col·locació funciona mitjançant files i columnes on a cada cel·la hi correspon un element.



Figura 8: Assistent de *karat Studio* (pas 4)

Per acabar, en l'últim pas ens pregunta si volem enllaçar amb la definidora de formularis per posteriors modificacions, que veurem més detalladament a continuació, i l'opció de finalitzar i crear el formulari.



Figura 9: Assistent de *karat Studio* (pas 5)

La segona opció per a la creació de formularis consisteix en la definidora de formularis, com s'ha esmentat anteriorment, que és una eina de creació de formularis molt més senzilla d'utilitzar i que permet un ampli ventall d'opcions gràcies a la seva interfície gràfica i al gran nombre de propietats modificables.

Bàsicament, consisteix en una finestra tipus on s'afegeixen els controls que hi ha dins de l'objecte de negoci el qual estigui associat al formulari. Cada finestra correspon a un grup comprimible de l'estructura Walnut ja sigui pertanyent a la part fixa o a una pestanya. Aquesta estructura és modificable afegint-hi més pestanyes i a la vegada més grups comprimibles. Per afegir elements a la finestra cal arrossegar-los i col·locar-los allà on l'usuari desitgi per, a continuació, atorgar-li les propietats més adients.

A més a més, es poden afegir altres tipus d'elements, ja siguin botons, gràfics, calendaris, taules, etc.

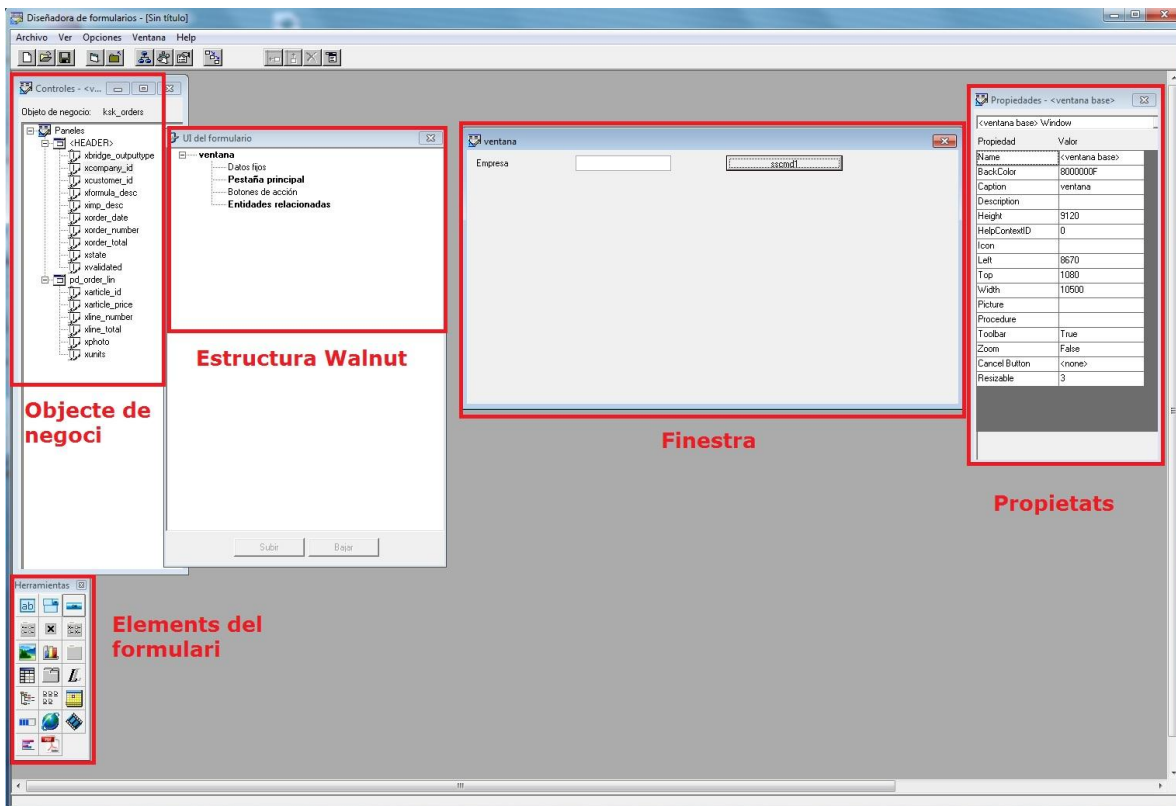


Figura 10: Definidora de formularis

2.2.1. Diagnòstic

A *priori*, pot semblar que no cal disposar d'un nou assistent ja que aquest compleix amb garanties tots els requisits que es podrien demanar dins d'un entorn de desenvolupament. Tot i això, l'inconvenient que té és que no es disposa d'un entorn de desenvolupament en la nova versió de *karat*, per tant, per a la creació de formularis encara es fa servir la definidora d'una versió antiga.

2.3. Requisits del projecte

2.3.1. Requisits funcionals del projecte

A continuació, es llisten tots els requisits funcionals que ha de complir el software que es crearà:

RF1. Permetre la selecció de qualsevol objecte de negoci del producte.

RF2. Posterior visualització dels segments i elements de l'objecte de negoci carregat prèviament en una vista en arbre.

RF3. Construcció de l'estructura del nou formulari mitjançant drag&drop² entre dues vistes en arbre.

RF4. Assignació de propietats per a cada tipus d'element seleccionat.

RF5. Creació d'elements més utilitzats com quadres de text (Edit), quadres desplegable (Combo), etiquetes (Label), imatges (Picture) i capsa de control (Check).

RF6. Creació de contenidors com taules (Grid), pestanyes (Card) i grups comprimibles (Collapsible).

RF7. Generar el formulari ja finalitzat.

² drag&drop és una expressió informàtica que es refereix a l'acció de moure amb el ratolí objectes d'una finestra a una altra o entre parts d'una mateixa finestra. Els objectes arrossegats són habitualment arxius, però també poden ser arrossegats altres tipus d'elements en funció del programa.

RF8. Carregar l'estructura dels formularis prèviament creats.

RF9. Visualitzar el formulari creat.

2.3.2. Requisits no funcionals del projecte

A més a més dels requisits funcionals, el projecte ha de contemplar els següents punts:

RNF1. Complir les directrius Walnut.

RNF2. Tolerància a errors i accions incorrectes.

RNF3. Disseny visual agradable i fàcil d'utilitzar per l'usuari.

RNF4. El software ha de permetre ser ampliable.

2.3.3. Restriccions del sistema

Entenem com a restricció de sistema tota aquella limitació la qual estem condicionats dins del nostre projecte com per exemple, màquines, llenguatge, plataformes de desenvolupament, en aquest projecte són les següents:

1. Seguir els estàndards de UNIT4.
2. L'aplicació s'ha d'adaptar al sistema de l'entitat.
3. El projecte s'ha de finalitzar abans del 30 de juny de 2012.

2.3.4. Catalogació i prioritització dels requisits

Requisits Funcionals	Essencial	Condicional	Opcional
RF1	X		
RF2	X		
RF3	X		
RF4	X		
RF5	X		
RF6		X	
RF7	X		
RF8		X	
RF9		X	

Taula 4: Catalogació i prioritització dels requisits funcionals

Requisits no Funcionals	Essencial	Condicional	Opcional
RNF1	X		
RNF2	X		
RNF3		X	
RNF4		X	

Taula 5: Catalogació i prioritització dels requisits no funcionals

2.4. Solució proposada

Per a l'execució del projecte s'ha tingut en compte que només hi ha una alternativa viable ja que la plataforma *karat* funciona amb el llenguatge de programació Java. Per això, dins d'aquesta alternativa es contemplen dues possibles opcions a escollir.

La primera consistiria en fer servir íntegrament el llenguatge Java i l'altre opció, que és l'escollida, seria utilitzar l'API³ pròpia de *karat* que contempla tots els mètodes implantats per a la creació del formulari.

³ és el conjunt de funcions i procediments (o mètodes, a la programació orientada a objectes) que ofereix certa biblioteca per ser utilitzat per un altre programari com una capa d'abstracció.

2.5. Planificació del projecte

2.5.1. Recursos del projecte

En aquest apartat es presenten tots els recursos referents al desenvolupament del nostre projecte.

El cost dels recursos humans és aproximat, ja que per aquest projecte l'alumne rebrà un import de 2.352€. El cost fictici del projecte suposaria al voltant d'uns 22.280€, com es detalla en l'apartat 2.6 del pressupost.

- Recursos humans:

Funció	Cost / h
Director del projecte	100€ /h
Analista	50€ /h
Programador	30€ /h
Tècnic de proves	20 €/h

Taula 6: Recursos humans del projecte

- Recursos materials:

- Llicència *karat*
- Equip:
 - PC - Intel Core 2Q8400 @ 2.66GHz
 - 6,00 Gb RAM
 - Disc dur de 500 Gb
 - Connexió a internet
 - Software
- Software:
 - Windows 7
 - Eclipse
 - Karat Studio
 - Karat Escriptori

2.5.2. Planificació temporal

El projecte es desenvoluparà entre el 14 de novembre de 2011 i el 26 de juny del 2012 amb una dedicació aproximada de 560h.

Aquest calendari pot variar ja que no s'han tingut en compte possibles dies festius o dies lliures agafats per l'alumne.

Les fases del projecte es desenvolupen de forma lineal, és a dir, no es comença una fins que no s'acabi l'anterior.

A continuació, es mostra una taula amb el llistat de les tasques a realitzar amb la seva duració estimada.

Tasca	Nom de la tasca	Duració	Predecessora
1	Assistent de formularis	162 dies	
2	Formació	10 dies	
3	Curs de formació	10 dies	
4	Planificació	15 dies	2
5	Estudi de viabilitat	3 dies	3
6	Aprovació de l'estudi de viabilitat	3 dies	5
7	Documentació de l'estudi de viabilitat	4 dies	6
8	Pla del projecte	2 dies	7
9	Aprovació del pla de projecte	1 dia	8
10	Documentació del pla de projecte	2 dies	9
11	Anàlisi de l'aplicació	11 dies	4
12	Anàlisi de requisits	4 dies	10
13	Estudi de la situació actual	4 dies	12
14	Aprovació de l'anàlisi	1 dia	13
15	Documentació de l'anàlisi	2 dies	14
16	Disseny de l'aplicació	18 dies	11
17	Curs de desenvolupament	5 dies	15
18	Disseny de la interfície gràfica	3 dies	17
19	Aprovació de la interfície gràfica	1 dia	18
20	Presentació inicial	9 dies	19
21	Desenvolupament de l'aplicació	63 dies	16
22	Documentació de l'API	2 dies	20
23	Preparació de l'entorn de desenvolupament	1 dia	22
24	Codificació	60 dies	23
25	Tests i proves	17 dies	21
26	Proves unitàries	7 dies	24
27	Tests	4 dies	26

28	Documentació de les proves	2 dies	27
29	Correcció d'errors	3 dies	28
30	Aprovació de l'aplicació	1 dia	29
31	Documentació	28 dies	25
32	Generar documentació	26 dies	30
33	Tancament del projecte	1 dia	32
34	Defensa del projecte	1 dia	33

Taula 7: Tasques del projecte

Tot seguit, es mostra el diagrama de Gantt de la planificació del projecte amb el número de tasca equivalent al de la taula anterior.

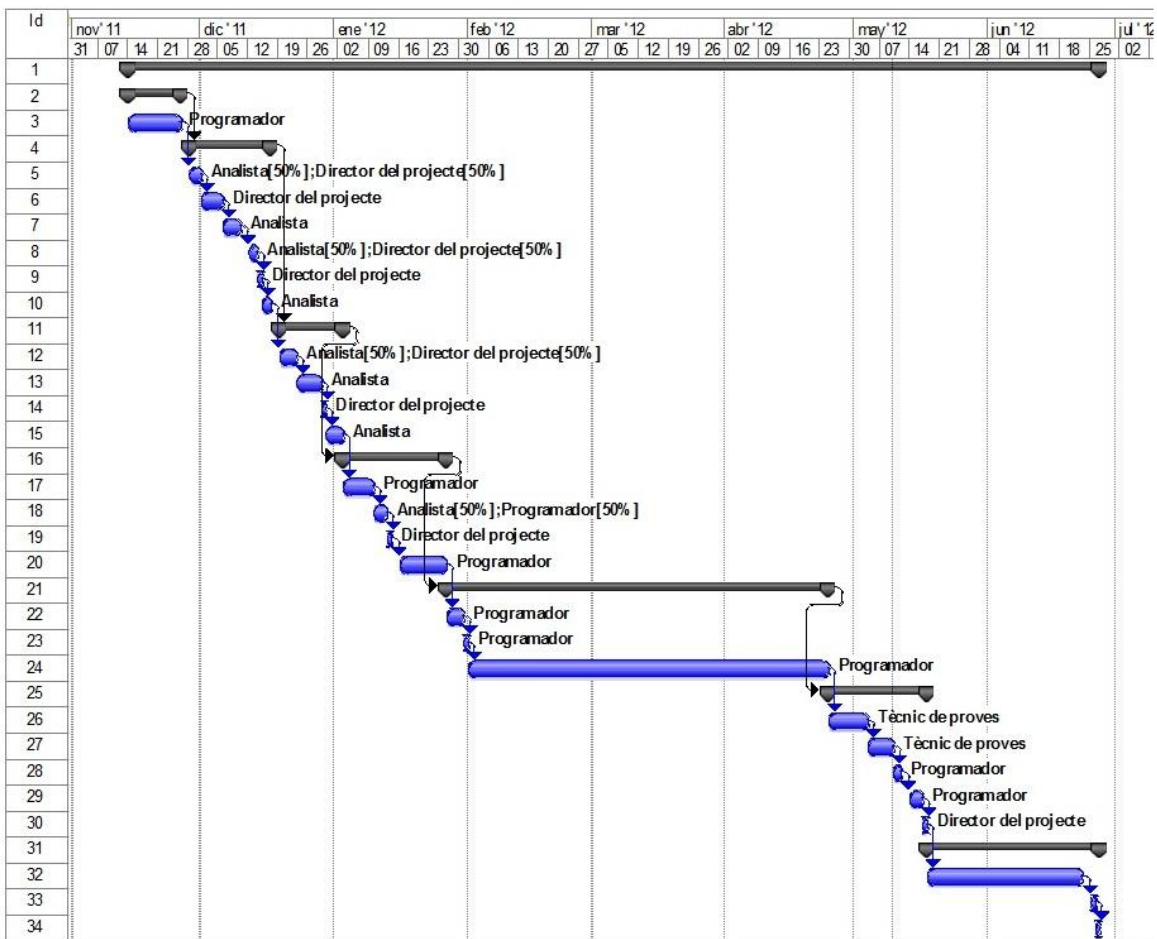


Figura 11: Diagrama de Gantt

2.6. Pressupost

2.6.1. Estimació del cost del personal

A continuació, es mostrarà una taula on es reflecteix el cost destinat al personal dels recursos del projecte.

Funció	Cost /h	Hores	Cost total
Director de projecte	100€ /h	46 h	4600 €
Analista	50€ /h	72 h	3600 €
Programador	30€ /h	382 h	11460 €
Tècnic de proves	20€ /h	60 h	1200 €
Total:		560 h	20860 €

Taula 8: Costos estimats del personal

2.6.2. Estimació del cost dels recursos

A continuació, es mostrarà una taula on es reflecteix el cost dels recursos destinats per realitzar el projecte.

Recurs	Cost
Ordinador personal	600 €

Taula 9: Costos estimats dels recursos

Els programes utilitzats per a la realització del projecte són propietat de UNIT4, fet que no comporta un cost addicional.

2.6.3. Resum i anàlisi del cost

Taula resum amb els costos totals del projecte.

Recurs	Cost
Cost del personal	20860 €
Cost de recursos	600 €
Total:	21460 €

Taula 10: Resum i anàlisi del cost

Cal tenir en compte que és un cost fictici ja que el cost real serà el que

l'alumne percebrà, un preu acordat de 2.352 € per la realització de tot el projecte.

2.7. Conclusions

Per concloure amb l'estudi de viabilitat es valoraran els beneficis que s'esperen obtenir i els inconvenients que puguin aparèixer en el projecte.

Pel que fa als beneficis:

- Adaptar una nova eina de creació de formularis per a l'última versió de la plataforma *karat*.
- Interfície amb l'estil Walnut.
- Disseny senzill per a l'usuari.
- Poder reeditar formularis.

El principal inconvenient del projecte és el següent:

- ❖ Que no compleixi les expectatives de l'usuari ja que la versió antiga ofereix moltes propietats que encara no estan implantades a més d'una interfície molt més visual.

Un cop analitzats els beneficis i inconvenients arribem a la conclusió que el projecte resulta totalment **viable**.

3. Anàlisi de requisits

3.1. Perfils d'usuari

A l'aplicació poden accedir tots els usuaris de *karat*:

PU. Usuari *karat*: Qualsevol usuari amb llicència *karat* pot accedir a aquesta aplicació i interactuar amb totes les seves funcionalitats.

3.2. Requisits funcionals

- ❖ RF1. Permetre la selecció de qualsevol objecte de negoci del producte: L'usuari pot tenir associat una sèrie de productes ja siguin finances, recursos humans i logística entre d'altres, i cadascun d'ells disposa d'objectes de negoci. El software ha de permetre el llistat i posterior elecció de tots els objectes de negoci sobre el qual estarà basat el formulari.
- ❖ RF2. Posterior visualització dels segments i elements de l'objecte de negoci carregat prèviament en una vista d'arbre: Un cop l'usuari decideixi quin serà l'objecte de negoci en el que es basarà el formulari, automàticament s'ha de carregar tota la seva estructura en una vista en arbre (TreeView). Un objecte de negoci pot estar dividit en diferents nivells d'informació, que se'n diuen panells (Segments) i cadascun d'ells té els elements (Items) que el conformen.
(En posteriors capítols s'explicarà més detalladament els tipus de vistes existents).
- ❖ RF3. Construcció de l'estructura del nou formulari mitjançant drag&drop entre dues vistes en arbres: Un cop carregada la informació de l'objecte de negoci en la vista en arbre s'ha de permetre a l'usuari, mitjançant un drag&drop entre dues vistes en arbre, la creació de l'estructura que tindrà el formulari complint sempre amb el format Walnut.
- ❖ RF4. Assignació de propietats per a cada tipus d'element seleccionat: El software ha de permetre distingir el tipus de vista que mostrarà l'element seleccionat, mostrar les diferents propietats que té cadascun i

la possibilitat de canviar els valors d'aquest. A més a més, a mesura que es vagin canviant els tipus de vista les diferents propietats s'han d'anar actualitzant automàticament.

- ❖ RF5. Creació d'elements més utilitzats com quadres de text (Edit), quadres desplegable (Combo), etiquetes (Label), imatges (Picture) i caps de control (Check): L'aplicació ha de permetre en una primera instància la creació dels elements més utilitzats en el desenvolupament de formularis. Aquests elements es crearan a partir de les propietats més imprescindibles que se'ls ha atorgat anteriorment.
- ❖ RF6. Creació de contenidors com taules (Grid), pestanyes (Card) i grups comprimibles (Collapsible): Per que es puguin mostrar tots els elements de forma organitzada i complir els requisits de les directrius Walnut s'han de poder crear contenidors que recullin els elements que s'han creat anteriorment. Aquests contenidors també requeriran d'unes propietats que s'han de poder modificar seguint el mateix funcionament que les vistes.
- ❖ RF7. Generar el formulari ja finalitzat: Un cop estigui creada l'estructura de contenidors del formulari i els elements assignats dins d'aquests contenidors l'últim pas en la creació del formulari que ha de permetre l'aplicació consisteix en guardar tota aquesta informació a la base de dades.
- ❖ RF8. Carregar l'estructura dels formularis prèviament creats: L'aplicació ha de permetre recuperar de la base de dades tota l'estructura que conforma el formulari i tornar a mostrar l'estructura en les vistes en arbre de l'objecte de negoci i dels contenidors i elements que tenia cadascun d'ells. A més a més, ha de mostrar les propietats que tenien tots els elements, permetre la seva modificació, actualitzar els nous valors afegits i posterior guardat a la base de dades.
- ❖ RF9. Visualitzar el formulari creat: Després d'haver guardat a la base de dades el formulari, l'assistent ha de permetre visualitzar-lo i comprovar la seva correcta funcionalitat.

3.3. Requisits no funcionals

- ❖ RNF1. Complir les directrius Walnut: S'ha de seguir estrictament la guia d'estil Walnut per al disseny dels formularis de l'aplicació, tant dels criteris visuals com dels interns que no són visibles per a l'usuari comú però sí per a usuaris amb certes llicències més avançades.
- ❖ RNF2. Tolerància a errors i accions incorrectes: L'aplicació haurà d'estar desenvolupada de manera que tindrà en compte que l'usuari que la utilitzi pot cometre errors durant l'entrada de dades, modificacions en les propietats i realitzar accions de forma diferent a la prevista. Així doncs, s' haurà de controlar les possibles entrades incorrectes permetent a l'usuari la correcció de les errades.
- ❖ RNF3. Disseny visual agradable i fàcil d'utilitzar per l'usuari: La interfície de l'assistent ha de ser comprensible per a l'usuari i ha de facilitar en tot moment el seu ús. Això és necessari perquè l'usuari no es vegi aclaparat per les eines de l'editor i pugui realitzar el seu treball d'una forma còmoda i senzilla.
- ❖ RNF4. El software ha de permetre ser ampliable: L'aplicació ha d'estar dissenyada correctament per permetre possibles ampliacions, concretament, d'elements del formulari que es puguin afegir que quedin pendents o afegir més propietats per a cada tipus de vista.

4. Desenvolupament

4.1. Introducció

En el capítol que es veurà a continuació, es detalla tota la fase de desenvolupament del projecte. En els primers punts veurem les plataformes tecnològiques utilitzades i les diverses propostes d'interfícies gràfiques. Els punts restants seran els més específics en quant al desenvolupament ja que s'explicaran com funcionen les diverses vistes i contenidors utilitzats i quina funció executen. Seguidament, les propietats necessàries per la visualització d'aquests elements i finalment tots els events que s'han fet servir amb la seva corresponent definició i funció detallada.

4.2. Eines utilitzades

Degut a que el software estarà integrat dins de la plataforma *karat* és necessari que el nostre software estigui completament codificat en Java. Més concretament, amb l'API de *karat* destinat al disseny de formularis.

4.2.1. Java

Java és un llenguatge de programació orientat a objectes, desenvolupat per Sun Microsystems a principis dels anys 90. El llenguatge en si mateix recull molta de la seva sintaxi de C i C++, però té un model d'objectes més simple i elimina eines de baix nivell, que solen induir a molts errors, com la manipulació directa de punters o memòria.

4.2.2. Eclipse

Per a la realització del software s'utilitza Eclipse. Eclipse és un dels IDE⁴ més

⁴ Un IDE és un entorn de programació que ha estat empaquetat com un programa d'aplicació, és a dir, consisteix en un editor de codi, un compilador, un depurador i un constructor d'interfície gràfica.

utilitzats a tot el món. Es tracta d'un programari amb llicència OpenSource⁵ i és totalment gratuït. Consta d'infinat d'eines per fer més còmode el desenvolupament d'aplicacions i la gestió de projectes.

Els motius de l'elecció d'aquest entorn es basa en que l'empresa UNIT4 ha adaptat un plug-in⁶ per facilitar el desenvolupament de les seves aplicacions.

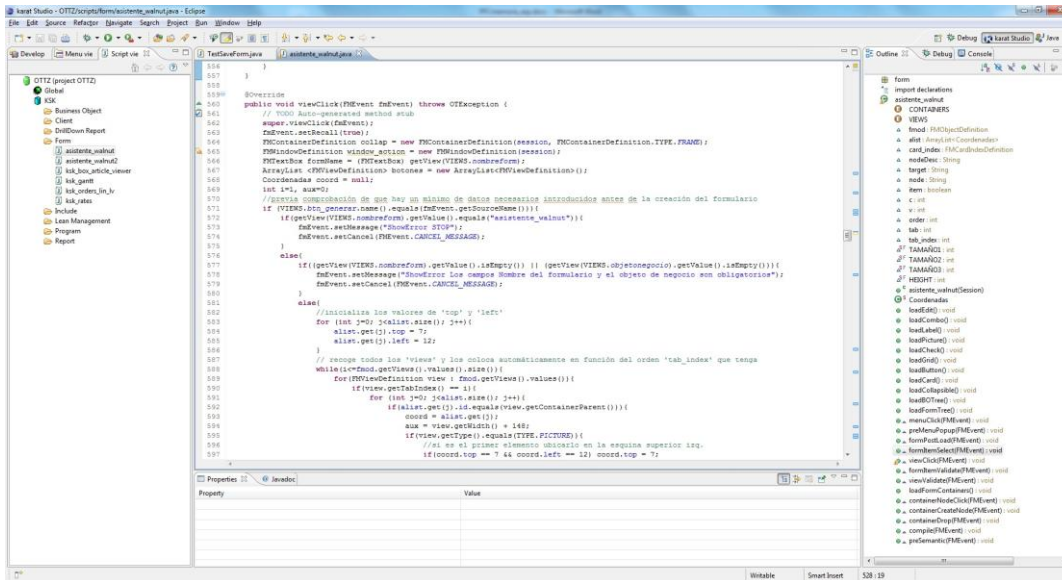


Figura 12: Entorn de desenvolupament Eclipse

4.3. Interfície d'usuari

Durant el disseny de la interfície s'han proposat diversos esbossos sobre com podria funcionar l'assistent. De totes les possibilitats s'ha optat per la que oferia un mètode més clar i senzill d'utilitzar i que a més a més s'adaptés a les necessitats de l'usuari.

A continuació veurem algunes d'aquestes propostes i finalment l'escollida amb els seus motius corresponents.

- ❖ **Proposta 1:** en una primera instància es va pensar en un assistent típic de diferents passos on a cada pas es demanaven les diferents parts que

⁵ és el terme amb el qual es coneix al software distribuït i desenvolupat lliurement.

⁶ Un plug-in és una aplicació que es relaciona amb una altra per aportar-li una funció nova i generalment molt específica.

necessita el formulari. Els primers passos estaven pensats per escollir l'objecte de negoci i les dades bàsiques, els següents passos pensats per les propietats del formulari. Per aquest pas es proposava una taula (Grid) on s'anirien afegint els elements, funcionament semblant al d'altres assistents de *karat*. Els principals inconvenients eren que als usuaris no els agradava aquesta forma de treballar i que cada vista té diferents tipus de propietats i diferent nombre, cosa que feia que el sistema no suportés aquesta opció.

- ❖ **Proposta 2:** La següent proposta que es va considerar, i que s'assimila més a la proposta escollida, consistia en agrupar-ho tot en una sola finestra tenint dos grups comprimibles, un amb dues vistes en arbre, una amb l'objecte de negoci i l'altre amb l'estructura del formulari on anar afegint dades amb drag&drop. L'altre grup comprimible tindria tot el conjunt de propietats pertanyents a cada tipus de control o vista. L'inconvenient d'aquest sistema és que el fet de tindre les propietats per separat incomodava la seva modificació.

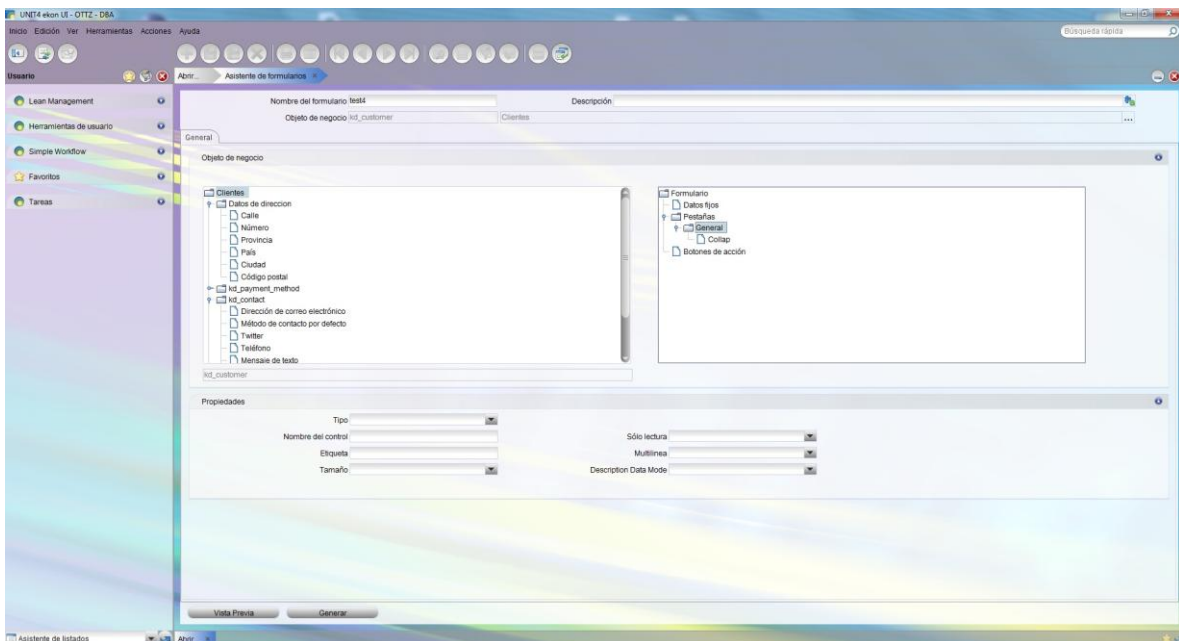


Figura 13: Proposta de disseny de la interfície gràfica

- ❖ **Proposta 3:** Finalment, la proposta escollida i que corregia el problema de l'anterior consistia en agrupar les dues vistes en arbre i les propietats en el mateix grup comprimible per tal de facilitar la seva modificació.

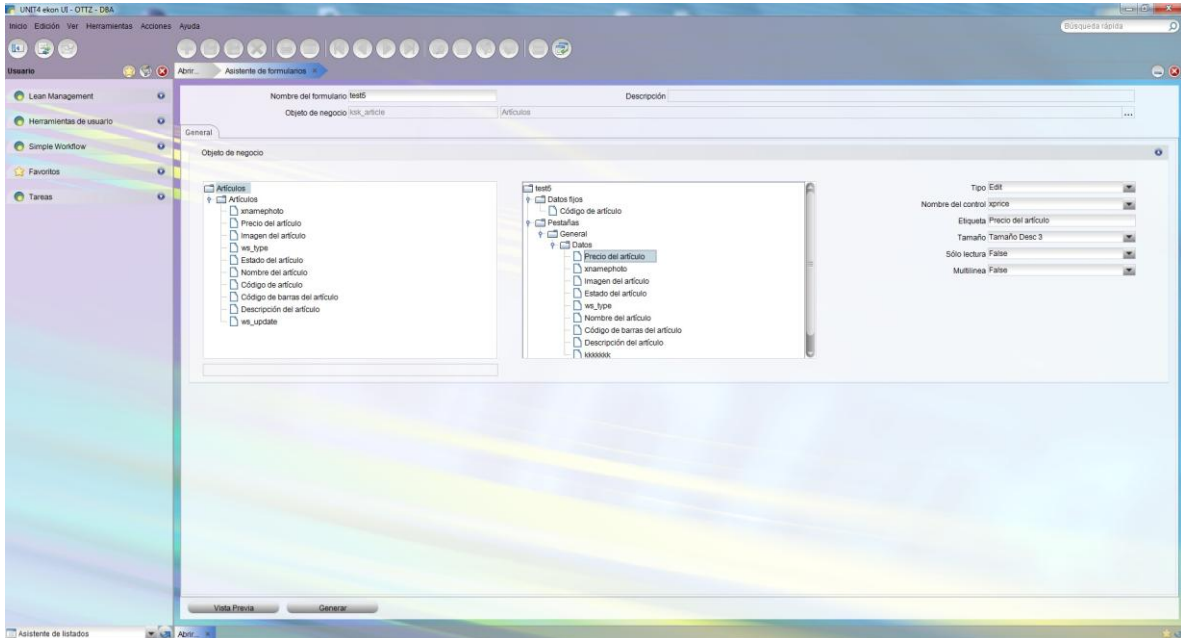


Figura 14: Proposta de disseny escollida

4.4. View's

Un view, o vista esmentat anteriorment, correspon a tots els elements pertanyents a un formulari que ens permetrà *a posteriori* visualitzar les dades que estiguin associades a la base de dades. De tots els tipus de vistes existents en el nostre projecte només se'n tracten les més utilitzades i comuns.

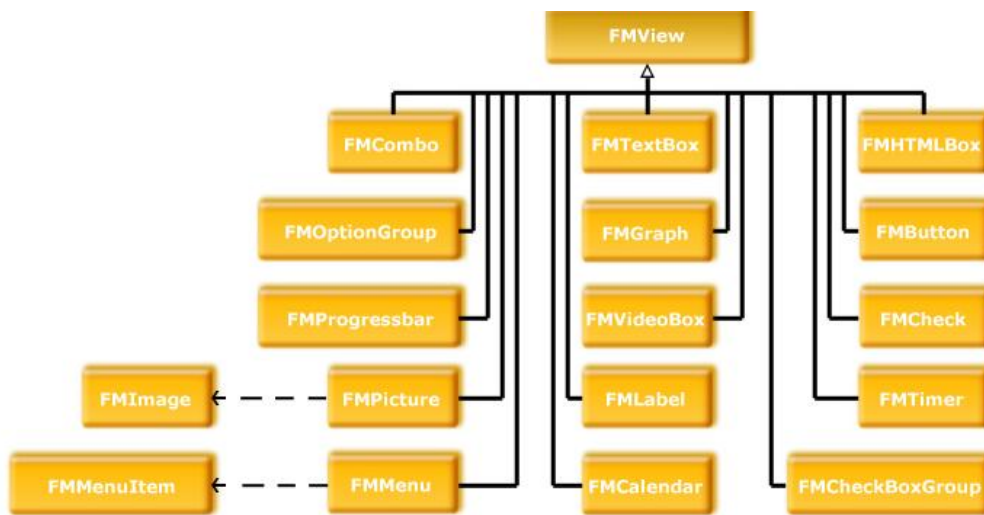


Figura 15: Diagrama de view's

4.4.1. Edit (TextBox)

La primera vista que veurem és la més utilitzada pels usuaris i consisteix en els quadres de text (Edit). Tal com indica el seu nom es tracta d'un objecte que no solament permet mostrar qualsevol dada de tipus alfanumèric sinó que a més té com a objectiu permetre als usuaris interactuar amb les dades modificant-les o esborrant-les. Pot associar-se a un control de l'objecte de negoci.

Figura 16: Quadres de text

Per poder crear les vistes calen unes propietats necessàries per a la seva visualització. A l'apartat 4.5 s'esmentaran les propietats per a tots els elements.

4.4.2. Label

El Label o etiqueta és l'objecte per excel·lència per incrustar textos no modificables per l'usuari final en temps d'execució. La seva principal funció no és altra que la de mostrar dades.

A continuació, es pot veure en la figura un exemple d'una etiqueta associada al control on mostra, en aquest cas, el nom d'un article sense possibilitat d'edició en funció de l'identificador que té a la base de dades.

Figura 17: Etiqueta (Label)

4.4.3. Check

És l'objecte de tipus caixa de selecció. Són independents i poden optar per dos valors: marcat o desmarcat.

En la figura següent hi ha un exemple del funcionament de la caixa de selecció que, en aquest cas, consta com a marcat el camp corresponent a empresa activa.

The screenshot shows a form with the following fields:

- * Nombre del cliente: Telefónica, S.A.
- * Calle: C/Miranda de Ebro
- * Número: 299
- * Provincia: Madrid
- * Ciudad: Getafe
- Empresa activa

 The 'Empresa activa' checkbox is highlighted with a red rectangle.

Figura 18: Caixa de selecció

4.4.4. Combo

S'utilitza per crear objectes de tipus llista desplegable de dades. Normalment s'associen a un control lligat a una llista de sistema o al resultat d'una consulta dependent (MDQO) ⁷.

Tot seguit, es pot veure un exemple on es carrega en una llista desplegable tots els elements pertanyents al resultat d'una MDQO.

The screenshot shows a dropdown menu with the following items:

- Nombre del control: xcustomer_name
- xsms
- xweb
- xcontact_name
- xlanguage (highlighted)
- xcustomer_id
- xpayment_method_id
- xcontact_id
- xaddress_id

Figura 19: Llista desplegable

⁷ Permet enllaçar consultes prèviament definides en el repositori amb l'objecte de negoci. L'objecte de negoci permet definir diverses funcionalitats utilitzant aquestes consultes: Càlcul de descripcions, camps calculats, valors possibles, màscares dinàmiques, validacions i cerques.

4.4.5. Picture

Es tracta d'un objecte capaç de mostrar qualsevol format gràfic. Aquest objecte pot associar-se a un arxiu gràfic, un control de tipus BLOB⁸ o una imatge de preimpressió.

Seguidament, veiem en la imatge com mostra en el requadre de la imatge el registre associat a la base de dades.

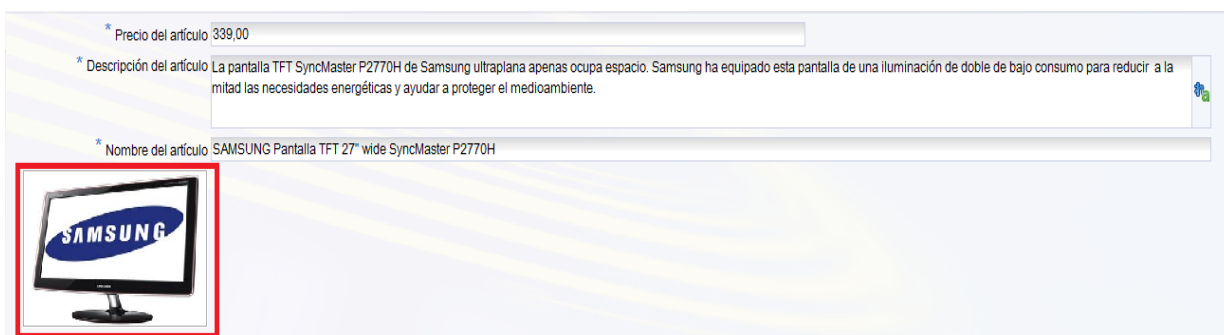


Figura 20: Imatge

4.4.6. Button

Es tracta d'un objecte de tipus botó, encara que se li pot associar qualsevol tipus de codi script disposen d'una sèrie de funcions predefinides que realitzen les accions més comuns.

Els botons poden aparèixer dins de cada grup comprimible per realitzar funcions íntegres del grup o poden aparèixer dins del grup de botons d'entitat on tenen funcionalitat global per a tot el formulari.

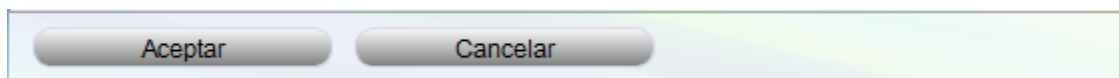


Figura 21: Botons

⁸ Els BLOB (Binary Large Objects, objectes binaris grans) són elements utilitzats en les bases de dades per emmagatzemar dades de gran grandària que canvien de forma dinàmica. Generalment, aquestes dades són imatges, arxius de so i altres objectes multimèdia.

4.5. Containers

Els containers, o contenidors, ens permetran agrupar totes les vistes que es creïn. Els tipus de contenidors que veurem són els més bàsics dins l'entorn de Walnut.

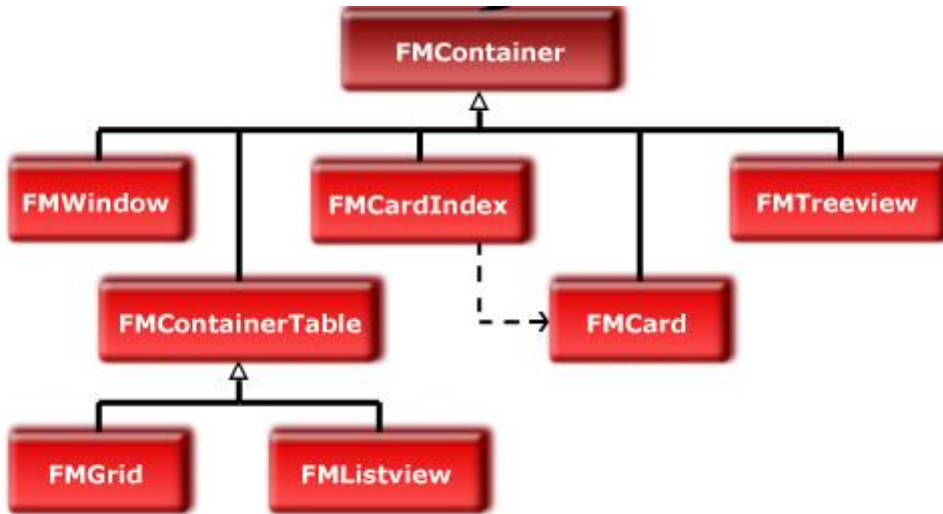


Figura 22: Diagrama de Containers

4.5.1. Grid

El contenidor de tipus grid és una estructura en forma de taula on es poden incrustar controls d'objectes de negoci a través de columnes. Aquestes columnes es poden modificar i afegir nous valors.

En la figura següent es pot veure com les dades estan carregades en el grid i que cada columna correspon a un control pertanyent a l'objecte de negoci.

sgrid1			
* Código	* Nombre	* Precio	Código EAN
1	SAMSUNG Pantalla TFT 27" wide SyncMaster P2770H	339,00	8400002000015
2	Epson LQ-2080 Impact Printer	592,07	8400002000022
3	Lexmark 2390 Plus	235,43	8400002000039
4	OKI ML-3320	464,00	8400002000046
5	Monitor tactil elo 17	485,00	8400002000053
6	Gateway LT3100 netbook	375,88	8400002000060
7	Gateway NV5933u	456,67	8400002000077

Figura 23: Taula (grid)

4.5.2. Card

El següent contenidor que veurem són les pestanyes, junt amb els grups comprimibles (frames), els més importants a l'hora de dissenyar nous formularis ja que d'ells dependrà pràcticament tota l'estructura.

Per dur a terme l'estructura de les pestanyes cal entendre abans que s'ha de crear prèviament un contenidor per a les pròpies pestanyes (CardIndex). Aquest recollirà les pestanyes creades i les anirà col·locant seguidament.

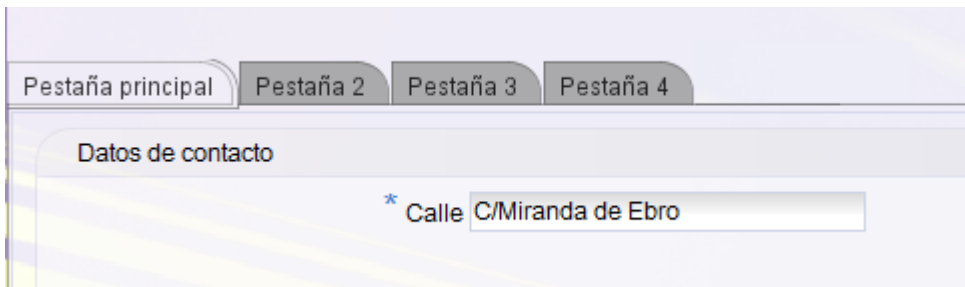


Figura 24: Pestanyes agrupades

4.5.3. Frame (Collapsible)

Els frame, Collapsible o grup comprimible, són els contenidors més importants ja que d'ells en depèn gran part de la funcionalitat del sistema Walnut. La seva funció consisteix en agrupar dades d'un mateix tipus per facilitar-ne la seva visualització de tal manera que sigui còmoda per l'usuari.

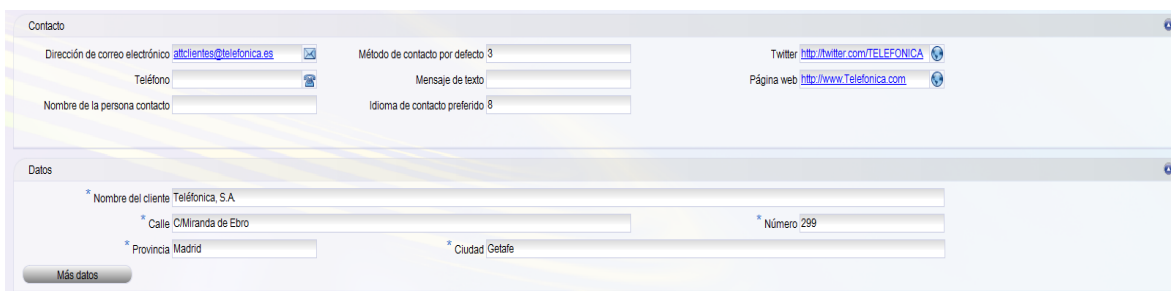


Figura 25: Grups comprimibles

Una de les particularitats dels grups comprimibles és que permet modificar l'ordre dels grups en funció del que l'usuari desitgi, tal com es pot veure a continuació.

Figura 26: Grups comprimibles intercanviats

L'altra particularitat és que, tal com diu el seu propi nom, permet comprimir aquests grups per tal de visualitzar la informació més imprescindible en aquell moment per l'usuari i amagar aquelles dades que en aquell instant no siguin necessàries.

Figura 27: Grup comprimit

4.5.4. TreeView

El següent contenidor que veurem potser no és un dels més utilitzats però en el nostre projecte és fonamental per el seu desenvolupament ja que d'ell en depèn l'estructura del nou formulari. La vista en arbre (TreeView) ens permet visualitzar estructures, ja sigui de l'objecte de negoci o del propi formulari.

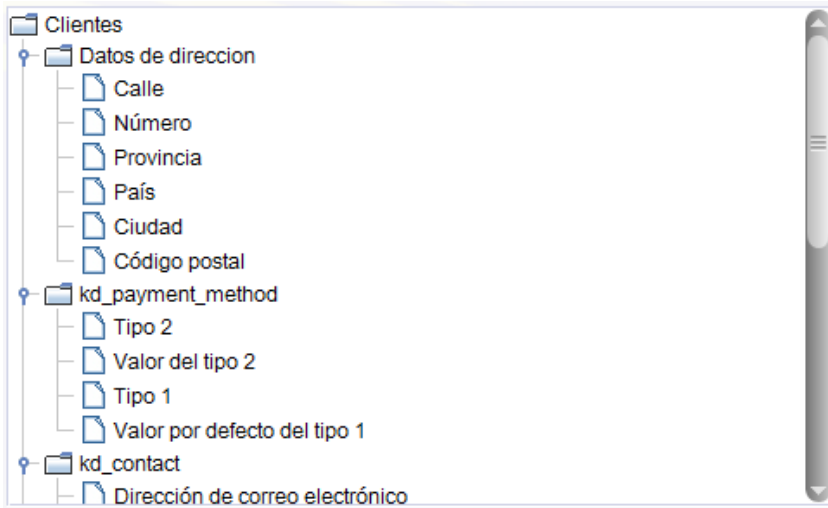


Figura 28: Vista en arbre

4.6. Propietats

Tal com s'ha esmentat en apartats anteriors cada vista requereix d'unes propietats imprescindibles per a la seva visualització. Tot seguit veurem detalladament aquestes propietats i quina funció tenen.

4.6.1. Nom del control

El nom del control d'una vista correspon al valor que mostrarà aquesta vista en funció del control que tingui associat a la base de dades. Per entendre-ho millor en la següent figura es pot veure com funciona aquesta propietat.

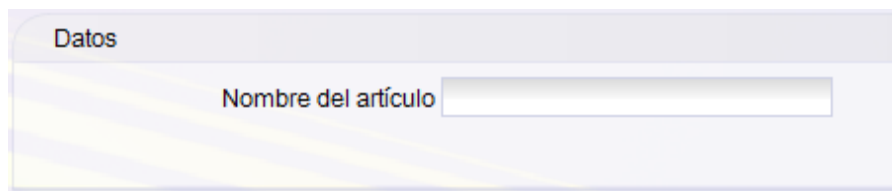


Figura 29: Vista sense control associat

Datos

* Nombre del artículo

Figura 30: Vista amb control associat

L'assistent permet la possibilitat de canviar el nom de control que té associat per qualsevol altre que pertanyi a l'objecte de negoci.

4.6.2. Etiqueta

L'etiqueta o label és la propietat encarregada de mostrar textos no modificables associats a una vista per tal de donar una breu descripció.

Datos

* Nombre del artículo SAMSUNG Pantalla TFT 27" wide SyncMaster P2770H

* Precio del artículo

Figura 31: Etiqueta d'una vista

4.6.3. Tamany

El tamany és la propietat més important ja que s'encarrega de la visualització de les vistes. S'ha de tenir en compte que hi ha diverses propietats pel que fa al tamany que cal considerar, entre d'altres, l'alçada, el llarg, etc. dels elements d'una vista. Tot seguit es detallaran més concretament aquestes propietats.

- ❖ Width (ample): determina l'ample de les vistes. Hi ha fins a tres nivells d'amplada per a cada tipus de vista.

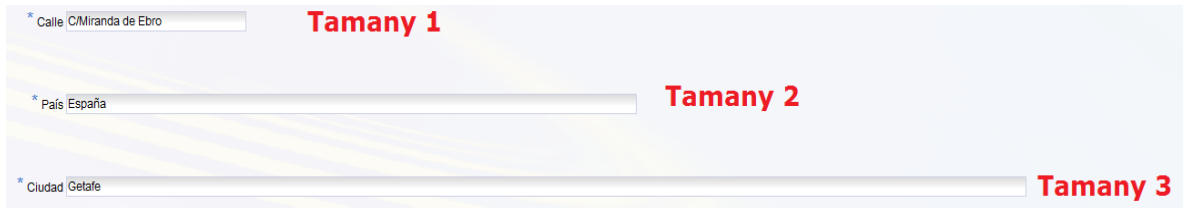


Figura 32: Diversos tipus d'ample

La distribució a seguir de les directrius Walnut funciona a nivell de columnes, en aquest cas, fins a tres columnes. Per exemple, una vista de tamany 3 ocuparia tot l'ample del formulari mentre que una vista de tamany 1 tindria espai fins a dues vistes més de tamany 1 o una de tamany 2 d'ample.

- ❖ Height (alçada): El height fixa l'alçada que tindrà la vista. A excepció de les imatges, serà constant per a totes les vistes com quadres de text, etiquetes, llistes desplegable, etc.

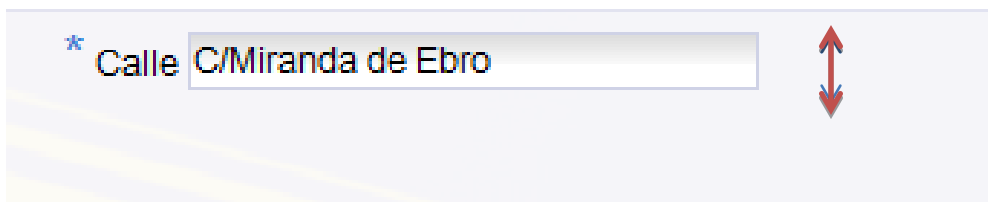


Figura 33: Alçada de les vistes

- ❖ Top (posició superior): Coordenada corresponent a la posició en l'eix vertical que tindrà una vista.



Figura 34: Coordenades top

- ❖ Left (posició a l'esquerra): Coordenada corresponent a la posició en l'eix horitzontal que tindrà una vista.

Datos

* Nombre del cliente	Teléfono, S.A.
* Calle	C/Miranda de Ebro
* Número	299
* Provincia	Madrid
* Ciudad	Getafe

Figura 35: Coordenada left

- ❖ TextWidth (ample del quadre de text): Correspon a l'ample de la part modificable d'una vista.
- ❖ LabelWidth (ample de l'etiqueta): Correspon a l'ample de l'etiqueta de la vista tot i que el seu contingut no es podrà modificar.
- ❖ DescWidth (ample del quadre de descripció): Part corresponent a l'ample, tot i que no sempre pot aparèixer, del quadre de descripció que pugui tenir associada una vista.



Figura 36: Composició de l'ampada d'una vista

4.6.4. Read Only

La propietat de només lectura (Read Only) indica quan l'usuari vol que la vista sigui exclusivament de lectura sense la possibilitat de modificar. Una forma senzilla d'identificar que una vista és només de lectura és que té un color tirant a gris a diferència del blanc habitual.

Figura 37: Quadre de text amb propietat de només lectura

4.6.5. Multilínia

Aquesta propietat és exclusiva de les vistes de tipus quadre de text i la seva principal funció és permetre visualitzar controls amb una enorme quantitat de text de forma que vagin saltant de línia i així facilitar-ne la seva visualització.

Figura 38: Quadre de text multilínia

4.6.6. Mode de visualització

Propietat pertanyent a les llistes desplegable que permet l'opció d'afegir un quadre de descripció on en funció del valor de la llista escollit mostri una breu descripció d'aquest.

Figura 39: Mode de visualització de les llistes desplegable

4.7. Events

Per al desenvolupament del projecte s'ha treballat amb events que pertanyen a l'API de *karat* on cada acció realitzada sobre el formulari disparava l'event que s'ha hagut de codificar per tal d'assolir els objectius marcats al projecte. Tot seguit veurem els diversos events utilitzats i quina funció desenvolupen.

4.7.1. Menú click

Aquest event es dispara a l'escollir una opció del menú pop-up⁹ activat quan es fa botó dret sobre qualsevol node de l'arbre de l'estructura del formulari. Entre d'altres, aquest menú pop-up mostrarà les opcions disponibles depenent del node que cliquem com per exemple, afegir botó, afegir grup comprimible, borrar elements, etc. Un cop es cliqui una opció executarà la part de codi corresponent a l'opció escollida.

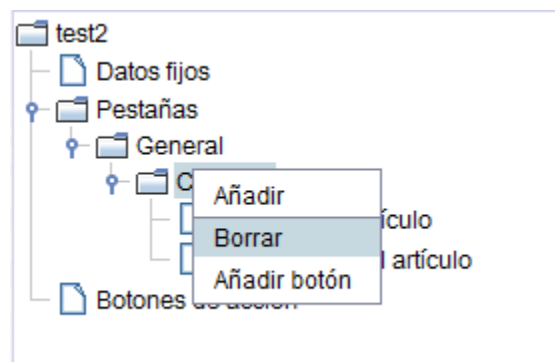


Figura 40: Menú pop-up corresponent a un grup comprimible

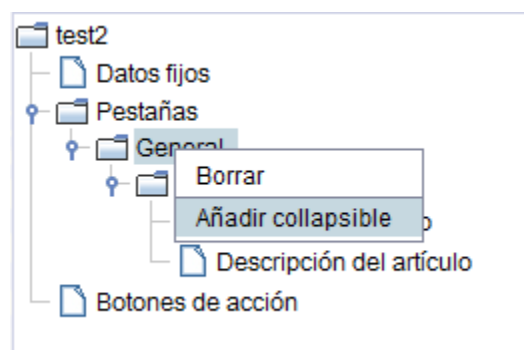


Figura 41: Menú pop-up corresponent a una pestanya

⁹ El terme denomina a les finestres o menús que emergeixen automàticament

4.7.2. Form Item Select

Mitjançant aquest event, que s'activa quan es selecciona qualsevol dels valors que conté un element de l'objecte de negoci, podem controlar en el nostre projecte cada cop que es vulgui canviar el tipus de vista per a un element seleccionat. En funció del valor escollit en crearà d'un tipus o de l'altre.

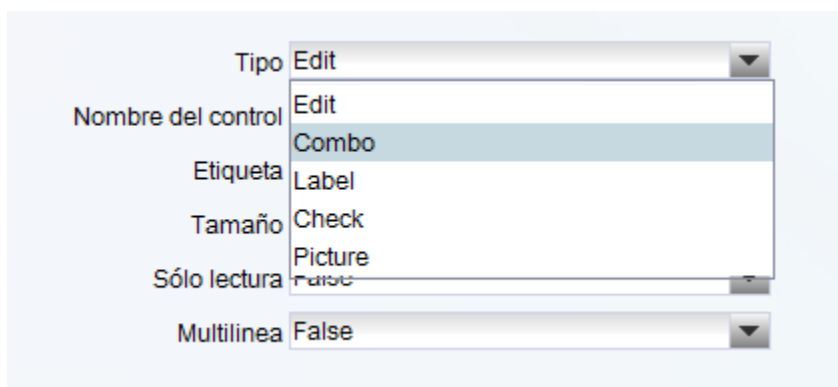


Figura 42: La selecció del tipus de vista activarà l'event

4.7.3. Form Item Validate

Aquest event es dispara quan es modifica el valor a través d'una acció directa de l'usuari o en el procés previ al guardar un registre per verificar que el valor introduït compleix les normes que l'aplicació pugui requerir.

Probablement és l'event més, o dels que més, important del projecte ja que és l'encarregat de verificar tots i cada un dels valors que l'usuari anirà afegint a mesura que vagi creant el seu formulari. Més concretament, tot l'apartat de propietats ja que hi ha fins a cinc propietats i cada una pot arribar a tenir diverses opcions a escollir en funció dels gustos o necessitats de l'usuari.

Per a l'assignació de propietats es treballa directament amb l'API de *karat* que ofereix la possibilitat de que totes les vistes que es creïn afegir-les a una llista automàticament de tal manera que cada cop que es vol modificar alguna propietat es recull la vista d'aquesta llista, es modifiquen les propietats que l'usuari vulgui i es guarda actualitzada.

Un petit inconvenient que té aquest event és que s'han de fer les validacions una a una, per exemple, quan es modifica el tamany d'una vista o l'etiqueta necessita realitzar una comprovació prèvia sobre quin tipus de vista està

afegint propietats i el mateix per a qualsevol propietat, recollint el valor de la llista, modificant-lo i tornant-lo a guardar. Això pot propiciar l'aparició d'errors ja que si una propietat no és afegida en una vista pot ser que no es pugui visualitzar correctament.

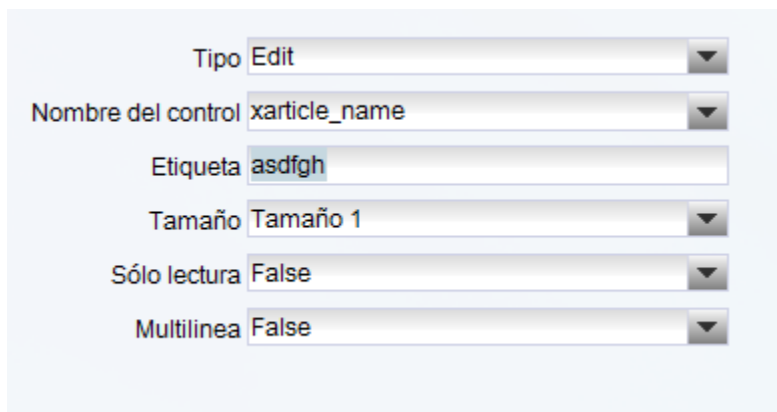


Figura 43: La modificació d'una propietat activa l'event

4.7.4. Form Post Load

Event que s'activa just abans de carregar completament el formulari, en aquest cas l'assistent. En el nostre cas no té cap funció vital en el desenvolupament de l'aplicació més enllà d'activar la funcionalitat del menú pop-up i d'amagar l'assistent fins que no s'afegeixin unes dades predeterminades com el nom del formulari i l'objecte de negoci que tindrà associat.

4.7.5. View Click

Aquest event, que s'activa quan es clica qualsevol botó, és també dels més importants dins del nostre projecte ja que s'encarrega de la col·locació de totes les vistes al nou formulari creat, de guardar totes aquestes vistes a la base de dades i de mostrar el formulari creat.

Com s'acaba de comentar, les dues funcions més importants que realitza aquest event són la col·locació i guardar a la base de dades les vistes. Pel procés de col·locació s'ha de tenir en compte que cada grup comprimible té els seus propis valors de top i left, cosa que *a priori* dificulta la feina ja que un grup comprimible pot tenir més vistes que un altre i que el número de vistes

del formulari canvia constantment. Per controlar aquest problema s'ha creat una llista on cada element pertany a un objecte amb cada un dels grups comprimibles creats i amb els seus valors de top i left. De tal manera que a l'hora de col·locar vistes es comprova a quin grup comprimible pertany i se li atorga un valor de top i left en funció del tamany de la vista. També s'ha comprovat que quadressin totes les vistes respectant les directrius Walnut.

Pel que fa al procés de guardar les vistes a la base de dades, és un procés que realitza automàticament l'API de *karat* mitjançant una comanda.

Un requisit fonamental a l'hora de controlar aquest event és que s'ha d'identificar quin és el botó que activa l'event ja que sinó qualsevol botó realitzaria una funció que no li correspon.

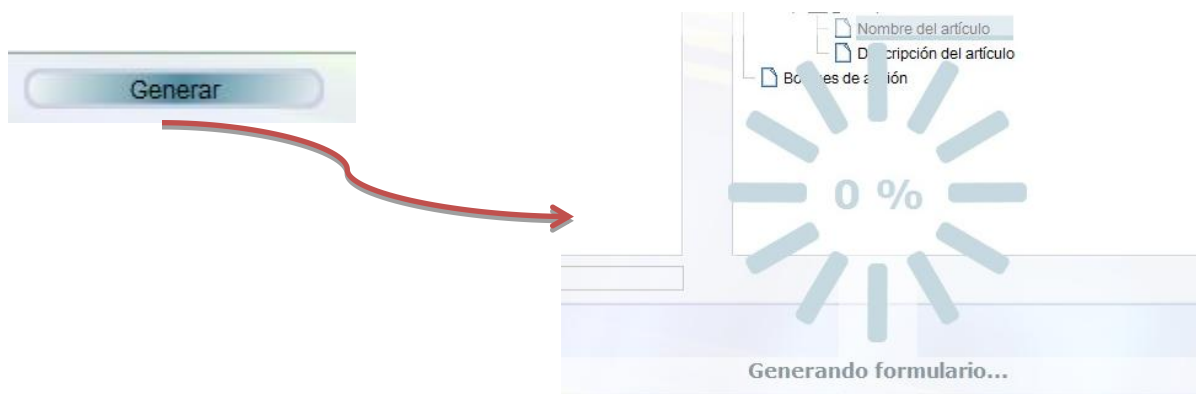


Figura 44: Funcionament de l'event

4.7.6. View Validate

Aquest event funciona exactament igual que el Form Item Validate amb l'única diferència que només s'activarà quan la vista en qüestió no tingui associat cap control de l'objecte de negoci.

En la nostra aplicació tindrà utilitat quan es dona nom al formulari, ja que comprova si és un formulari nou o un ja existent. A més a més, la seva funció principal és la de carregar els formularis que hagin sigut creats amb l'assistent. Per dur a terme això cal fer unes comprovacions prèvies, concretament, verificar que el formulari ja existeix a la base de dades. Un cop verificat, mirar si ha estat creat pel propi assistent i si és així recuperar totes les dades corresponents al formulari: contenidors i vistes.

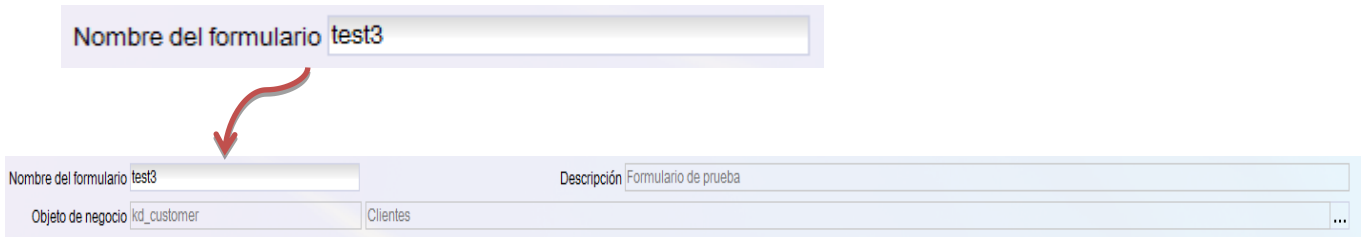


Figura 45: Validació d'un formulari ja existent

4.7.7. Container Drop

El següent event a veure juntament amb els que vindran a continuació són els pertanyents a les estructures en arbre utilitzades per a la creació del que serà l'estructura del formulari. Aquest en concret s'activa quan finalitza una acció de drag&drop.

La seva principal funció és la d'afegir el node seleccionat de l'arbre de l'objecte de negoci al node on l'usuari s'hagi posicionat amb el ratolí. Per dur a terme aquesta tasca l'API de *karat* en facilita el seu desenvolupament ja que té una funció que recull el valor de l'objecte on es realitza l'acció de drop.

Entre d'altres funcions, aquest event s'encarrega d'atorgar uns valors per defecte a la vista que sigui arrossegada.

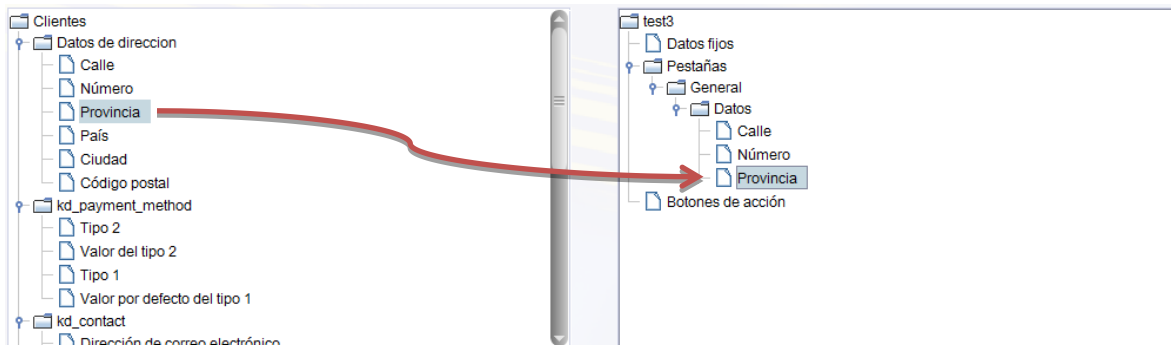


Figura 46: Element afegit mitjançant drag&drop amb posterior activació de l'event

4.7.8. Container Create Node

Aquest event s'activa a conseqüència de la creació d'un node en un arbre. En el desenvolupament de la nostra aplicació té un objectiu fonamental i és crear l'estructura de l'arbre del formulari tant l'estructura per defecte com la que l'usuari vagi modificant a mesura que afegeixi pestanyes, grups comprimibles, botons, etc.

S'ha de tenir en compte el fet que quan es crea un node nou cal diferenciar quina serà la seva funcionalitat. Per això, s'ha de verificar el valor d'aquest node nou. Aquest valor ve donat quan es crea el node a partir de l'event comentat anteriorment Menú click. Un cop reconeguda la seva funció l'aplicació li atorga les propietats necessàries.

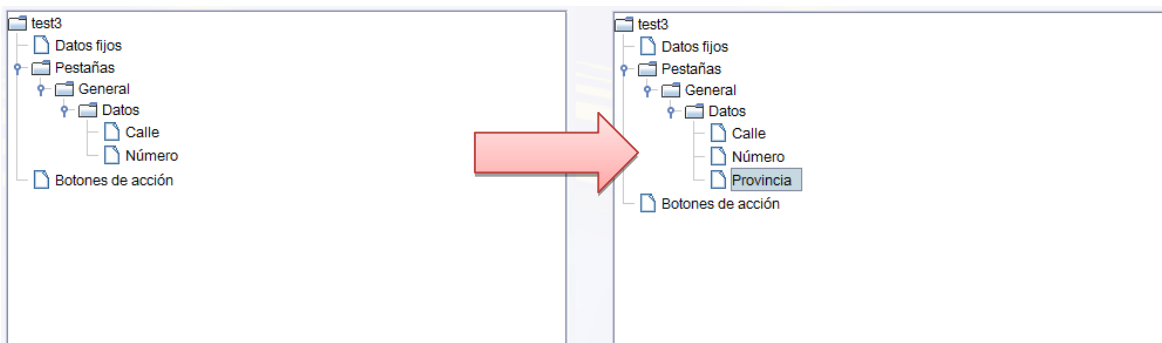


Figura 47: Node creat que activa l'event

4.7.9. Container Node Click

Mitjançant aquest event podem veure totes les propietats corresponents a cada node. Aquest event s'activa cada cop que fem clic a qualsevol node de l'arbre del formulari.

La seva principal funció, com s'ha comentat anteriorment, és la de mostrar les propietats corresponents a cada node que es seleccioni. Per dur a terme aquesta tasca es comprova, quan es clica un node, quin tipus de vista és i en funció de la vista mostrarà les propietats corresponents.

Per mostrar els valors de les propietats cal recollir de la llista de vistes els valors que tenen associats i fer un seguit de comprovacions per tal de mostrar-les en els quadres de text i permetre posteriors modificacions.

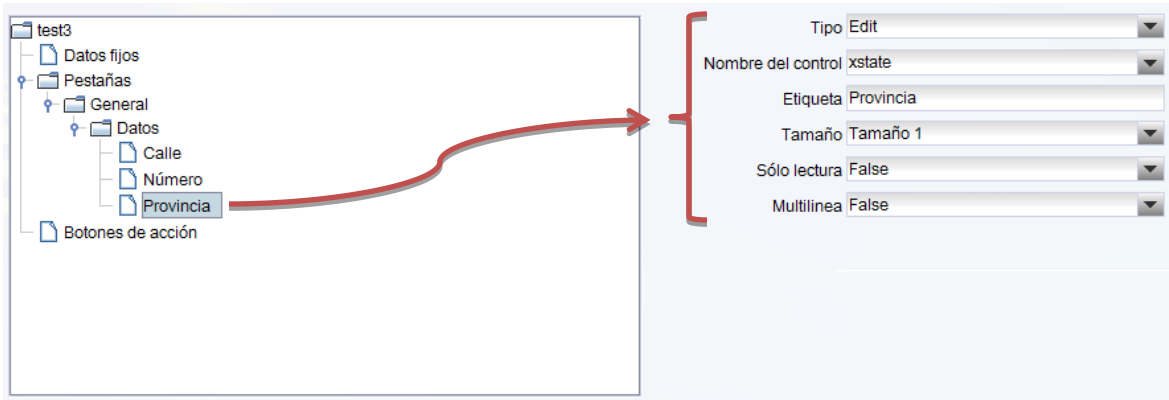


Figura 48: Cada node seleccionat mostra les seves corresponents propietats

4.7.10. PreSemantic

Aquest event pertanyent a l'API de *karat* s'utilitza per a la selecció de l'objecte de negoci del formulari obrint una finestra de selecció d'objectes de negoci associats al producte.



Figura 49: Finestra de selecció d'objectes de negoci

4.7.11. Altres mètodes

A més a més dels events esmentats durant l'anterior apartat, s'han fet servir mètodes creats manualment per realitzar unes tasques concretes pel que fa al desenvolupament de l'aplicació.

- ❖ LoadFormContainers: Aquest mètode es fa servir per crear l'estructura base del formulari, més concretament, la part corresponent a la part fixa, els botons d'entitat i l'àrea del formulari.
- ❖ Load: Està compost per diversos mètodes que corresponen a cada tipus de vista i contenidor que hi ha. La seva funció és mostrar i alternar les diferents propietats mitjançant el mètode de fer visible o invisible un quadre de text.

Tipo	Edit
Nombre del control	xdescription
Etiqueta	Descripción del artículo
Tamaño	Tamaño 1
Sólo lectura	False
Multilinea	False

Figura 50: Load corresponent als quadres de text

Tipo	Combo
Nombre del control	xdescription
Etiqueta	Descripción del artículo
Tamaño	Tamaño 1
Modo de visualización	Code + Desc

Figura 51: Load corresponent a les llistes desplegable

Tipo	Collapsible
Nombre del control	
Etiqueta	Datos de contacto

Figura 52: Load corresponent a un grup comprimible

- ❖ LoadBOTree: mètode encarregat d'omplir l'arbre de l'objecte de negoci en funció de l'objecte de negoci escollit prèviament. Per dur a terme aquesta tasca es llegeixen tots els valors de l'objecte de negoci i es crea l'arbre classificat per segments, sempre que en tingui, o directament els elements.

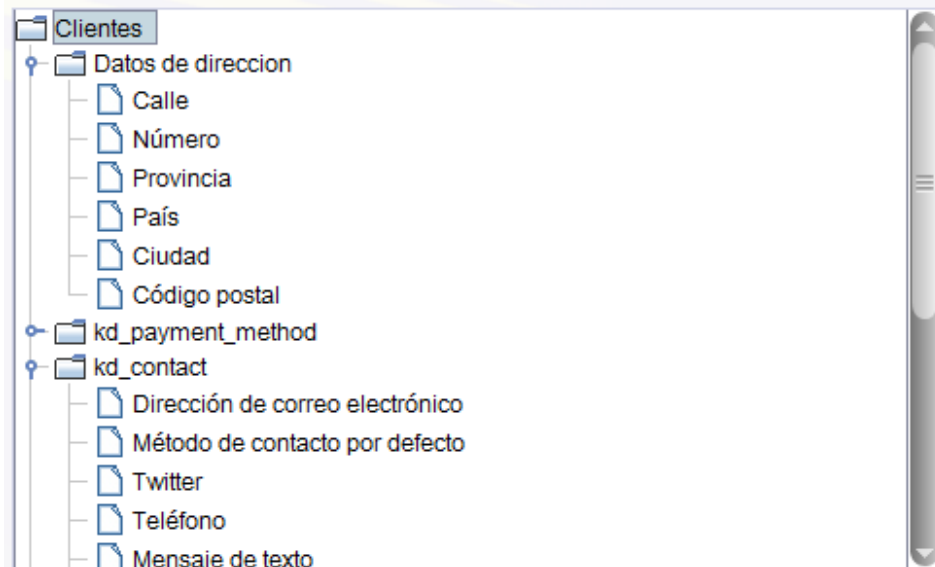


Figura 53: Objecte de negoci construït en forma d'arbre

- ❖ LoadFormTree: mètode utilitzat especialment per construir i reconstruir l'arbre del formulari amb la seva estructura tant si es crea per primera vegada com si es carrega un formulari ja creat per l'assistent.

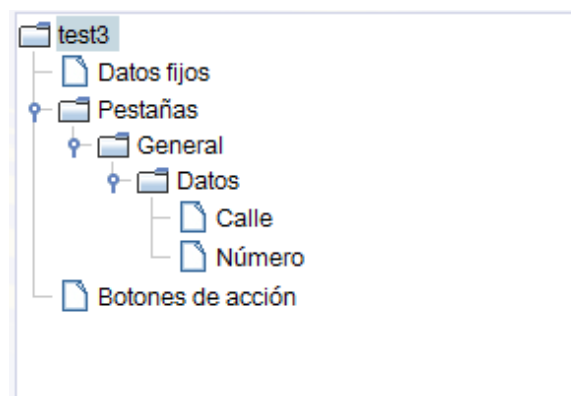


Figura 54: Estructura del formulari

5. Fase de proves

5.1. Control d'errors

Durant el desenvolupament del projecte i un cop finalitzat s'han realitzat diferents tipus de comprovacions per detectar i resoldre els diferents errors que podia tenir l'aplicació. A continuació se'n detallaran les més destacades.

- ❖ Afegir un element en un node incorrecte: S'ha de tenir en compte que durant la creació de l'arbre de l'estructura del formulari poden sorgir molts errors ja que utilitzant drag&drop es poden afegir nodes on no hauria d'estar permès. Per això, es verifica el valor del node on es produeix l'acció de drop i si no és correcte es mostra un missatge d'error i es cancel·la l'event.

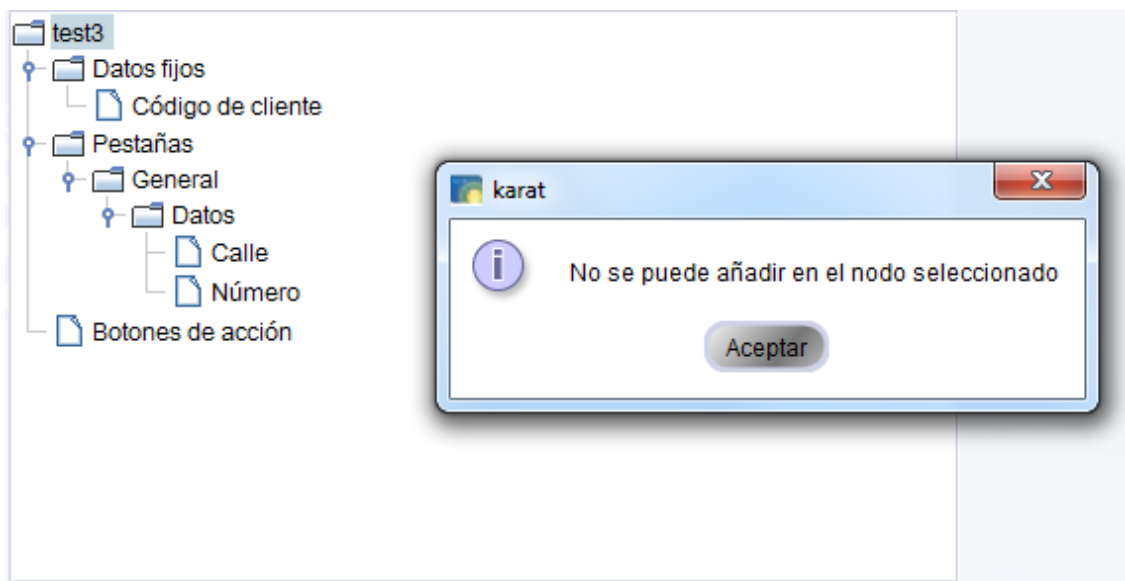


Figura 55: Missatge d'error al crear un node incorrecte

- ❖ Borrar elementos de l'arbre: Un altre focus de possibles errors consisteix en l'acció d'esborrar elements de l'estructura del formulari ja que no tots els nodes han de poder ser esborrats i a més a més un node pot tenir associat nodes fills penjant d'ell. D'aquesta manera es comprova prèviament el valor del node si pot ser esborrat, i si no es pot, mostra un missatge d'error i es cancel·la l'event.

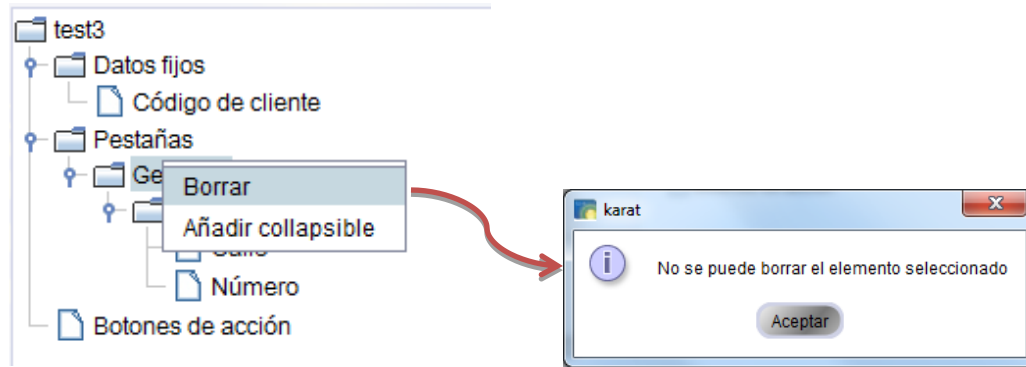


Figura 56: Missatge d'error al esborrar un node incorrecte

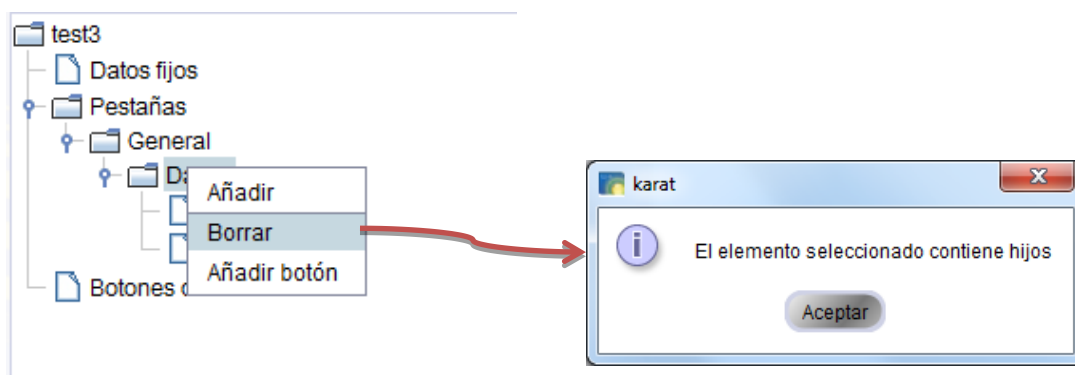


Figura 57: Missatge d'error al esborrar un node amb fills

- ❖ Selecció d'un objecte de negoci abans de donar nom al formulari: Abans de seleccionar l'objecte de negoci d'un formulari cal que aquest tingui un nom, per això es fa una comprovació prèvia i si no té un nom associat mostra un missatge d'error.

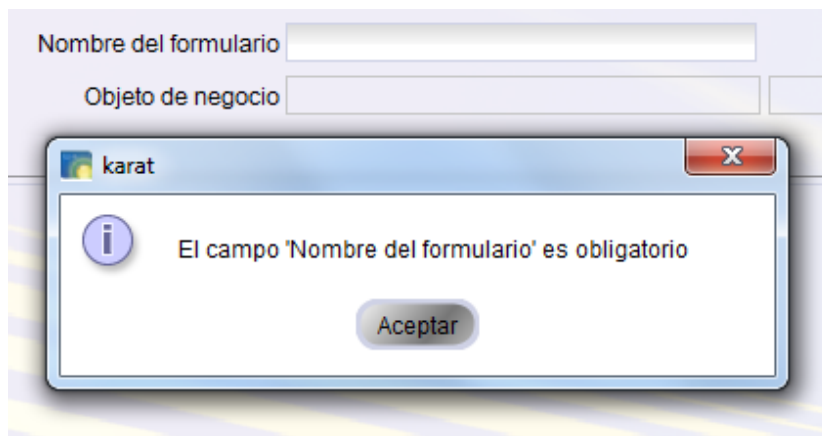


Figura 58: Missatge d'error al no tindre nom el formulari

- ❖ Afegir un node ja existent: Si s'intenta afegir un node que ja havia sigut afegit anteriorment a l'arbre del formulari l'assistent mostrarà un missatge d'error.



Figura 59: Missatge d'error quan s'afegeix un node duplicat

- ❖ Obrir un formulari que no ha sigut creat per l'assistent: Quan es guarda el formulari a la base de dades s'afegeix una propietat exclusiva per l'assistent per indicar que el formulari ha estat creat a partir d'aquest. Un cop es vol carregar un formulari, es comprova el valor d'aquesta propietat i si no ha estat creat per l'assistent mostra un missatge d'error.

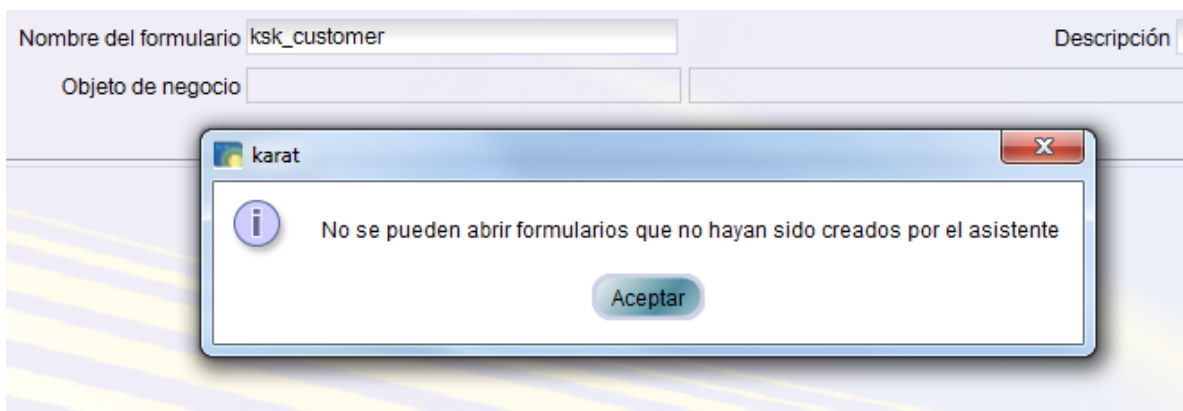


Figura 60: Missatge d'error al obrir un formulari que no s'ha creat amb l'assistent

- ❖ Crear un formulari sense nom ni objecte de negoci associat: Per crear un formulari cal que aquest tingui nom i un objecte de negoci associat, per tant, quan es clica el botó de Generar sense que tingui aquestes dues propietats mostrarà el missatge d'error corresponent.

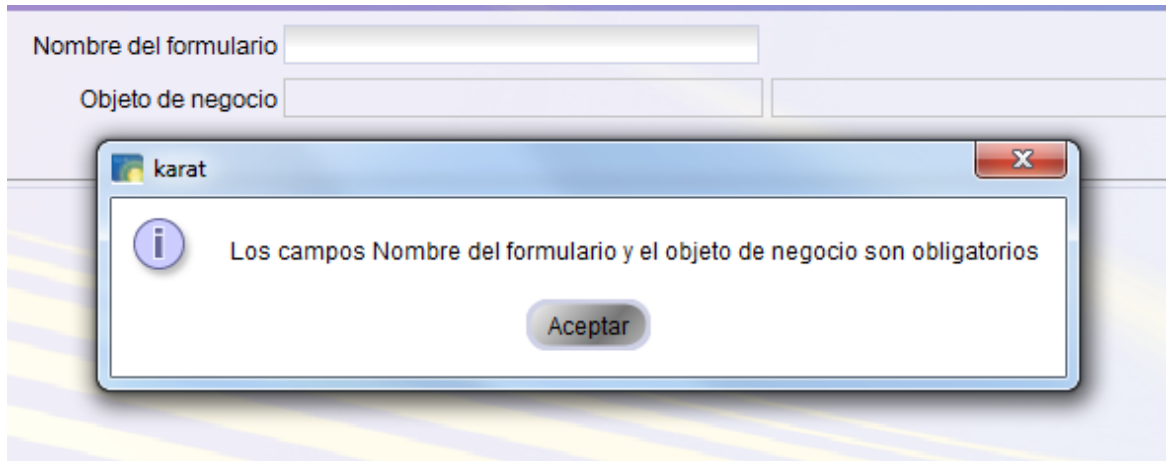


Figura 61: Missatge d'error quan no s'indica nom ni objecte de negoci

- ❖ No es pot donar el nom del propi assistent: Per evitar possibles errors no desitjats s'ha denegat la possibilitat de donar com a nom del formulari el mateix que l'assistent.

5.2. Proves

A més a més d'un control d'errors d'entrada durant el desenvolupament i posteriorment s'ha anat fent un seguit de proves per tal de verificar el bon funcionament de l'aplicació i que mostri els resultats esperats.

5.2.1. Proves unitàries

Les proves unitàries són aquelles que ens permeten assegurar que un determinat component de l'aplicació retorna una sortida correcta per una determinada entrada.

- ❖ Comprovació de totes les propietats de l'assistent: S'ha comprovat que les propietats s'alternen correctament en funció del tipus de vista que es seleccioni i que els valors de les llistes desplegable són correctes per cadascun.
- ❖ Validació de l'objecte de negoci: S'ha comprovat que el llistat d'objectes de negoci sigui el correcte per al producte corresponent i que la seva selecció conté tots els camps correctes.

- ❖ Comprovació de la creació de l'arbre del formulari: S'ha comprovat que tots els nodes són arrossegats correctament d'un arbre a l'altre en els llocs específics, que el menú pop-up del botó dret del ratolí crea l'estructura del formulari i que es poden afegir nous elements.
- ❖ Validació de creació del formulari: S'ha comprovat que es recullen tots els valors necessaris per fer la inserció a la base de dades de les vistes del formulari i que aquestes s'han afegit correctament.
- ❖ Comprovació de les propietats d'un formulari: S'ha comprovat que a mesura que es clica en un dels nodes de l'arbre mostra les propietats corresponents que tingui associat i que els valors es van actualitzant a mesura que es canvien.
- ❖ Comprovació de la correcta visualització del formulari creat: S'ha comprovat que un cop generat el formulari al clicar al botó de visualitzar-lo s'obre la finestra correctament amb el formulari segons les vistes que l'usuari hagi afegit.
- ❖ Comprovació de la correcta càrrega d'un formulari existent: S'ha comprovat que quan es carrega un formulari existent aquest mostra l'estructura i les propietats que tenia en el moment de la seva creació i, a més a més, permet la seva modificació.

5.2.2. Proves d'integració

Les proves d'integració tenen com a finalitat provar el funcionament conjunt de l'aplicació. Això vol dir assegurar que els mòduls i accions provades en el punt anterior funcionen correctament de forma conjunta.

- ❖ Prova 1: S'ha creat un nou formulari associant-li un objecte de negoci, s'han afegit diverses vistes i se'ls ha donat diverses propietats. Per acabar, es clica el botó de Generar i acte seguit el de visualitzar.
Resultat: Correcta visualització del formulari amb els valors corresponents a cada camp de la base de dades.
- ❖ Prova 2: S'ha modificat el tamany d'algunes vistes i es torna a clicar el botó de Generar i el de visualitzar.
Resultat: Les vistes es recalculen i mostren novament el formulari

reorganitzat en funció del nou tamany de les vistes.

- ❖ Prova 3: Es modifica la propietat de només lectura de diverses vistes i es torna a generar el formulari.
Resultat: Es visualitza el formulari modificat on les vistes de només lectura no permeten editar el contingut.
- ❖ Prova 4: Es modifica la propietat de multilínia de les vistes de quadres de text i es torna a generar el formulari.
Resultat: Es visualitza el formulari modificat on les vistes de quadre de text que s'hagin marcat amb la propietat multilínia mostren tota la informació corresponent dins del quadre habilitat.
- ❖ Prova 5: S'ha canviat el tipus de vista dels elements i s'han tornat a afegir les propietats corresponents. Es torna a generar el formulari.
Resultat: Formulari modificat correctament amb la seva corresponent visualització dels canvis fets.
- ❖ Prova 6: S'afegeixen més pestanyes i grups comprimibles amb les seves corresponents vistes i es torna a generar el formulari.
Resultat: Correcta modificació del formulari i verificació que les noves pestanyes i grups comprimibles contenen les vistes especificades prèviament.
- ❖ Prova 7: S'han esborrat vistes i contenidors amb el menú de botó dret i es torna a generar el formulari amb la seva posterior visualització.
Resultat: Les vistes esborrades ja no apareixen en el formulari.
- ❖ Prova 8: S'ha carregat un formulari creat anteriorment.
Resultat: es recupera tota la informació corresponent al formulari tal com el seu nom, l'objecte de negoci que té associat, l'estructura del formulari que tenia i les vistes i contenidors amb les seves corresponents propietats.
- ❖ Prova 9: S'han tornat a afegir vistes i contenidors en un formulari carregat, s'han modificat les propietats corresponents i s'ha tornat a generar el formulari.
Resultat: El formulari es torna a visualitzar tal com estava quan es va crear i a més a més es visualitzen les noves incorporacions tant de vistes com de contenidors.

6. Conclusions

6.1. Objectius assolits i no assolits

Un cop arribats a aquest punt cal fer balanç dels objectius que s'han assolit i dels que no. En línies generals podríem dir que s'han assolit gairebé tots els objectius que s'havien proposat a l'inici del projecte i que el projecte compleix amb els requisits que s'han detallat.

Tot seguit es llisten els objectius amb el seu compliment.

- ✓ Permetre la selecció de qualsevol objecte de negoci del producte.
- ✓ Creació dels arbres d'objecte de negoci i formulari amb posterior intercanvi d'elements entre ells.
- ✓ Assignació de propietats per a cada element seleccionat.
- ✓ Generar el formulari.
- ✓ Carregar l'estructura dels formularis prèviament creats.
- ✓ Compliment de les directrius Walnut.
- ✓ Ha de ser fàcil d'entendre i d'utilitzar amb una interfície atractiva per l'usuari.

A continuació es llisten els requisits funcionals i els no funcionals amb el seu compliment.

- ✓ RF1. Permetre la selecció de qualsevol objecte de negoci del producte.
- ✓ RF2. Posterior visualització dels segments i elements de l'objecte de negoci carregat prèviament en una vista en arbre.
- ✓ RF3. Construcció de l'estructura del nou formulari mitjançant drag&drop entre dues vistes en arbre.
- ✓ RF4. Assignació de propietats per a cada tipus d'element seleccionat.
- ✓ RF5. Creació d'elements més utilitzats com quadres de text (Edit), quadres desplegable (Combo), etiquetes (Label), imatges (Picture) i caps de control (Check).
- ✓ RF6. Creació de contenidors com taules (Grid), pestanyes (Card) i grups comprimibles (Collapsible).
- ✓ RF7. Generar el formulari ja finalitzat.
- ✓ RF8. Carregar l'estructura dels formularis prèviament creats.
- ✓ RF9. Visualitzar el formulari creat.

- ✓ RNF1. Complir les directrius Walnut.
- ✓ RNF2. Tolerància a errors i accions incorrectes.
- ✓ RNF3. Disseny visual agradable i fàcil d'utilitzar per l'usuari.
- ✓ RNF4. El software ha de permetre ser ampliable

6.2. Desviació de la planificació

S'ha aconseguit seguir la planificació temporal inicial gràcies en part a que durant la planificació es va ser generós en la distribució d'hores per a les tasques. L'únic detall a comentar és que durant la fase de desenvolupament s'ha anat barrejant amb la fase de test i proves ja que a mesura que s'anava codificant també es feien proves i controls d'errors.

Cal indicar també que en diverses ocasions al llarg del disseny i codificació s'ha hagut de refer treball per un mal plantejament o per decisions del tutor. Això no ha suposat cap problema i el resultat ha millorat amb cada modificació.

A continuació es mostra una taula amb les tasques que han tingut un desviament de les hores previstes, en aquest cas positivament.

Tasca	Dates previstes	Dates reals
Desenvolupament	26/01/2012 – 24/04/2012	26/01/2012 – 16/04/2012
Tests i proves	25/04/2012 – 17/05/2012	17/04/2012 – 27/04/2012
Documentació	18/05/2012 – 22/06/2012	02/05/2012 – 22/05/2012

Taula 11: Desviació de les dates del projecte

6.3. Possibles ampliacions i millores

Tot i que l'aplicació compleix amb gairebé tots els requisits que s'han demanat encara es pot ampliar considerablement l'assistent afegint-hi totes les vistes que encara no s'han implementat i també afegint moltes altres propietats que es contempen en la definidora del *karat Studio*.

A nivell de disseny una possible ampliació a llarg termini seria implementar una interfície gràfica i visual que permetés moure manualment les vistes que s'afegeixin i poder col·locar-les on l'usuari desitgi.

6.4. Valoració personal

A nivell personal aquest projecte ha resultat una experiència molt enriquidora ja que la Universitat Autònoma de Barcelona m'ha donat l'oportunitat de realitzar un projecte de gran envergadura en una empresa puntera a nivell internacional i perdre així les pors que tenia prèviament.

Un aspecte important que he tingut en compte és que he pogut realitzar un projecte des de zero amb totes les seves fases de desenvolupament: anàlisi, disseny, desenvolupament, documentació, etc. Tot i que en alguns casos m'han guiat durant el transcurs del projecte, la responsabilitat ha recaigut sobre mi en totes les seves fases.

He pogut comprovar en primera persona el grau de dificultat que té fer un projecte d'aquesta envergadura ja que és el primer cop que tenia contacte amb el sector informàtic dins el món laboral.

L'únic inconvenient, sense gran importància, a remarcar és el fet que ha estat un desenvolupament barrejant fases d'aprenentatge i fases més centrades en la realització del projecte. Per ser més clars, al treballar amb una eina nova cal un temps considerable d'habitució i es nota en la distribució del temps en les tasques.

Per tots aquests motius i per concloure, la valoració personal ha estat molt positiva encara que s'hagin dedicat més hores de les establertes per a un projecte final de carrera.

Bibliografia

A continuació es mostren les referències bibliogràfiques que s'han utilitzat per realitzar el projecte, aparentment poden semblar poques però donada la naturalesa del projecte el que més s'ha consultat és documentació interna de l'empresa.

Format escrit

- Documentació oficial de l'API de *karat*.
- Documentació referent a les directrius Walnut.

Format Web

- <http://es.wikipedia.org>
És una enciclopèdia lliure i poliglota de la Fundació Wikimedia (una organització sense ànim de lucre). Durant el transcurs del projecte s'ha fet servir per consultar definicions de conceptes com: IDE, drag&drop, BLOB, etc.
- <http://docs.oracle.com/javase/7/docs/api/>
API oficial de Java.

Contingut del CD

- ❖ Memòria del projecte en format PDF.
- ❖ Script de l'assistent amb el codi de l'aplicació en format PDF.
- ❖ Presentacions realitzades a UNIT4.

Eduard Sciortino Poch
a 22 de Maig de 2012