

UAB

Universitat Autònoma
de Barcelona

Kookan : share your knowledge

Memòria del projecte d'Enginyeria Tècnica en

Informàtica de Sistemes

realitzat per

Toni Tienda Fernàndez

i dirigit per

Yolanda Benítez Fernández

Escola Universitària d'Informàtica

Sabadell, Juliol de 2009

El/la sotasignant, Yolanda Benítez Fernàndez,
professor/a de l'Escola Universitària d'Informàtica de la UAB,

CERTIFICA:

Que el treball al que correspon la present memòria
ha estat realitzat sota la seva direcció
per en Toni Tienda Fernández
I per a que consti firma la present.
Sabadell, Juliol de 2009

Signat: Yolanda Benítez Fernández



Kooxlan
Share your knowledge

*Desenvolupat per Toni
Tienda Fernàndez*

*Dirigit per Yolanda
Benítez Fernàndez*

Juliol de 2009

CONTINGUT

Introducció	6
Presentació	6
Motivacions	6
Estat de l'art.....	7
Objectius	8
Estructura de la memòria	8
Introducció.....	8
Anàlisi previ	8
Estudi de viabilitat.....	9
Anàlisi funcional	9
Anàlisi tecnic	9
Codificació	10
Tests	10
Conclusions.....	10
Bibliografia	10
Agraïments.....	10
Annexos	10
Anàlisi previ.....	12
Objectius	12
Requeriments	12
Requeriments funcionals.....	12
Requeriments no funcionals	14
Definició de les funcionalitats	14
Gestió d'usuaris.....	14
Cerca d'usuaris.....	15

Accions contra usuaris.....	16
Agenda	16
Backoffice	17
Estudi de viabilitat	18
Introducció	18
Objectius	18
Requeriments	18
Avaluació de riscos.....	19
Planificació	19
Definició de les Tasques	19
PlaNificació.....	22
Pressupost.....	23
Equipament	23
Personal	24
Conclusions.....	26
Conclusions.....	28
Anàlisi funcional.....	29
Objectius	29
Casos d'ús.....	29
Actors	29
Índex de casos d'ús	30
Definició dels casos d'ús	31
Diagrames de casos d'ús	38
Conceptes teòrics.....	43
Introducció	43
Facebook.....	43

Google Maps	43
API – Application programming interface	44
Framework – MArc de treball.....	44
Programació per capes.....	45
MVC – Model View Controller.....	47
AOP – Aspect oriented programming	48
Introducció a OOP	48
Avantatges d'AOP	49
Llenguatges d'Script	50
Javascript.....	50
AJAX - Asynchronous JavaScript And XML	50
jQuery	51
Glosari de termes	51
Anàlisi tècnic	53
Introducció	53
Model de dades.....	53
Tsoobame	54
Kookan.....	55
Facebook.....	56
Estructura del projecte	57
Tsoobame.Credentials.....	57
Tsoobame.Data	57
Tsoobame.Facebook.....	58
Tsoobame.Model.....	58
Tsoobame.Business	58
Tsoobame.Kookan.Data	58

Tsoobame.Kookan.Business	58
Tsoobame.Kookan.Model.....	58
Tsoobame.Kookan.Credentials	58
Tsoobame.Kookan	59
Codificació.....	60
GEstió dels usuaris	60
Control d'Excepcions	60
Control de Permisos	61
Seguretat.....	62
Crides per Ajax.....	63
TEst	64
Introducció	64
Resultats.....	64
Tests Funcionals	64
TEstde rendiment	70
Tests d'estres	70
Conclusions	71
Objectius	71
Línies obertes	71
Conclusions finals.....	71
Bibliografia	72
Bibliografia esmentada a la memòria	72
Bibliografia consultada	72
ASP .NET	72
Javascript.....	72
jQuery.....	72

Facebook API.....	73
GoogleMaps API.....	73
Agraïments	74
Annexos.....	75
Requeriments del client	75
Anàlisi i Síntesi dels requeriments del client	77

INTRODUCCIÓ

PRESENTACIÓ

En la societat actual on l'adquisició de nous coneixements és quelcom necessari, la majoria de gent busca noves eines amb les quals aconseguir aquests nous coneixements. Les eines tradicionals com llibres, mestres, escoles estan deixant pas a les noves tecnologies com Internet, cursos en mp3, etc.

La nostra empresa pensa que la millor eina per a aprendre són els intercanvis persona-persona (mestre-alumne, entre companys de classe, ...). Aquest és un dels motius que van fer sorgir la idea de **Kookan**.

El projecte **Kookan** pretén ser una eina popular de cara a l'intercanvi de coneixements entre persones de tot el món. Fa servir les noves tecnologies en l'entorn web i les tendències de xarxes socials per a apropar la possibilitat de conèixer gent (mestres, companys d'estudi, ... amb les mateixes inquietuds, amb uns horaris semblants i a una distància adient per tal de realitzar aquests intercanvis.

En aquest document és presentaran detalladament les solucions que s'han implementat en base a les necessitats que s'han produït a partir de la idea general que s'ha explicat anteriorment. En veuran pas a pas les fases per les que ha anat passant el projecte des de l'anàlisi previ dels requeriments fins a les conclusions dels tests de l'aplicació així com les possibles millores que es podran realitzar a fases posteriors.

MOTIVACIONS

Hi ha força aspectes d'aquest projecte que em semblen interessants tant a nivell individual (personal i professional) com a nivell col·lectiu.

En primer lloc, crec que aportar una aplicació gratuïta que permeti a usuaris de tot el món cercar gent amb els mateixos interessos i amb les mateixes inquietuds d'aprendre nous coneixements i ensenyar els que ja es tenen d'una manera senzilla pot apropar l'ensenyament a col·lectius que potser no es poden pagar escoles particulars, o que potser no tenen una flexibilitat horària durant la setmana per a invertir en aquest tipus d'educació.

Per d'altra banda cada cop les relacions interpersonals es fan més via electrònica i encara que això està apropant la relació entre amics que viuen lluny, també està allunyant la relació entre amics que viuen a prop. Encara que es fa servir **Facebook** com a base per a l'aplicació, **Kookan** proposa les cites personals i físiques per a l'intercanvi de coneixements, i la tornada al traspàs de coneixement interpersonal per via oral tal i com es feia no fa tant de temps.

Com a motivació personal sempre m'ha agradat aprendre coses noves (idiomes, llenguatges de programació, cosmologia, etc) i la solució més fàcil era la cerca d'informació per Internet i la lectura de documents més o menys tècnics. Aquests documents solien ser d'un nivell que no es corresponia amb el que jo tenia a la matèria en qüestió i presentaven la dificultat de no poder preguntar els dubtes que podien sorgir. Amb aquest panorama crec que **Kookan** pot aportar molt a la gent amb interès en aprendre.

Per què es fa servir **Facebook**, GoogleMaps, ASP .NET, Ajax, JQuery, ... A part dels avantatges que aquestes eines presenten (difusió, robustesa, velocitat de resposta, atractiu visual, ...) el que em porta a fer servir aquestes tecnologies és l'afany d'aprendre noves tècniques i de fer servir eines que em serviran de cara a la meva carrera professional.

Aquest és un projecte que em motiva molt en molts aspectes.

ESTAT DE L'ART

Observant el mercat d'aplicacions per l'intercanvi de coneixements s'ha trobat que la majoria de aplicacions només fan referència a l'intercanvi d'idiomes. A part d'això, algunes tenen una funcionalitat molt bàsica gratuïta i quan vols contactar amb algun usuari has de subscriure't pagant una mensualitat.

Encara que algunes siguin totalment gratuïtes l'aplicació consta d'un buscador molt estàtic (país, província, ciutat) i un cop trobes algú que ensenyi el que vols aprendre has de gestionar les cites via e-mail.

D'altra banda, encara que n'hi ha moltes aplicacions d'aquest tipus, no s'han trobat gaires que estiguin integrades amb Facebook. Crec que la gent està donada d'alta a moltes aplicacions i d'aquí estan sorgint les idees del OpenID de Google o la obertura de la API a **Facebook** per tal de centrar els usuaris en una plataforma des de la que accedir a d'altres aplicacions.

Kookan proposa la creació d'Esdeveniments de **Facebook** per tal de controlar els intercanvis o les xerrades que l'usuari vulgui donar. Per tant pots fer un seguiment de les activitats que tens d'una manera senzilla, tant des de l'aplicació **Kookan** com des del propi **Facebook**. A més proposa la cerca per proximitat i la puntuació dels usuaris per tal de cercar els professors més adients.

Si **Kookan** no és una idea totalment nova i s'hi poden trobar coses semblants a la xarxa, crec que fa un ús molt correcte de les noves tecnologies per a facilitar la cerca i el manteniment dels intercanvis de coneixements entre usuaris i això la fa diferent a la resta de propostes.

OBJECTIUS

Com s'ha explicat a la introducció prèvia, el principal objectiu del projecte **Kookan** és del de proporcionar una eina útil i fàcil de fer servir perquè les persones amb interès en aprendre nous coneixements puguin trobar gent que estiguin interessades en ensenyar-los aquestes matèries. Aquest objectiu tant general es pot desglossar en els següents:

- Possibilitat de mantenir un currículum amb els coneixements que es tenen i els que es busquen.
- Possibilitat de cercar d'altres usuaris en funció de la proximitat.
- Possibilitat de puntuar els usuaris que siguin bons professors o bons alumnes per a crear una base de dades de confiança entre usuaris.
- Possibilitat de crear classes públiques en que un usuari ofereixi xerrades (o coses semblants) i a les que es puguin apuntar lliurement els usuaris interessats.
- Possibilitat de mantenir una llista amb els esdeveniments en els que l'usuari prendrà part.

ESTRUCTURA DE LA MEMÒRIA

Per tal de explicar el procés que ha portat a la consecució del projecte la memòria s'ha dividit en diferents seccions clarament diferenciades. En aquest apartat veurem aquestes seccions i una petita explicació de cadascuna:

INTRODUCCIÓ

És l'apartat on es troba el lector. Aquí es presenta el projecte com a idea i es defineixen els diferents aspectes que han donat lloc a la seva definició.

ANÀLISI PREVI

En aquesta secció és fa la recollida dels requeriments del client* i un anàlisi previ i síntesi d'aquests per tal de concretar-los i dividir-los correctament. Aquesta síntesi ha de ser acceptada pel client per tal de passar al següent punt, ja que a partir d'aquest document és basarà la resta del anàlisi i desenvolupament del projecte.

*En aquest projecte només hi ha una persona durant la presa de requeriments, definició, anàlisi i desenvolupament, però encara i així és parla de client, desenvolupador,... per la separació de rols necessària per a la consecució correcta d'un projecte.

ESTUDI DE VIABILITAT

A partir de l'anàlisi previ, es defineixen les tasques que s'hauran de fer, el personal necessari per a realitzar-les, els recursos de hardware o software necessaris, etc.

També es tenen en compte alternatives al projecte per veure si alguna aplicació comercial pot donar solució a les necessitats del client.

Amb tot això es valora la dificultat, el preu, i el temps que es trigarà en implementar l'aplicació i es decideix si el projecte és viable o no.

ANÀLISI FUNCIONAL

A l'anàlisi funcional es defineixen més detalladament les funcionalitats que ha de tenir l'aplicació i quins tipus d'usuaris podran realitzar cadascuna de les operacions que l'aplicació permet.

A partir d'aquesta definició es desglossaran les tasques que s'han de realitzar per tal de enfrontar el desenvolupament del projecte.

En aquesta secció només es defineix el **QUÈ** i el **QUI** de les operacions, és ha dir, què pot fer un usuari amb l'aplicació i qui pot realitzar cadascuna de les operacions (i qui no).

ANÀLISI TECNIC

En aquest apartat es defineix quines eines es faran servir, quina estructura tindrà el projecte, quines llibreries es poden aprofitar (tant de la pròpia companyia com de les que es poden trobar a Internet).

En resum, aquest apartat defineix el **COM** del projecte, és a dir, com es pretén realitzar el desenvolupament i amb quines eines.

CODIFICACIÓ

En aquesta secció es presentaran alguns exemples del codi que s'ha fet servir durant el desenvolupament. No pretén ser un anàlisi exhaustiu de tot el codi però sí una presentació d'alguns punts a destacar.

TESTS

Quan el desenvolupament finalitza és necessari un informe que demostrï que l'aplicació funciona correctament. En aquesta secció es presenta un informe amb els tests realitzats per a cadascun dels casos d'ús especificant l'operació que es realitza, quines dades es fan servir per al test, quin resultat s'espera i quin resultat produeix l'aplicació.

Durant aquesta fase del projecte solen sorgir petits errors que s'han de solucionar abans de fer l'entrega al client.

CONCLUSIONS

Aquí es presenten les conclusions del projecte com poden ser: dificultats que s'han trobat, possibles millores o ampliacions de cara al futur, etc.

BIGLIOGRAFIA

En aquest apartat s'especificaran les diferents fonts (llibres, webs, ...) que s'han fet servir per a la consecució del projecte.

AGRAÏMENTS

Volia agrair a algunes persones la seva ajuda tant de cara a la pròpia consecució del projecte com al ànim ens alguns moments del desenvolupament.

ANNEXOS

S'adjunten alguns documents que han estat necessaris per al desenvolupament del projecte, com poden ser els requeriments del client com la síntesi d'aquests requeriments entre d'altres.

ANÀLISI PREVI

OBJECTIUS

Per tal de comprendre les necessitats del client i, per tant, fer una valoració realista de l'envergadura del projecte, s'ha de fer un anàlisi previ del projecte que es pretén realitzar.

És necessari estructurar i definir concretament les necessitats que el client té per tal de crear un projecte que compleixi les els requeriments desitjats. També és de vital importància entregar el projecte finalitzat en les dates establertes i amb els costos definits en aquesta fase inicial del anàlisi.

Aquest anàlisi previ es el fonament sobre el que la resta de fases es recolzaran (tant fases posteriors d'anàlisi com el propi desenvolupament) .

REQUERIMENTS

El document adjunt **A0 - Requeriments del client** conté els requeriments que el client va demanar. A partir d'aquests requeriments previs, es va fer una redefinició per a facilitar la tasca d'anàlisi. Aquesta redefinició va ser acceptada per part del client amb data 03/03/2009.

Aquesta es la definició en detall d'aquest requeriments. A l'apartat **Requeriments funcionals** s'especifiquen les tasques que es realitzaran per a aquest projecte. A l'apartat **Requeriments no funcionals** s'especifica el comportament que ha de tenir l'aplicació en termes de rendiment, interacció amb l'usuari, ...

REQUERIMENTS FUNCIONALS

GESTIÓ D'USUARIS

Hi haurà un rol d'Administrador que podrà accedir al BackOffice.

Els usuaris seran Professors i Alumnes al mateix temps.

Poden especificar la seva situació (latitud, longitud).

Han d'especificar coneixements que s'ofereixen a ensenyar i coneixements que volen aprendre.

Es poden donar de baixa de dues formes:

Baixa temporal: El perfil es torna invisible per a tothom però l'usuari pot tornar a activar el perfil quan vulgui.

Baixa permanent: Les dades de l'usuari s'esborraran permanentment. Si vol tornar a fer servir l'aplicació s'haurà de tornar a enregistrar i tornar a introduir les seves dades denou.

INTERACTIVITAT AMB D'ALTRES USUARIS

Cerca d'usuaris:

La cerca d'usuaris es farà a partir de llistes categoritzades i fent clic:

Coneixements: selecció del coneixement que es vol cercar.

Localització: distància màxima d'on es troba l'usuari que fa la cerca.

Cerca intel·ligent dels usuaris que es corresponen amb les necessitats/horaris de l'usuari que fa la cerca.

L'ordenació de la cerca d'usuaris pot ser:

Popularitat (es calcula a partir del nombre d'usuaris que tenen el perfil com a favorit).

Nom.

Els usuaris poden puntuar els seus professors/alumnes.

AGENDA

Quan un alumne demana classe a un professor, establirà un horaris. Si el professor accepta les classes, el sistema afegirà aquestes "cites" a l'agenda. En cas de rebutjar-la, s'haurà d'informar del motiu a l'alumne que demanava la classe.

Cada usuari pot crear classes públiques a les que es podran apuntar lliurement els usuaris de **Kookan** als que els interessen la classe (o fins i tot usuaris de **Facebook** que no estiguin a **Kookan**).

BACKOFFICE

Des del BackOffice es podran gestionar els coneixements disponibles per a la web. Només hi haurà dos nivells de categorització: Categoria – Coneixement (Idiomes – Anglès per exemple).

Des del BackOffice es podran gestionar les traduccions de la web. Es podran visualitzar les claus que es fan servir a més d'un sistema base, des del qual es farà la traducció.

Des del BackOffice es podran visualitzar els Logs de l'aplicació per tal de revisar possibles errors.

Des del BackOffice es podran gestionar els usuaris (especificar qui es administrador).

REQUERIMENTS NO FUNCIONALS

APLICACIÓ

L'aplicació ha de ser multi idioma.

Ha d'estar integrada amb Facebook per aprofitar aquesta xarxa social i fer màrqueting viral.

Ha de tenir una alta interactivitat i per tant ha de funcionar amb rapidesa.

Ha de ser fàcil de fer servir. S'ha de prioritzar l'ús del ratolí en detriment de l'ús del teclat (important als filtres).

El disseny ha de ser senzill, professional però no massa seriós. Es tracta d'una aplicació d'us popular amb molts tipus de perfils (edats, cultures, ...).

S'han de crear camins redundants per arribar a un mateix resultat. Per exemple, s'ha de poder arribar al perfil d'un usuari des de els resultats de cerca, des de les cites amb aquest usuari, ...

S'ha de poder visualitzar i interactuar sense problemes amb els navegadors principals: IE Explorer 6+, Firefox 2+, Opera 8+, Safari 3+, Google Chrome 1+.

DEFINICIÓ DE LES FUNCIONALITATS

GESTIÓ D'USUARIS

LOGIN D'USUARIS A TRAVÉS DE FACEBOOK

Es tracta d'un sistema de Login integrat amb **Facebook**. Constarà d'un botó a la pàgina principal que obrirà la pantalla de Login que ofereix l'API de **Facebook**. A partir d'aquest moment l'usuari estarà dins de l'aplicació i podrà fer servir les funcionalitats que el seu rol li permetin.

LOGIN D'ADMINISTRADORS A BACKOFFICE

A la secció de BackOffice es poden trobar mòduls de gestió de l'aplicació i per tant només hi han d'accedir usuaris administradors amb permisos especials. S'ha de tenir especial cura en aquest aspecte per tal d'evitar danys irreparables a l'aplicació.

EDICIÓ DE PERFIL

Cada usuari ha de poder gestionar el seu perfil. Dins del perfil es troben dades com :

Localització (latitud, longitud).

Idioma (amb el que veurà l'aplicació).

Coneixements demandats

Coneixements oferts

BAIXA DELS USUARIS

Un usuari es pot donar de baixa de dues formes diferents:

Baixa temporal: l'usuari no estarà disponible (no podrà realitzar ni rebre accions per part d'altres usuaris), però les seves dades romandran emmagatzemades a la base de dades de l'aplicació.

Baixa permanent : s'esborraran totes les dades de l'usuari.

El fet de donar-se de baixa permanent no implica una impossibilitat de tornar a registrar-se.

CERCA D'USUARIS

CERCA PER CONEIXEMENTS I LOCALITZACIÓ

Es presentarà un formulari on l'usuari especificarà els coneixement que està buscant i la distància màxima que es vol desplaçar (si l'usuari no té especificada la seva localització, aquest camp no tindrà cap efecte).

CÀRREGA D'USUARIS QUE ET PODEN INTERESSAR

Llista d'usuaris que ensenyen coneixements que interessen a l'usuari que ha accedit a l'aplicació i que es troben a una distància predeterminada.

ORDENACIÓ DELS RESULTATS

Els resultats es podran ordenar per popularitat, nom dels usuaris, distància a la que es troben, o nivell al coneixement que hem buscat (si l'hem especificat).

ACCIONS CONTRA USUARIS

PUNTUACIÓ D'USUARIS

Els usuaris poden puntuar a la resta d'usuaris (mai a sí mateixos) en funció dels criteris que creguin adients. Així a cada usuari es mostrarà la puntuació mitjana que la resta d'usuaris els ha donat.

CITES

DEMANAR

Quan es demana cita a un usuari s'ha d'especificar l'horari de la cita, la localització, el coneixement que es vol intercanviar.

ACCEPTAR / REBUTJAR

Un cop s'accedeix a la pàgina de Login es presenta una llista de les cites que han demanat a l'usuari. En base a aquesta llista, l'usuari pot acceptar la cita o rebutjar-la (enviarà un missatge a l'usuari que havia demanat la cita).

AGENDA

VISUALITZACIÓ DE L'ESTAT DE L'AGENDA

Es visualitzarà un calendari amb les cites o esdeveniment als que l'usuari actual està apuntat.

BACKOFFICE

GESTIÓ DE CONEIXEMENTS

A la pàgina de gestió de coneixements es podran crear/eliminar/editar categories i coneixements establint els noms en els idiomes dels que es disposi a la web.

GESTIÓ DELS TEXTOS DE LA WEB

Per a gestionar els textos de la web es presentarà una llista amb les claus que es fan servir a la web, els texts en un idioma base (a escollir) i els text en un idioma objectiu. Els missatges en l'idioma base faciliten la traducció a d'altres idiomes (només la clau que és fa servir a la web no és suficientment aclaridora).

GESTIÓ D'USUARIS

A la gestió d'usuaris s podrà especificar els usuaris que són administradors i els que no ho són.

CONSULTA DE LOGS

A la secció dels Logs es podran visualitzar el missatges interns que va generant l'aplicació. Es podrà especificar el nivell mínim dels missatges que es volen veure (debug, info, warning, error) i el número de registres que es volen visualitzar.

ESTUDI DE VIABILITAT

INTRODUCCIÓ

Amb l'estudi de viabilitat es pretén fer una valoració dels costos, dels riscos i de la durada d'aquest projecte. Es definiran molt detalladament totes les tasques que s'han de realitzar, i quins perfils les han de realitzar. També s'estudiaran alternatives per tal de satisfer les necessitats del client. Amb això es podrà fer una valoració molt aproximada del impacte d'aquest projecte i permetrà decidir si es tracta d'un projecte viable o no.

OBJECTIUS

L'objectiu primordial del projecte **Kookan** es el d'oferir als usuaris una eina útil i senzilla de fer servir per tal d'intercanviar coneixements. Donat que l'educació (aprendre idiomes, fer classes de repàs, ...) és més important que mai, una eina que permeti cercar gent segons els coneixements que ofereixen o demanen i amb una gestió de les cites (classes) els usuaris podran accedir a uns coneixements que d'altra manera potser no es podrien permetre.

Desglossant una mica aquest objectiu, hi trobem el següent:

- Possibilitat de cercar usuaris que ofereixin coneixements que interessin a l'usuari que realitza la cerca.

- Possibilitat de valorar els professors/alumnes per tal de premiar (amb més popularitat) als usuaris més implicats.

- Possibilitat de mantenir una agenda amb les cites o classes que es tinguin.

- Integració amb **Facebook** per tal d'arribar al màxim nombre de persones possible.

Entre els objectius principals també destaquen que l'aplicació ha de tenir una alta interactivitat i una alta velocitat de resposta.

REQUERIMENTS

Com a requeriment de Hardware serà necessari:

- PC Portàtil per al desenvolupament del Projecte.

- Opcionalment es podria fer servir un PC com a servidor de base de dades per a realitzar les proves. Aquest requeriment, pot ser substituït pel mateix PC de desenvolupament com a Servidor de proves.

Com a requeriment de Software serà necessari:

PC de Desenvolupament:

Windows XP Professional Edition.
IIS.
Visual Studio 2008.
.NET Framework 3.5.
Microsoft Office 2007

PC de Servidor de Proves :

Windows 2003 Server (si s'escau).
Servidor de base de dades SQL Server 2005*.

AVALUACIÓ DE RISCOS

Problemes de l'ús de Javascript/Ajax quan es facin servir navegadors nous, minoritaris o noves versions dels navegadors majoritaris.
Problemes derivats d'alguna actualització o canvi en les polítiques de **Facebook**.
Problemes derivats d'alguna actualització o canvi en les polítiques de GoogleMaps.

PLANIFICACIÓ

DEFINICIÓ DE LES TASQUES

Tasca	Duració
Anàlisi Previ	5,5 dies
Anàlisi de requeriments	12 hores
Definició de funcionalitats	20 hores
Estudi de viabilitat	5 dies
Requeriments	4 hores
Avaluació de riscos	3 hores
Pressuposts	3 hores
Planificació	16 hores
Conclusions	2 hores
Anàlisi funcional	3,63 dies
Casos d'us	20 hores
Anàlisi tècnic	4,13 dies

Definició del model de dades	8 hores
Definició de l'estructura de l'aplicació	8 hores
Definició del model de classes	8 hores
Codificació	53,75 dies
BDD	2,88 dies
Creació de la BDD	8 hores
Creació dels procedures	6 hores
Aplicació	50,88 dies
Home	10 dies
Accés a enregistrament	4 hores
Login de Facebook	12 hores
Accés a la cerca	4 hores
Accés al calendari	12 hores
Accés al backoffice	4 hores
Visualització cites que han contestat	4 hores
Visualització persones que et poden interessar	4 hores
Visualització esdeveniments que poden interessar	4 hores
Visualització de cites demanades	4 hores
Visualització usuaris més valorats	4 hores
Visualització usuaris més participatius	4 hores
Perfil d'usuari	7,25 dies
Visualització de dades d'usuari	8 hores
Edició de dades d'usuari	16 hores
Canvi d'idioma	4 hores
Valorar usuari	6 hores
Configuració permisos Facebook	4 hores
Demanar cita	8 hores
Cerca d'usuaris	4,88 dies
Cerca BDD Local	6 hores
Recepció dades Facebook	12 hores
Ordenació dels resultats	8 hores
Accés a perfil d'usuari	4 hores
Accés a perfil d'usuari a Facebook	2 hores
Cerca d'esdeveniments	5,25 dies
Cerca BDD Local	6 hores
Recepció dades Facebook	12 hores
Ordenació dels resultats	8 hores
Accés a perfil d'usuari	4 hores
Accés a perfil de l'esdeveniment a Facebook	4 hores
Agenda	7,25 dies

Visualitzar calendari	24 hores
Visualitzar llista completa esdeveniments	4 hores
Visualitzar detall de l' esdeveniment	4 hores
Recepció dades Facebook	12 hores
Accés a perfil de l' esdeveniment a Facebook	2 hores
Home Backoffice	2,38 dies
Accés a gestió de coneixements	4 hores
Accés a traduccions	4 hores
Accés a gestió d'usuaris	4 hores
Visualització de missatges d'usuaris (suggeriments, ...)	4 hores
Gestió de coneixements	4,63 dies
Gestionar categories de coneixements	12 hores
Gestionar coneixements	12 hores
Gestió de traduccions	3 dies
Crear nou idioma	4 hores
Traduir missatges de la web	16 hores
Marcar idioma com a disponible	4 hores
Gestió d'usuaris	1 dia
Canviar rol de l'usuari	8 hores
GeoLocalització	5 dies
Càrrega de mapa des de GoogleMaps	4 hores
Càrrega mapa editable GoogleMaps	4 hores
GeoLocalització Inversa - Càrrega de ciutat per Lat,Long	16 hores
Creació funcions de càlcul de distàncies	6 hores
Aplicar Mapa a Perfil usuari, Esdeveniment	6 hores
Aplicar càlcul de distàncies a cerques	4 hores
Tests	3,38 dies
Home	4 hores
Perfil d'usuari	1 hora
Cerca d'usuaris	1 hora
Cerca d' esdeveniment s	2 hores
Agenda	3 hores
Backoffice	1 hora
Coneixements	2 hores
Traduccions	2 hores
Gestió d'usuaris	1 hora
GeoLocalització	4 hores
Documentació	6,13 dies
Introducció	2 hores
Anàlisi	8 hores

Codificació	4 hores
Tests	3 hores
Unificació memòria i revisions	2 hores
Manual d'usuari	16 hores

PLANIFICACIÓ

Nombre de tarea	Duración	Comienzo	Fin	Nombres de los recursos
<input type="checkbox"/> Anàlisi Previ	5,5 días	mié 04/03/09	mié 11/03/09	Toni Tienda
Anàlisi de requeriments	12 horas	mié 04/03/09	vie 06/03/09	Toni Tienda
Definició de funcionalitats	20 horas	vie 06/03/09	mié 11/03/09	Toni Tienda
<input type="checkbox"/> Estudi de viabilitat	5 días	mié 11/03/09	mié 18/03/09	Toni Tienda
Requeriments	4 horas	mié 11/03/09	jue 12/03/09	Toni Tienda
Avaluació de riscos	3 horas	jue 12/03/09	vie 13/03/09	Toni Tienda
Presuposts	3 horas	vie 13/03/09	vie 13/03/09	Toni Tienda
Planificació	16 horas	vie 13/03/09	mié 18/03/09	Toni Tienda
Conclusions	2 horas	mié 18/03/09	mié 18/03/09	Toni Tienda
<input type="checkbox"/> Anàlisi funcional	3,63 días	mié 18/03/09	mar 24/03/09	Toni Tienda
<input type="checkbox"/> Anàlisi tècnic	4,13 días	mar 24/03/09	lun 30/03/09	Toni Tienda
Definició del model de dades	8 horas	mar 24/03/09	mié 25/03/09	Toni Tienda
Definició de l'estructura de l'aplicació	8 horas	mié 25/03/09	vie 27/03/09	Toni Tienda
Definició del model de classes	8 horas	vie 27/03/09	lun 30/03/09	Toni Tienda
<input type="checkbox"/> Codificació	53,75 días	lun 30/03/09	jue 11/06/09	Toni Tienda
<input type="checkbox"/> BDD	2,88 días	lun 30/03/09	jue 02/04/09	Toni Tienda
<input type="checkbox"/> Aplicació	50,88 días	jue 02/04/09	jue 11/06/09	Toni Tienda
<input type="checkbox"/> Home	10 días	jue 02/04/09	jue 16/04/09	Toni Tienda
<input type="checkbox"/> Perfil d'usuari	7,25 días	jue 16/04/09	lun 27/04/09	Toni Tienda
<input type="checkbox"/> Cerca d'usuaris	4,88 días	lun 27/04/09	lun 04/05/09	Toni Tienda
<input type="checkbox"/> Cerca d'events	5,25 días	lun 04/05/09	lun 11/05/09	Toni Tienda
<input type="checkbox"/> Agenda	7,25 días	lun 11/05/09	mié 20/05/09	Toni Tienda
<input type="checkbox"/> Home Backoffice	2,38 días	jue 21/05/09	lun 25/05/09	Toni Tienda
<input type="checkbox"/> Gestió de coneixements	4,63 días	lun 25/05/09	sáb 30/05/09	Toni Tienda
<input type="checkbox"/> Gestió de traduccions	3 días	lun 01/06/09	mié 03/06/09	Toni Tienda
<input type="checkbox"/> Gestió d'usuaris	1 día	jue 04/06/09	jue 04/06/09	Toni Tienda
<input type="checkbox"/> GeoLocalització	5 días	vie 05/06/09	jue 11/06/09	Toni Tienda
<input type="checkbox"/> Tests	3,38 días	vie 12/06/09	mié 17/06/09	Toni Tienda
<input type="checkbox"/> Documentació	6,13 días	mié 17/06/09	jue 25/06/09	Toni Tienda

Figura 1 – Tasques del projecte amb duració i dates

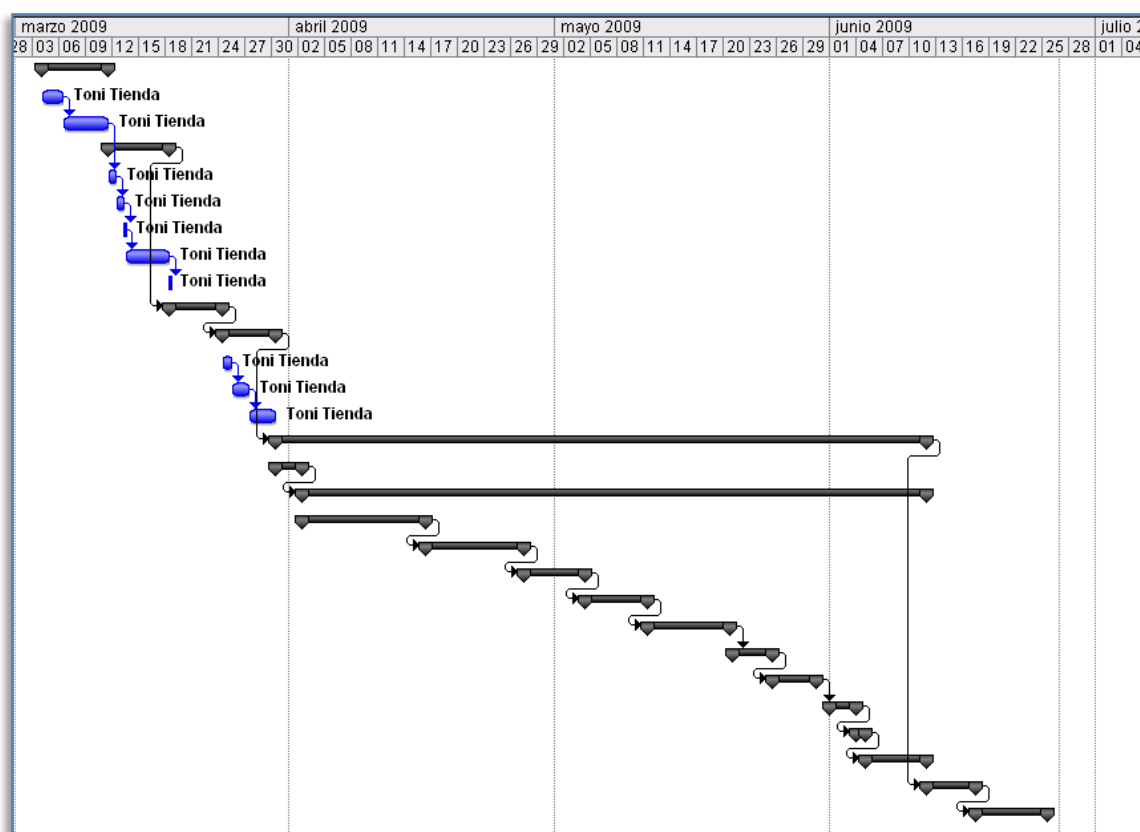


Figura 2 – Planificació de les tasques del projecte amb duració i dates

PRESSUPOST

EQUIPAMENT

Equipament	Cost
PC de Desenvolupament	
Windows XP Professional Edition.	114,84 €
IIS.	(inclòs al WINDOWS XP PROFESSIONAL EDITION)
Visual Studio 2008(academic version)	175 €
.NET Framework 3.5.	
Microsoft Office 2007(student)	93,83 €
SQL Server 2005	966 €
Servidor de producció	7,10 € /mes*
ASP 3.5 Hosting	inclòs
Servidor de base de dades SQL Server 2008.	inclòs
IIS7	inclòs

Total*	1.349,67 €
---------------	-------------------

*El preu real són 10\$ al mes. S'ha fet servir el canvi a data 27/06/2009 1\$ = 0.7104 €

** Preu mensual no inclòs.

PERSONAL

El sou del personal en tasques de desenvolupament és de 30€/hora. El sou en tasques d'anàlisi és de 40€/hora.

Tasca	Duració	Preu Hora	Total
Anàlisi Previ	5,5 dies		1.280 €
Anàlisi de requeriments	12 hores	40	480 €
Definició de funcionalitats	20 hores	40	800 €
Estudi de viabilitat	5 dies		1.120 €
Requeriments	4 hores	40	160 €
Avaluació de riscos	3 hores	40	120 €
Pressuposts	3 hores	40	120 €
Planificació	16 hores	40	640 €
Conclusions	2 hores	40	80 €
Anàlisi funcional	3,63 dies		800 €
Casos d'us	20 hores	40	800 €
Anàlisi tècnic	4,13 dies		1.060 €
Definició del model de dades	8 hores	40	320 €
Definició de l'estructura de l'aplicació	8 hores	40	320 €
Definició del model de classes	8 hores	40	320 €
Codificació	53,75 dies		11.030 €
BDD	2,88 dies		560 €
Creació de la BDD	8 hores	30	240 €
Creació dels procedures	6 hores	30	180 €
Aplicació	50,88 dies		9.840 €
Home	10 dies		1.800 €
Accés a enregistrament	4 hores	30	120 €
Login de Facebook	12 hores	30	360 €
Accés a la cerca	4 hores	30	120 €
Accés al calendari	12 hores	30	360 €
Accés al backoffice	4 hores	30	120 €
Visualització cites que han contestat	4 hores	30	120 €
Visualització persones que et poden interessar	4 hores	30	120 €

Visualització esdeveniments que poden interessar	4 hores	30	120 €
Visualització de cites demanades	4 hores	30	120 €
Visualització usuaris més valorats	4 hores	30	120 €
Visualització usuaris més participatius	4 hores	30	120 €
Perfil d'usuari	7,25 dies		1.380 €
Visualització de dades d'usuari	8 hores	30	240 €
Edició de dades d'usuari	16 hores	30	480 €
Canvi d'idioma	4 hores	30	120 €
Valorar usuari	6 hores	30	180 €
Configuració permisos Facebook	4 hores	30	120 €
Demandar cita	8 hores	30	240 €
Cerca d'usuaris	4,88 dies		960 €
Cerca BDD Local	6 hores	30	180 €
Recepció dades Facebook	12 hores	30	360 €
Ordenació dels resultats	8 hores	30	240 €
Accés a perfil d'usuari	4 hores	30	120 €
Accés a perfil d'usuari a Facebook	2 hores	30	60 €
Cerca d' esdeveniments	5,25 dies		960 €
Cerca BDD Local	6 hores	30	180 €
Recepció dades Facebook	12 hores	30	360 €
Ordenació dels resultats	8 hores	30	240 €
Accés a perfil d'usuari	4 hores	30	120 €
Accés a perfil de l'esdeveniment a Facebook	4 hores	30	120 €
Agenda	7,25 dies		1.380 €
Visualitzar calendari	24 hores	30	720 €
Visualitzar llista completa d'esdeveniments	4 hores	30	120 €
Visualitzar detall de l'esdeveniment	4 hores	30	120 €
Recepció dades Facebook	12 hores	30	360 €
Accés a perfil de l'esdeveniment a Facebook	2 hores	30	60 €
Home Backoffice	2,38 dies		480 €
Accés a gestió de coneixements	4 hores	30	120 €
Accés a traduccions	4 hores	30	120 €
Accés a gestió d'usuaris	4 hores	30	120 €
Visualització de missatges d'usuaris (suggeriments, ...)	4 hores	30	120 €
Gestió de coneixements	4,63 dies		720 €
Gestionar categories de coneixements	12 hores	30	360 €
Gestionar coneixements	12 hores	30	360 €
Gestió de traduccions	3 dies		720 €
Crear nou idioma	4 hores	30	120 €
Traduir missatges de la web	16 hores	30	480 €

Marcar idioma com a disponible	4 hores	30	120 €
Gestió d'usuaris	1 dia		240 €
Canviar rol de l'usuari	8 hores	30	240 €
GeoLocalització	5 dies		1.200 €
Càrrega de mapa des de GoogleMaps	4 hores	30	120 €
Càrrega mapa editable GoogleMaps	4 hores	30	120 €
GeoLocalització Inversa - Càrrega de ciutat per Lat,Long	16 hores	30	480 €
Creació funcions de càlcul de distàncies	6 hores	30	180 €
Aplicar Mapa a Perfil usuari, Esdeveniment	6 hores	30	180 €
Aplicar càlcul de distàncies a cerques	4 hores	30	120 €
Tests	3,38 dies		630 €
Home	4 hores	30	120 €
Perfil d'usuari	1 hora	30	30 €
Cerca d'usuaris	1 hora	30	30 €
Cerca d' esdeveniments	2 hores	30	60 €
Agenda	3 hores	30	90 €
Backoffice	1 hora	30	30 €
Coneixements	2 hores	30	60 €
Traduccions	2 hores	30	60 €
Gestió d'usuaris	1 hora	30	30 €
GeoLocalització	4 hores	30	120 €
Documentació	6,13 dies		1.400 €
Introducció	2 hores	40	80 €
Anàlisi	8 hores	40	320 €
Codificació	4 hores	40	160 €
Tests	3 hores	40	120 €
Unificació memòria i revisions	2 hores	40	80 €
Manual d'usuari	16 hores	40	640 €
Total			15.890 €

CONCLUSIONS

El preu total del projecte seria de:

1.349,67 € de l'equipament més 15.890€ de personal és a dir 17.239,67 €, més els 10\$ mensuals del servidor de producció.

Com que el client és la mateixa empresa que crearà el projecte, els quantitats apareixen a preu de cost.

CONCLUSIONS

Es tracta d'un projecte amb una envergadura assumible però amb un cert risc de cara al desenvolupament com és el temps d'investigació per tal de connectar amb **Facebook** i amb GoogleMaps. Hi ha força documentació al respecte i, per tant, no és preveuen grans problemes per tal d'assolir aquesta connectivitat.

Per tot això es considera que aquest projecte és totalment viable.

ANÀLISI FUNCIONAL

OBJECTIUS

Els objectius de l'anàlisi funcional de l'aplicació és el de definir detalladament les funcionalitats que ha d'implementar l'aplicació i a partir d'aquí definir concretament quines són les tasques que els desenvolupadors han de realitzar. En poques paraules, l'anàlisi funcional defineix **QUÈ** ha de permetre l'aplicació i per tant **QUÈ** s'ha de desenvolupar. En cap cas s'especificaran termes tècnics.

CASOS D'ÚS

En primer lloc farem la definició dels casos d'ús que són la representació gràfica de les funcionalitats que permet l'aplicació. Primer es defineixen uns actors i després unes funcions a executar i en última instància els diagrames de les funcionalitats per als actors definits. D'aquesta manera a part de definir **QUÈ** fa una funcionalitat i com se separa en diferents passes, es veurà **QUI** la pot realitzar.

ACTORS

Aquí definim els actors que interactuaran amb l'aplicació. Segons els requeriments del client s'ha trobat que hi haurà tres tipus d'usuari que interactuaran amb l'aplicació: usuari anònim, estàndard i administrador.

S'ha creat un actor genèric que contindrà les funcions comuns a tots els usuaris, així evitem la repetició i facilitem la comprensió dels casos d'ús. A continuació es defineixen en detall aquests tipus d'usuari:

USUARI GENÈRIC

És un usuari conceptual, que no té una existència real a la web. Es fa servir per a facilitar la comprensió dels casos d'us i per a evitar repeticions als diagrames. La resta d'actors amplien directa o indirectament les funcionalitats que les que té accés aquest actor.

USUARI ANÒNIM

És l'usuari que accedeix a la web sense estar enregistrat. Pot arribar des d'un enllaç extern, mitjançant una cerca a un buscador de Internet, ... Podrà fer servir les funcionalitats bàsiques.

USUARI ESTÀNDARD

És l'usuari normal de la web. És un usuari enregistrat que podrà fer servir totes les funcionalitats de la web excepte les de administració de la web.

USUARI ADMINISTRADOR

El l'usuari que pot realitzar totes les accions de manteniment de la web. Estén de l'usuari estàndard.

ÍNDEX DE CASOS D'ÚS

Id	Nom
1	Login
2	Enregistrar-se
3	Log out
4	Veure perfil a Facebook
5	Veure Esdeveniment en Facebook
6	Visualitzar Home
7	Visualitzar Índex
8	Llistar perfils interessants
9	Llistar esdeveniments interessants
10	Llistar pròxims esdeveniments
11	Llistar peticions de cites
12	Visualitzar localització
13	Editar localització
14	Visualitzar coneixements
15	Editar coneixements
16	Puntuar perfil

17	Modificar idioma de la web
18	Cerca d'usuaris
19	Cerca d'esdeveniments
20	Creació d'esdeveniment públic
21	Petició de cita
22	Visualització de detall d'esdeveniment
23	Visualització de calendari
24	Anar a mes següent
25	Anar a mes anterior
26	Anar a any següent
27	Anar a any anterior
28	Anar a data actual
29	Llistar categories
30	Llistar coneixements per categoria
31	Esborrar categoria
32	Crear categoria
33	Editar categoria
34	Esborrar coneixement
35	Crear coneixement
36	Editar coneixement
37	Buscar missatges web
38	Afegir nou missatge web
39	Editar missatge web
40	Guardar missatge web
41	Esborrar missatge web
42	Recupera missatge web
43	Guardar canvis dels missatges web
44	Refrescar cache dels missatges web
45	Buscar missatges de log

DEFINICIÓ DELS CASOS D'ÚS

1 LOGIN

El Login es el procés mitjançant el qual un usuari s'identifica a l'aplicació. És necessari per tal de d'obtenir les seves dades, conèixer quines accions ha rebut per part d'altres usuaris (petició de cites, acceptació o rebuig de cites proposades...) i per tal de proporcionar més o menys funcionalitats en funció del tipus d'usuari que sigui.

En concret, en aquesta aplicació el Login es farà a través del sistema d'autenticació de **Facebook** i, per tant, s'haurà d'obrir una finestra per tal de que l'usuari pugui realitzar aquesta acció.

Facebook, com a mesura de seguretat, demana als seus usuaris si volen que una aplicació pugui accedir a les seves dades. Per tant, en cas de que l'usuari estigui correctament autenticat a **Facebook** però no tingui l'aplicació configurada, s'haurà de mostrar aquest fet i permetre que l'usuari accepti que **Kookan** accedeixi a les seves dades.

2 ENREGISTRAR-SE

L'usuari es dona d'alta a l'aplicació per tal de fer servir les funcionalitats a les que només poden accedir els usuaris enregistrats.

3 LOG OUT

Permet tancar la sessió amb la aplicació. Molt útil quan s'està fent servir ordinadors d'accés públic.

4 VEURE PERFIL A FACEBOOK

Obre una finestra on es mostren les dades de l'usuari específic a **Facebook** (sempre i quan les polítiques de **Facebook** permetin a l'usuari actual veure aquestes dades).

5 VEURE ESDEVENIMENT EN FACEBOOK

Obre una finestra on es mostren les dades de l'esdeveniment específic a **Facebook** (sempre i quan les polítiques de **Facebook** permetin a l'usuari actual veure aquestes dades).

6 VISUALITZAR HOME

Qualsevol usuari pot accedir a la pàgina de Home des de la qual es poden realitzar diferents accions en funció del tipus d'usuari que estigui accedint.

7 VISUALITZAR INDEX

Els usuaris enregistrats tenen una pàgina d'inici personalitzada amb les dades generades a partir de les preferències de l'usuari que ha accedit a l'aplicació.

8 LLISTAR PERFILS INTERESSANTS

Carrega una llista d'usuaris que segons els coneixements que volen aprendre, els coneixements que volen adquirir i la distància a la que es troben poden ser interessants per a l'usuari que ha accedit a l'aplicació (a partir d'aquí **usuari actual**).

9 LLISTAR ESDEVENIMENTS INTERESSANTS

Carrega una llista d'esdeveniments públics que segons els coneixements que s'imparteixen i la distància a la que es troben, poden ser interessants per a l'usuari actual.

10 LLISTAR PRÒXIMS ESDEVENIMENTS

Carrega una llista amb els esdeveniments als que està apuntat l'usuari actual i que tindran lloc en breu.

11 LLISTAR PETICIONS DE CITES

Carrega una llista amb les peticions de cites que encara no ha contestat l'usuari actual.

12 VISUALTZAR LOCALITZACIÓ

Mostra la localització que té configurada el perfil del qual estem consultant les dades.

13 EDITAR LOCALITZACIÓ

Si el perfil consultat és el del propi usuari permet modificar la localització.

14 VISUALITZAR CONEIXEMENTS

Permet consultar la llista de coneixements del perfil que s'està consultant.

15 EDITAR CONEIXEMENTS

Si el perfil consultat és el del propi usuari, permet modificar aquesta llista.

16 PUNTUAR PERFIL

Permet puntuar un perfil per tal de facilitar la cerca d'usuaris.

17 MODIFICAR IDIOMA DE LA WEB

Canvia l'idioma en que es mostraran els missatges de la web.

18 CERCA D'USUARIS

Cerca un conjunt d'usuaris en funció de les dades que es facin servir com a filtre.

19 CERCA D'ESDEVENIMENTS

Cerca un conjunt d'esdeveniments en funció de les dades que es facin servir com a filtre.

20 CREACIÓ D'ESDEVENIMENT PUBLIC

Un usuari pot crear esdeveniments de lliure accés (que no vol dir gratuït). Aquests tipus d'esdeveniments són els que apareixeran a les llistes d'esdeveniments productes de cerques o de les llistes intel·ligents de la pàgina d'inici.

21 PETICIÓ DE CITA

Crea un esdeveniment privat per a l'usuari actual i l'usuari amb el qual s'està creant la cita.

22 VISUALITZACIÓ DE DETALL D'ESDEVENIMENT

Mostra totes les dades referents a un esdeveniment concret.

23 VISUALITZACIÓ DE CALENDARI

Carrega un calendari on apareixen els esdeveniments als que està apuntat l'usuari actual. Per defecte, el mes que es mostra és l'actual.

24 ANAR A MES SEGÜENT

Mou el calendari un mes endavant.

25 ANAR A MES ANTERIOR

Mou el calendari un mes enrere.

26 ANAR A ANY SEGÜENT

Mou el calendari un any endavant.

27 ANAR A ANY ANTERIOR

Mou el calendari un any enrere.

28 ANAR A DATA ACTUAL

Mou el calendari al mes actual.

29 LLISTAR CATEGORIES

Carrega la llista de categories de coneixements. Sense filtres.

30 LLISTAR CONEIXEMENTS PER CATEGORIA

Carrega la llista de coneixements en funció de la categoria seleccionada.

31 ESBORRAR CATEGORIA

Elimina una categoria.

32 CREAR CATEGORIA

Crea una nova categoria. S'ha d'especificar el nom en els idiomes dels que es disposen a la web.

33 EDITAR CATEGORIA

Permet editar el nom d'una categoria.

34 ESBORRAR CONEIXEMENT

Elimina un coneixement.

35 CREAR CONEIXEMENT

Crea un nou coneixement. S'ha d'especificar el nom en els idiomes dels que es disposen a la web.

36 EDITAR CONEIXEMENT

Permet editar el nom d'un coneixement.

37 BUSCAR MISSATGES WEB

Cerca els missatges (textos) que es mostren a la web.

38 AFEGIR NOU MISSATGE WEB

Crea un nou missatge per a la web.

39 EDITAR MISSATGE WEB

Permet editar un missatge de la web en un idioma en concret.

40 GUARDAR MISSATGE WEB

Guarda els canvis realitzats a un missatge.

41 ESBORRAR MISSATGE WEB

Elimina un missatge de la web.

42 RECUPERA MISSATGE WEB

Un missatge esborrar (i abans de guardar canvis) pot ser recuperat.

43 GUARDAR CANVIS DELS MISSATGES WEB

Guarda totes les dades modificades (creacions, modificacions i eliminacions) als missatges.

44 REFRESCAR CACHE DELS MISSATGES WEB

Actualitza el Cache i, per tant, publica els canvis realitzats als missatges.

45 BUSCAR MISSATGES DE LOG

Carrega la llista de missatges del log en funció d'un filtre.

DIAGRAMES DE CASOS D'ÚS

Es presentaran els diagrames de casos d'ús en funció dels usuaris. Cal aclarir que un usuari estàndard i anònim estenen a l'usuari genèric. Tanmateix un usuari administrador estén a l'usuari estàndard. Això vol dir que les funcionalitats que l'usuari estàndard pot fer servir no són només les que es defineixen al seu diagrama sinó també les que es defineixen al diagrama de l'usuari genèric.

Els casos d'ús sense numerar, són simples contenidors d'altres casos d'ús. Per exemple el mòdul Search conté els casos d'us: 18 - Cerca d'usuaris, 19 - Cerca d'esdeveniments, 29 - Llistar categories, 30 - Llistar coneixements per categoria.

A més es por observar que alguns casos d'ús es repeteixen. Això també s'ha fet així per poder analitzar quins casos d'ús es poden reutilitzar un cop es comenci el desenvolupament.

USUARI GENÈRIC

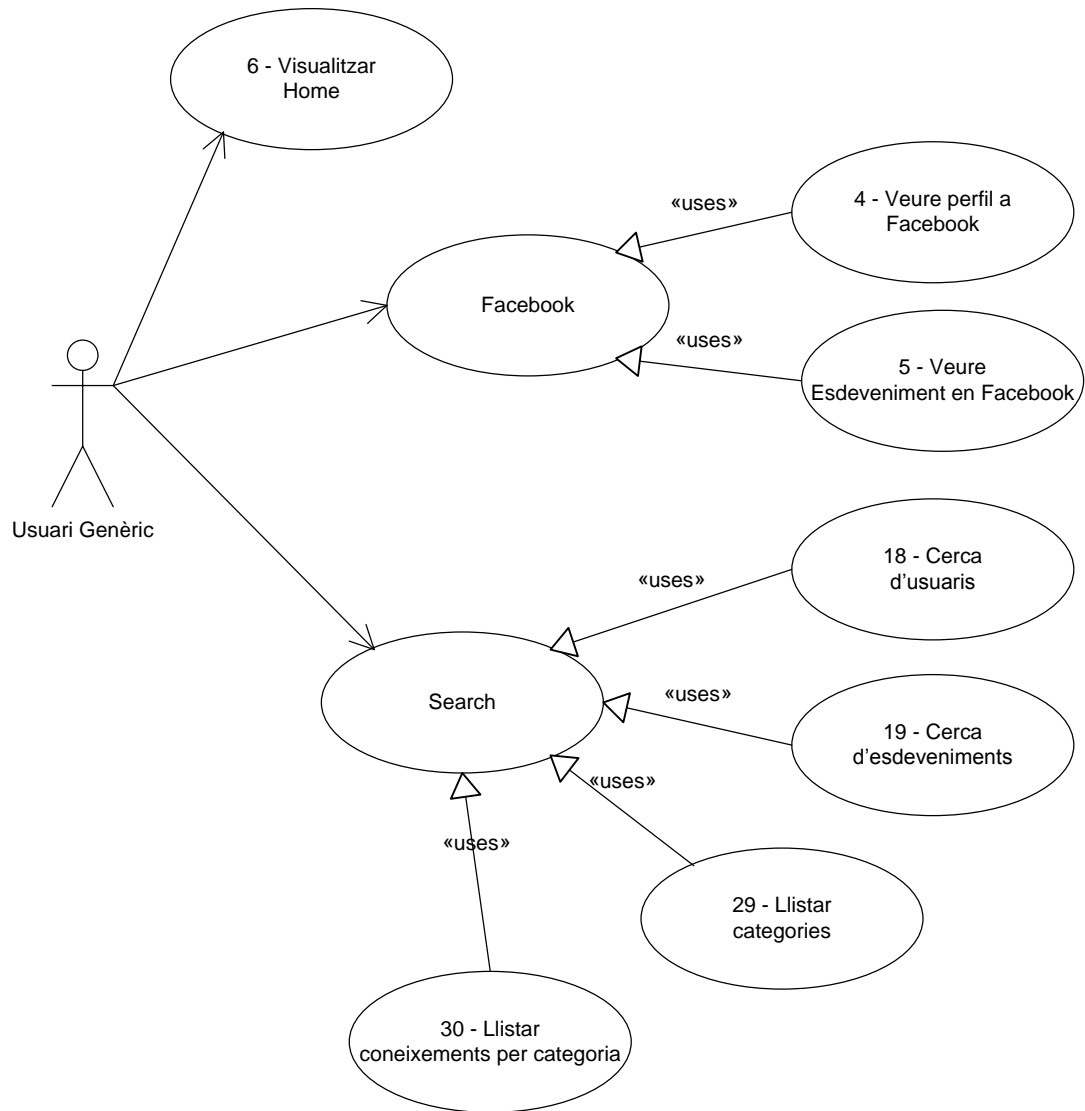


Figura 3 – Casos d'ús de l'usuari genèric

USUARI ANÒNIM

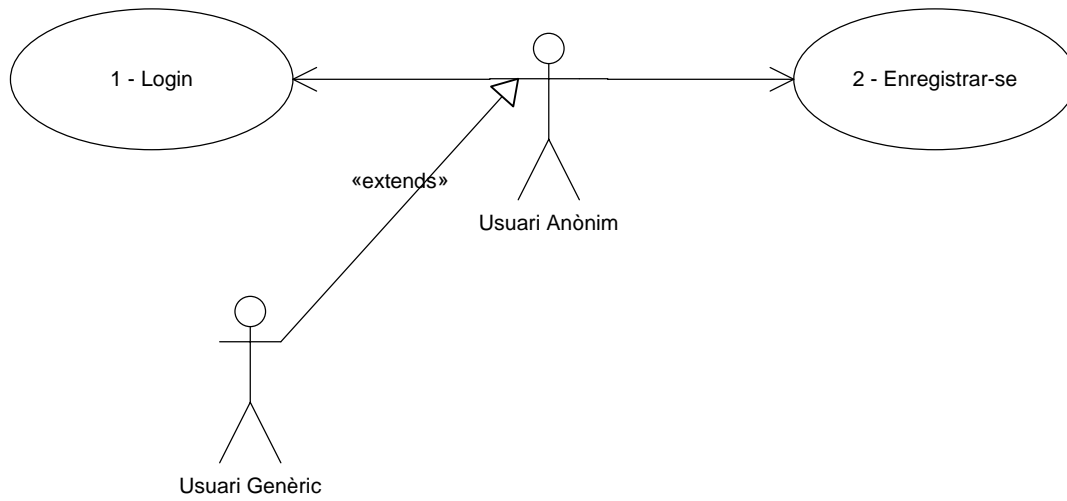


Figura 4 – Casos d'ús de l'usuari anònim

USUARI ESTÀNDARD

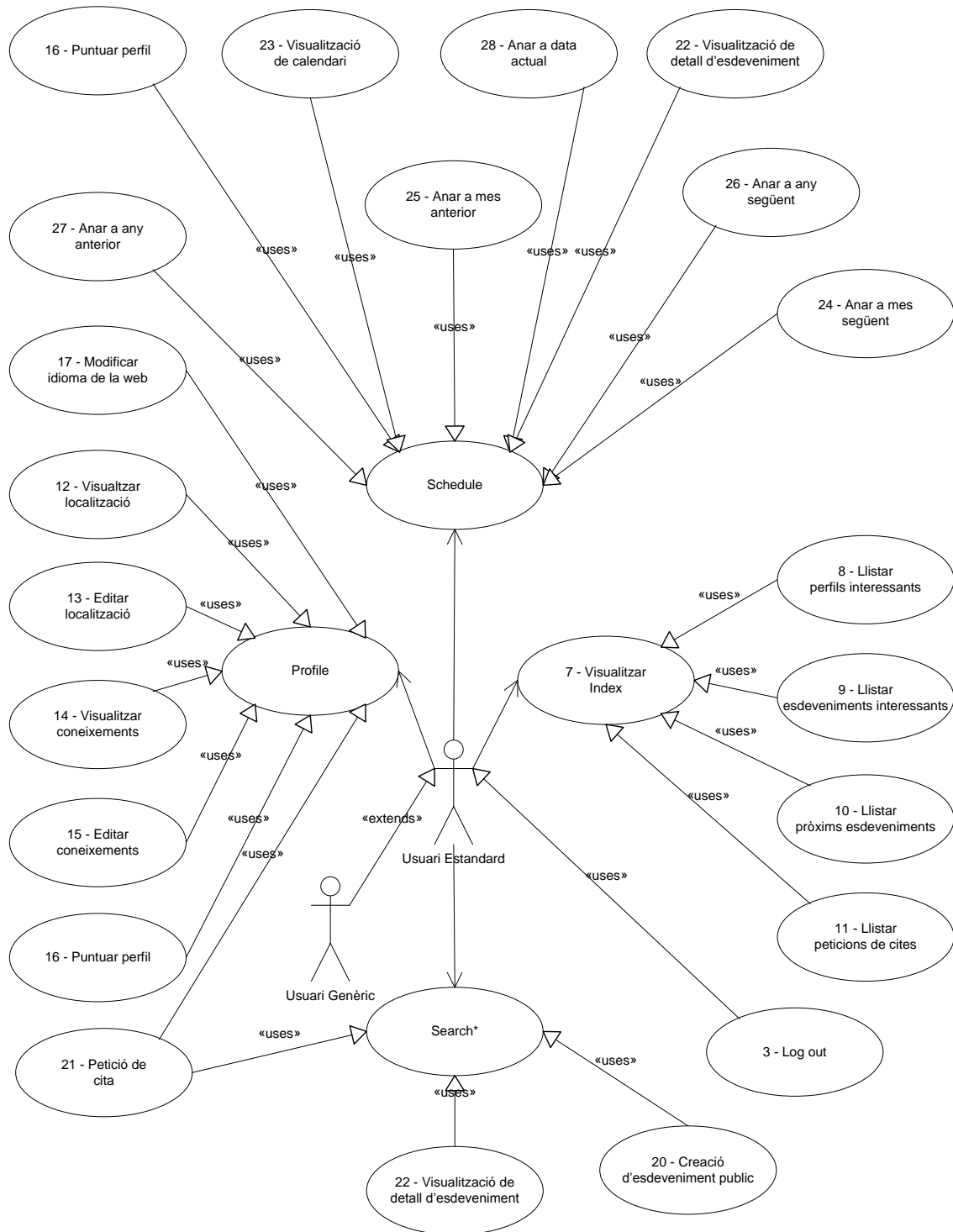


Figura 5 – Casos d'ús de l'usuari estàndard

*Estén el mòdul "Search" de l'usuari genèric.

USUARI ADMINISTRADOR

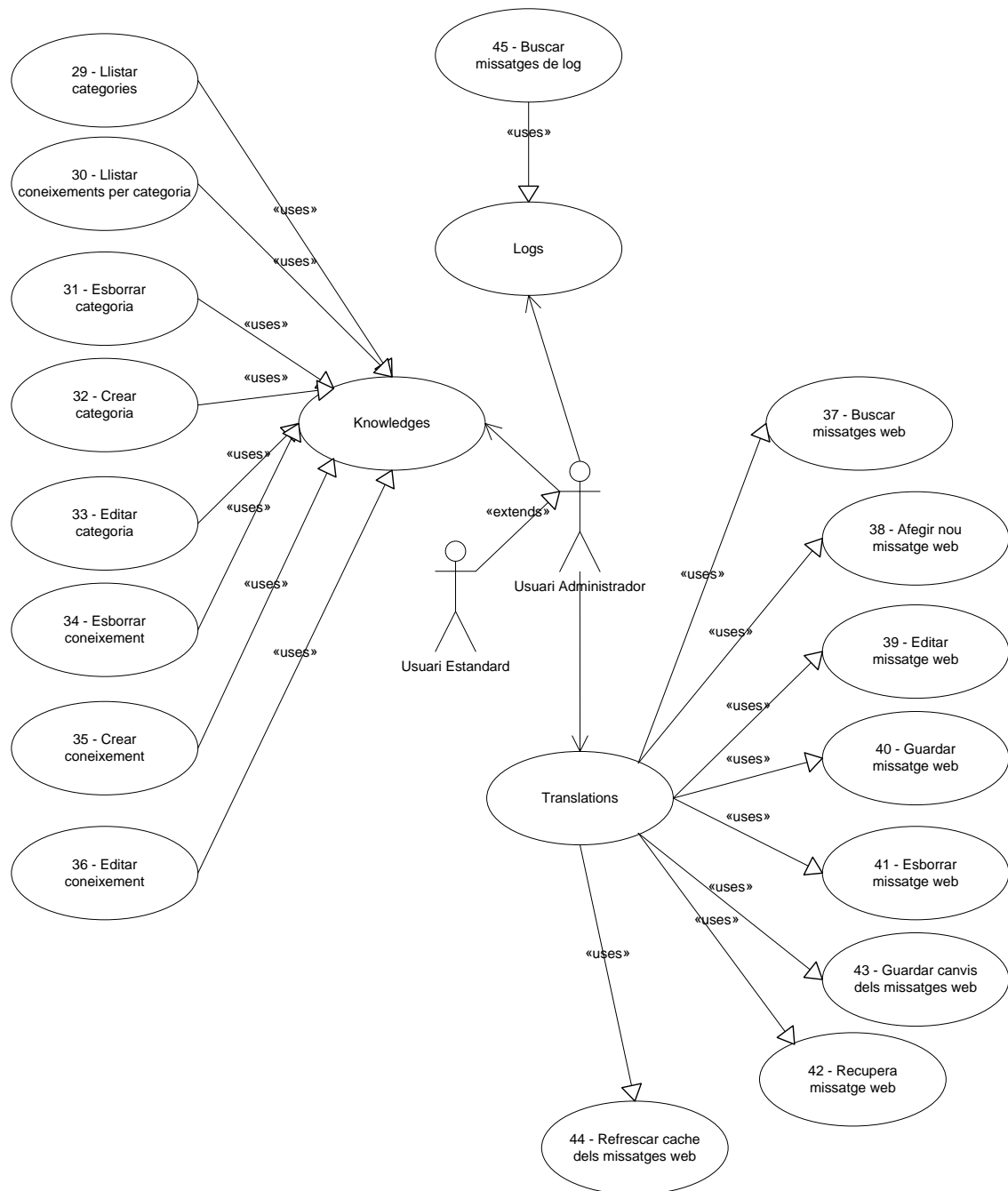


Figura 6 – Casos d'ús de l'usuari administrador

CONCEPTES TEÒRICS

INTRODUCCIÓ

Abans d'entrar de ple a l'anàlisi tècnic, es presentaran alguns dels conceptes que s'hi faran servir. Sense aquesta explicació és possible que alguns dels punts que s'exposen més endavant no quedi clars. Si el lector ja està familiaritzat amb aquests conceptes, es pot saltar aquest punt o bé consultar només els que no conegui.

FACEBOOK

Facebook es una de les xarxes socials més populars del món. Aquí hi ha alguns números trobat a la seva pàgina d'estadístiques[1].

Es calcula que té uns 200 milions d'usuaris a tot el món : gran nombre de possibles impactes de publicitat.

Cada dia entren uns 100 milions : "audiència" fidel.

Cada persona té una mitjana de 120 amics : gran difusió en màrqueting viral.

Els números de **Facebook** son força impressionants. Aquest fet ha impulsat a moltes companyies a fer servir aquesta xarxa per a promoure els seus negocis o inclús basar els seus negocis en aquesta xarxa. Hi ha moltes aplicacions creades específicament per a **Facebook** (com la que estem descrivint a aquest document) que basen els seus ingressos en la publicitat que hi contenen.

A Kookan es farà servir **Facebook** per a la gestió dels usuaris (sistema de login, fotos dels usuaris) i per a la creació i consulta dels esdeveniments.

GOOGLE MAPS

Com totes el que ofereix Google, GoogleMaps[2] ha estat un èxit des dels seus inicis. La seva funcionalitat bàsica és la de cercar localitzacions a un mapa però s'ha anat ampliant amb diferents mòduls com cerca de rutes, visualització d'imatges interessants de cada punt del mapa o el nou i impactant Street View[3].

Actualment és una de les aplicacions del seu tipus més utilitzades.

A **Kookan**, es farà servir GoogleMaps per a la localització dels usuaris o dels esdeveniments sense haver d'introduir els camps país, província, ciutat (amb les complicacions que allò porta com traduccions, creació de llistes de ciutats per província o Estat dependent del país, etc.).

API – APPLICATION PROGRAMMING INTERFACE

Una API, com el seu nom indica, és una interfície que proporcionen algunes aplicacions per tal que d'altres desenvolupadors puguin accedir a les seves funcionalitats d'una manera lliure però controlada. Lliure en el sentit que els programadors tenen accés a les dades de l'aplicació que proporciona la API i controlada en el sentit en què els propietaris de l'aplicació principal decideixen a quina informació permet l'accés aquesta API.

Desenvolupar una API pot semblar un error pel fet de posar les dades que conté la teva aplicació (amb el cost que això comporta) a tercers. Però de fet, mirant-ho a més llarg termini, una API augmenta la popularitat del teu projecte i fa que d'altres aplicacions es lliguin a la teva i per tant no facin servir les de la competència.

Fer servir una API (sobretot les API de grans companyies) sembla senzill en quant al desenvolupament donat que hi sòl haver molta documentació. Les dificultats apareixen normalment quan el desenvolupador de l'aplicació externa intenta realitzar algunes operacions (perquè el client així les vol) que no permet la API directament. Quan això succeeix és força difícil trobar un camí secundari per tal d'assolir el que es pretenia i si es troba, sol ser poc eficient.

Per altra banda, fer servir una API fa que s'hagi d'estar atent a possibles actualitzacions. Per una part, pot ser que una actualització de la API faci que algun dels mòduls de l'aplicació deixin de funcionar (per sort no se solen fer actualitzacions no compatibles amb versions anteriors) i per d'altra poden haver actualitzacions que permetin realitzar operacions que abans no es permetien i per tant pot proporcionar la possibilitat d'optimitzar una part de l'aplicació client de la API.

FRAMEWORK – MARC DE TREBALL

Un framework és literalment un marc de treball. En entorns de desenvolupament de software es tracta d'un conjunt d'eines que faciliten la programació de noves aplicacions.

Fa un temps relativament curt va sorgir aquesta paraula, però realment des de fa molt de temps existeixen aquests marcs de treball. Per posar un exemple, al principi dels ordinadors programables tot funcionava amb '0' i '1' directament. Es feien servir les famoses targetes perforades que no eres més que això, una codificació de '0' i '1' representat per forat o no-forat. Amb el temps van sorgir els llenguatges d'alt nivell que fan servir paraules més humanes per a realitzar el desenvolupament i després unes altres eines traduïa tot a '0' i '1'.

Els frameworks actuals són de més alt nivell (frameworks sobre frameworks) i fins i tot moltes empreses fan servir els seus propis marcs de treball per tal d'estalviar temps en de desenvolupament en els projectes. En aquests frameworks tenen operacions que solen ser comunes a tots els projectes i així s'estalvien desenvolupar-les una i una altra vegada (sembla una idea simple, però no totes les companyies reutilitzen els seus codis).

PROGRAMACIÓ PER CAPES

La programació a la web, des de bon començament va estar força associada al desenvolupament per capes.

Però, que són les capes?

Com a capa, s'entén un conjunt de classes/objectes que realitzen un tipus d'operació en concret i que es poden comunicar amb unes capes també en concret. Així es defineix un flux d'informació estructurat i es facilita la correcció d'errors i l'actualització d'una aplicació si algun element (Base de dades, tipus de presentació, ...) ha de ser modificat o renovat, ja que només implicaria una de les capes (o una part d'una de les capes) i la resta es mantindria com fins el moment. Així a part de fer més curt el desenvolupament, evitaria haver de testejar de nou totes les capes.

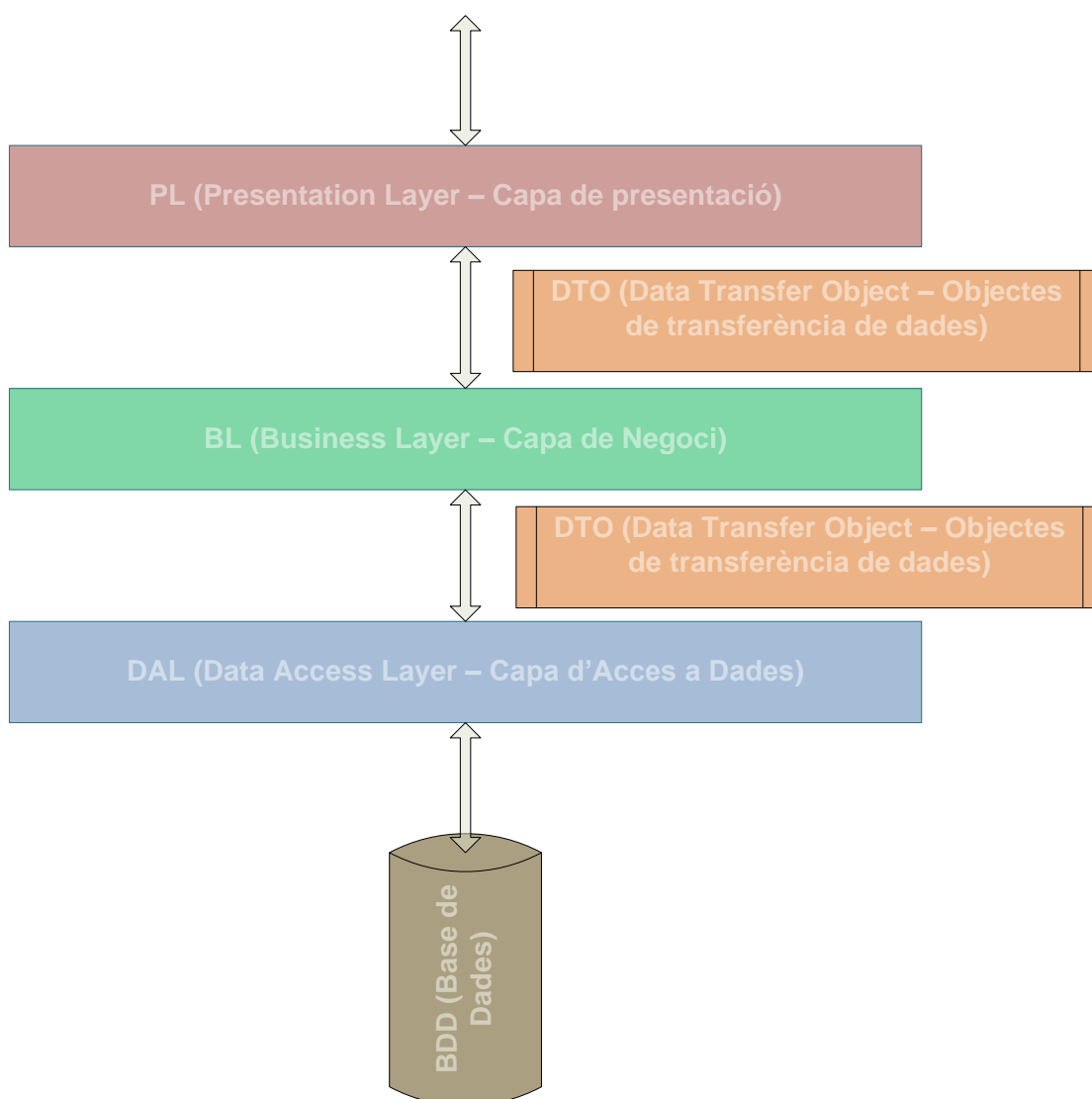


Figura 7 – Diagrama d'un desenvolupament per capes

Diagrama d'un desenvolupament per capes

Per posar un exemple de la utilitat de la programació per capes, imagina que haguéssim de crear una API per al nostre projecte. Si el projecte estigués desenvolupat per capes, una API seria una nova capa de presentació, que en comptes de ser una interfície gràfica per a un usuari seria una interfície que proporcionaria uns mètodes que permetrien a tercers a accedir a les dades de la nostra aplicació.

Si preguntes perquè no fer servir la BL com a API, la resposta es que seria convenient una capa addicional per tal de controlar quines dades permetem veure, per controlar possibles atacs fent servir la API, etc.

Si el desenvolupament no estigués ben estructurat, seria impossible ampliar un projecte per a crear una API i actualitzar-lo per a canviar la seva presentació seria una tasca llarga i complexa.

MVC – MODEL VIEW CONTROLLER

Un dels models per capes més populars és el MVC o Model Vista Controlador. En aquest model les classes es divideixen en tres capes (veure punt anterior 'Programació per capes' per entendre-ho completament):

Model : aquesta és la capa que conté les classes que fan referència a les dades. Normalment està dividida en dues capes més:

Data Layer: com hem vist abans és la capa que accedeix a les dades).

Business Layer: està formada per les classes que realitzen operacions de negoci (càlculs, validacions, etc).

Vista : és la capa de presentació clàssica. Fa d'interfície entre la aplicació i entitats externes (usuaris, o altres aplicacions).

Controlador: aquesta es la novetat del MVC. En el model per capes clàssic, una pàgina (capa de presentació) demanava unes dades a la capa de negoci i presentava el resultat. Però, què passa si durant la petició de dades hi ha algun error?, i si volem presentar una pàgina o una altra en funció dels resultats de les operacions executades?

Aquí entra el joc el Controlador. En aquest model en comptes de demanar pàgines (o vistes) s'executen accions i el controlador és el que defineix quina vista ha de retornar en funció dels resultats d'una operació.

Per exemple, volem crear una cerca, i volem que els resultats mostrats depenguin d'unes regles com les següents:

Si hi ha un error volem que es mostri un altre cop la pàgina de cerca amb l'error en vermell.

Si hi ha un resultat volem que es mostri el detall de l'únic element que s'ha trobat.

Si hi ha més d'un resultat, volem que es mostri la llista amb tots els resultats.

Així el controlador executaria la acció "Cerca" (crident a la capa de negoci) i en funció dels resultats mostraria una vista.

Amb aquest model, tindríem una vista de "Formulari de cerca", una de "Llista de resultats" i una de "Detall" que només s'encarregarien de mostrar les dades que pertocquen sense tenir cap lògica.

Amb aquest model, per tant, encara se separen més les diferents capes i faciliten les gestions, modificacions i reparacions de l'aplicació.

AOP – ASPECT ORIENTED PROGRAMMING

AOP o Programació Orientada a Aspectes és un nou paradigma de programació que s'ha donat com a conseqüència d'algunes mancances que s'han trobat a la ja clàssica Programació Orientada a Objectes.

En primer lloc explicarem a grans trets de què es tracta la programació orientada a objectes (a partir d'ara OOP de les sigles en anglès) i després explicarem quines millores aporta la programació orientada a aspectes (a partir d'ara AOP).

INTRODUCCIÓ A OOP

En un principi la programació era seqüencial i s'escriuen una instrucció darrera d'una altra. Això creava programes força difícils de seguir, de testejar i de mantenir. Es va pensar en models que fossin més fàcils a l'hora de dissenyar les aplicacions. El model que va tenir més força (i que es fa servir avui dia en la majoria de desenvolupaments comercials) és el OOP.

Aquest model proposa dissenyar i programar en base a classes i objectes. Cada classe seria una abstracció d'un tipus d'objecte que contindria un conjunt d'atributs i mètodes i un objecte seria una instància de la classe. Per posar un exemple:

Podríem crear la classe Alumne. Podria tenir els següents atributs:

- Nom
- Cognoms
- NIU
- DNI
- Temps d'estudi diari
- ...

I els següents mètodes:

- Estudiar (assignatura)
- Anar a classe (aula)
- Anar al bar()
- Fer examen (assignatura)

...

En el cas dels mètodes poden rebre paràmetres. Per exemple el mètode *estudiar* ha de rebre per paràmetre quina assignatura ha d'estudiar, el mètode *anar a classe* ha de rebre quina aula és, etc.

Amb aquesta classe podríem crear l'alumne *Pepe Martínez* amb les seves dades concretes que tindria els mateixos mètodes que qualsevol altre però amb dades diferents (diferents assignatures, diferents aules, diferent temps d'estudi diari, ...).

Aquest model presenta molts avantatges ja que si hi hagués algun error en una de les classes (per exemple la classe "Alumne") hauríem de corregir aquest mètode i tots els alumnes instanciats tindrien el mètode correcte. També evitaríem repetir codi, ja que la definició de la classe només estaria a un lloc. També afavoreix el seguiment del codi ja que les classes i els objectes tenen noms més o menys familiars, i el codi es fàcilment llegible. Com podria ser la línia: `pepe.Estudia(Càlcul);`

AVANTATGES D'AOP

Com hem vist el model OOP funciona molt bé. Llavors, per què cal crear un nou paradigma?

Amb el temps s'ha anat veient que durant la codificació d'una aplicació hi ha algunes necessitats que no es poden abstraure com un objecte i que s'han d'anar repetint durant tot el programa. Podem posar per exemple els tractaments d'errors (o excepcions) que es fan servir a la majoria de mètodes de l'aplicació.

Posem per cas els controladors que es veien al punt 'MVC – Model View Controller'. A cadascun dels mètodes dels controladors tindrem el cas de que si hi ha un error realitzem una acció en concret. A més tenim també que si l'aplicació és coherent, si hi ha una operació que dóna error mostrem a l'usuari una pàgina d'error (sempre la mateixa).

Això vol dir que tindrem el codi equivalent a :

```
Si hi ha un error :  
  Guarda el missatge d'error al Log de l'aplicació.  
  Mostra la pàgina d'error amb el missatge pertinent.
```

Aquest codi estaria afegit a cadascun dels mètodes dels controladors de la aplicació. Si en algun moment volguéssim canviar el comportament en cas d'error ho hauríem de fer a cadascun dels mètodes de cadascun dels controladors.

A aquests tipus de comportaments s'anomenen aspectes. No depenen d'un objecte en particular sinó que es tracta d'unes accions produïdes per uns fets que són comuns a tota la aplicació (o a tots els mètodes del mateix tipus de tota l'aplicació).

A **Kookan** no es fa servir ben bé l'AOP sinó una aproximació. Com si fos un afegit a OOP per a poder gestionar alguns aspectes de l'aplicació. L'AOP encara està una mica verd, encara que ja hi ha Frameworks que ho implementen.

LLENGUATGES D'SCRIPT

Un llenguatge d'script és un llenguatge de programació que no és compilat, sinó que és interpretat per una aplicació externa que s'anomena intèrpret. Hi ha molts llenguatges d'script com Javascript, Bash (per a comandes de Unix), PHP per a aplicacions web, ...

JAVASCRIPT

JavaScript es un llenguatge d'script basat en el concepte de prototip, implementat originàriament per Netscape Communications Corporation, i que va derivar en l'estàndard ECMAScript. És conegut sobretot pel seu ús en pàgines web, però també s'utilitza en altres aplicacions.

Malgrat el seu nom, JavaScript no deriva del llenguatge de programació Java, però tots dos compareixen una sintaxi similar inspirada en el llenguatge C. Semànticament, JavaScript és més pròxim als llenguatges Self i ActionScript (basat també en l'ECMAScript). El nom "JavaScript" és una marca registrada per Sun Microsystems.

Els scripts en Javascript fets servir a pàgines web són interpretats pels navegadors. Abans tots la gestió gràfica s'executava al servidor web que retornava una web molt estàtica. A l'actualitat (i cada cop més) es fa servir javascript per a modificar l'aspecte de la pàgina o fer algunes gestions abans d'enviar dades de nou al servidor. Així, per una part la resposta gràfica de l'aplicació web és més ràpida i interactiva i per altra part s'allibera al servidor de tasques purament visual.

AJAX - ASYNCHRONOUS JAVASCRIPT AND XML

AJAX són les sigles de Asynchronous Javascript And Xml, (JavaScript asíncron i XML), un conjunt de tecnologies que permeten actualitzar continguts web sense haver de tornar a carregar la pàgina. Això obre la porta a pàgines web molt interactives.

Ajax és asíncron en tant que les dades addicionals són demanades i carregades en un segon pla, sense interferir en la presentació i el comportament de la pàgina. Habitualment les funcions d'Ajax es criden des del llenguatge JavaScript. Les dades són adquirides usant l'objecte XMLHttpRequest, que està disponible als llenguatges d'script que corren en navegadors moderns, o també usant Remote Scripting en navegador que no suportin XMLHttpRequest. En qualsevol cas, no és necessari que el contingut asíncron sigui formatat en XML.

Ajax és multi plataforma i es pot usar en diversos sistemes operatius, arquitectures de computador i navegadors web, ja que es basa en estàndards oberts com JavaScript i DOM. Hi ha implementacions open source de frameworks i llibreries.

Una possible tècnica per programar en AJAX és mitjançant un motor AJAX (escrit en JavaScript i amagat en un marc ocult). Aquest motor és l'encarregat de mostrar la interfície gràfica i comunicar-se amb el servidor. Mitjançant AJAX, l'usuari no es connecta directament amb el servidor, sinó que primer ho fa amb el motor d'AJAX i aquest és el que fa la sol·licitud al servidor. Un cop això, el servidor retorna la sol·licitud que passa a través del motor i renderitza el resultat, de manera que mai es veurà la pàgina completament blanca com passa amb els models clàssics d'aplicacions web.

JQUERY

Es pot dir que JQuery és un Framework per a Javascript. Conté un conjunt de funcions que faciliten en gran mesura la gestió dels elements del DOM (elements que es troben a la pàgina com caixes de texts, taules, ...). El nom li ve de que permet fer cerques semblants a les que es fan servir a les consultes (queries en anglès) d'SQL. Es pot buscar un element o conjunt d'elements en funció del nom, identificador, tipus d'element, ... Un cop trobat aquest conjunt d'elements se li poden aplicar operacions de canvi d'aspecte, afegir o eliminar continguts, ...

L'avantatge del JQuery és que degut a que és programari lliure, té molts plugins o funcions afegides que realitzen tot tipus d'operacions. A més, si un d'aquest plugin no s'adequa totalment al que el desenvolupador necessita, pot ser modificat degut a que també solen ser programari lliure i obert a modificacions o actualitzacions.

GLOSARI DE TERMES

Plugin: es tracta d'uns petits programes que s'afegeixen a d'altres per a ampliar la seva funcionalitat. Per posar un exemple, el navegador Firefox pot fer servir diferents plugins per a augmentar les seves capacitats, per corregir textos, per a traduir les webs que s'estan visitant, ...

Query: un query es una consulta que es fa a una determinada font de dades. Normalment es tracta de consultes a servidors de bases de dades on s'especifica en un format concret què es vol, quines condicions han de complir els elements i en quin ordre es vol (entre d'altres coses més avançades). Per exemple, podria ser una sentència com: [Donam una llista de cotxes] [Que siguin de color vermell] [Ordenats per número de matrícula]. En aquest exemple la primera part es el que volem rebre, la segona la condició dels elements que volem rebre i la tercera en quin ordre els volem.

Encara que, com dèiem, el típic es fer servir queries per a accedir a dades d'un servidor de bases de dades, noves tecnologies fan servir aquest tipus de llenguatge per a seleccionar dades de llistes que s'estan processant (com fa LINQ, veure [4]) o per a seleccionar elements de la pàgina que s'està visualitzant (com va JQuery, veure [5]).

Rsvp: vol dir enviar una resposta a un esdeveniment. Prové de la frase en Francès "Répondez s'il vous plaît" (veure [6] per a una ampliació).

Usuari actual: sempre que a aquest document es fa referència a l'usuari actual, s'està referint a l'usuari que ha fet login a l'aplicació, és a dir, al usuari que està realitzant les accions.

ANÀLISI TÈCNIC

INTRODUCCIÓ

Abans d'endinsar-nos en el desenvolupament pròpiament dit, s'ha de fer un anàlisi més tècnic a partir de l'anàlisi funcional. Segons les necessitats del client, els requeriments funcionals i no funcionals, es decideix la millor estratègia de cara a desenvolupar l'aplicació amb garanties d'assolir completament el què el client espera.

Normalment es desenvolupa un projecte començant pels ciments. En cas dels projectes informàtics no és diferent. A partir dels anàlisis previs es decideix en primer lloc l'estructura que tindran les dades. A partir d'aquí creen les capes d'accés a dades i les capes de negoci. I finalment les capes de presentació (controladors i pàgines web).

Com que el disseny de les pàgines les sol realitzar un altre rol (dissenyadors i maquetadors), de vegades es fa paral·lelament al desenvolupament de les capes inferiors. Analistes tècnics i dissenyadors tenen les mateixes pautes a seguir (que les ha marcat el client junt amb els analistes funcionals) i per tant les conclusions han de ser les mateixes. D'aquesta manera també es permet al client participar en l'aspecte que tindrà la seva aplicació a més de demostrar que el projecte avança. El típic problema del desenvolupament "de baix a dalt" es què el client no veu res (les capes inferiors són les més importants però no permeten veure res gràfic) i té la impressió de que el seu projecte està molt enrederit encara que no sigui així.

MODEL DE DADES

En aquest apartat ens hem de parar un moment i recordar que el client volia crear una plataforma que contingués diverses aplicacions, encara que en un principi només existeixi **Kookan**. Amb aquest requeriment, s'ha optat per separar les bases de dades en dos de diferents. En primer lloc tindrem la base de dades **Tsoobame** (que és el nom de l'empresa client) i en segon lloc la pròpia base de dades **Kookan**.

També hem de fer referència al model de dades que té **Facebook** ja que algunes parts del model de dades estan fortament lligades a aquesta plataforma.

TSOOBAME

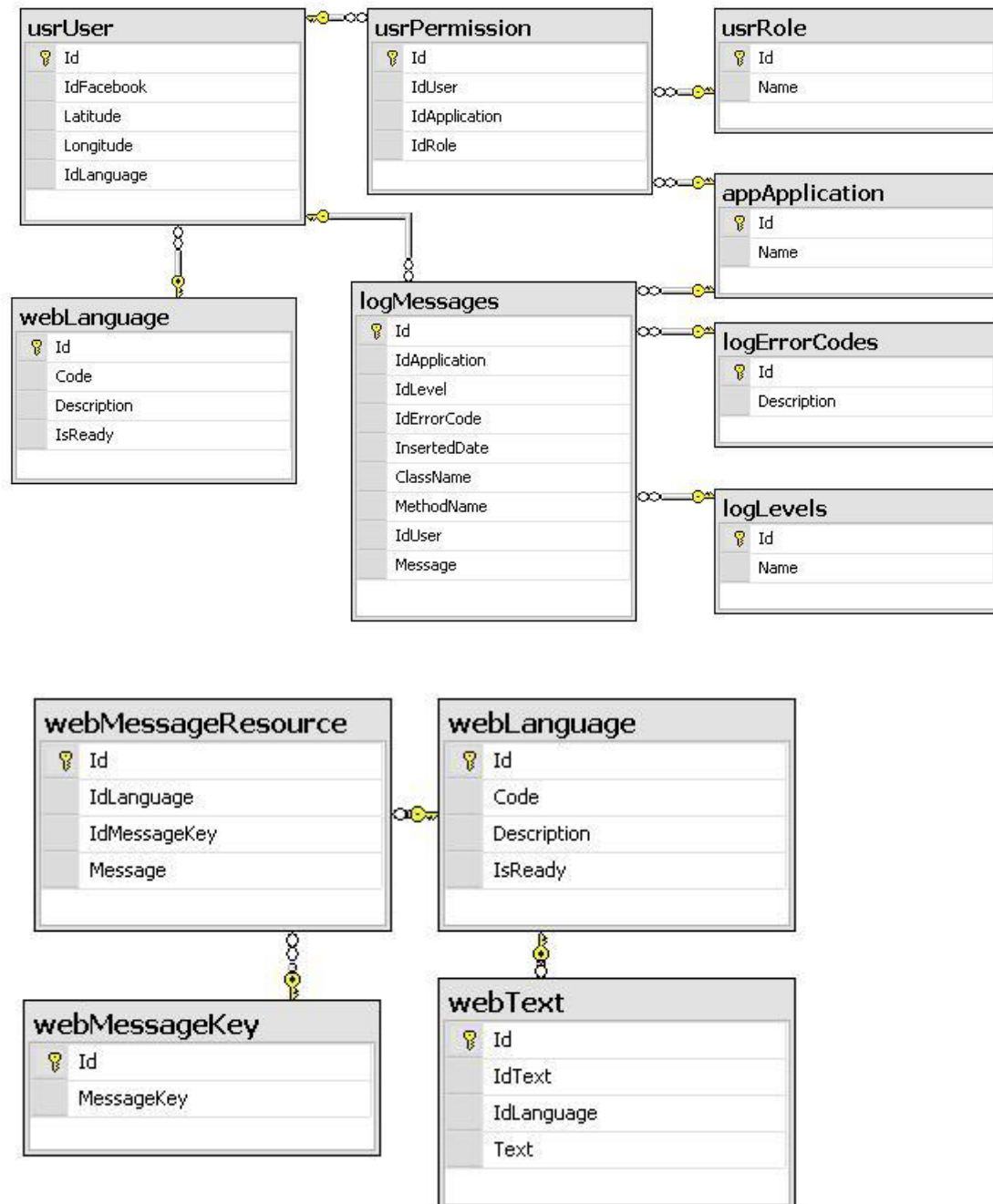


Figura 8 – Base de dades de Tsoobame

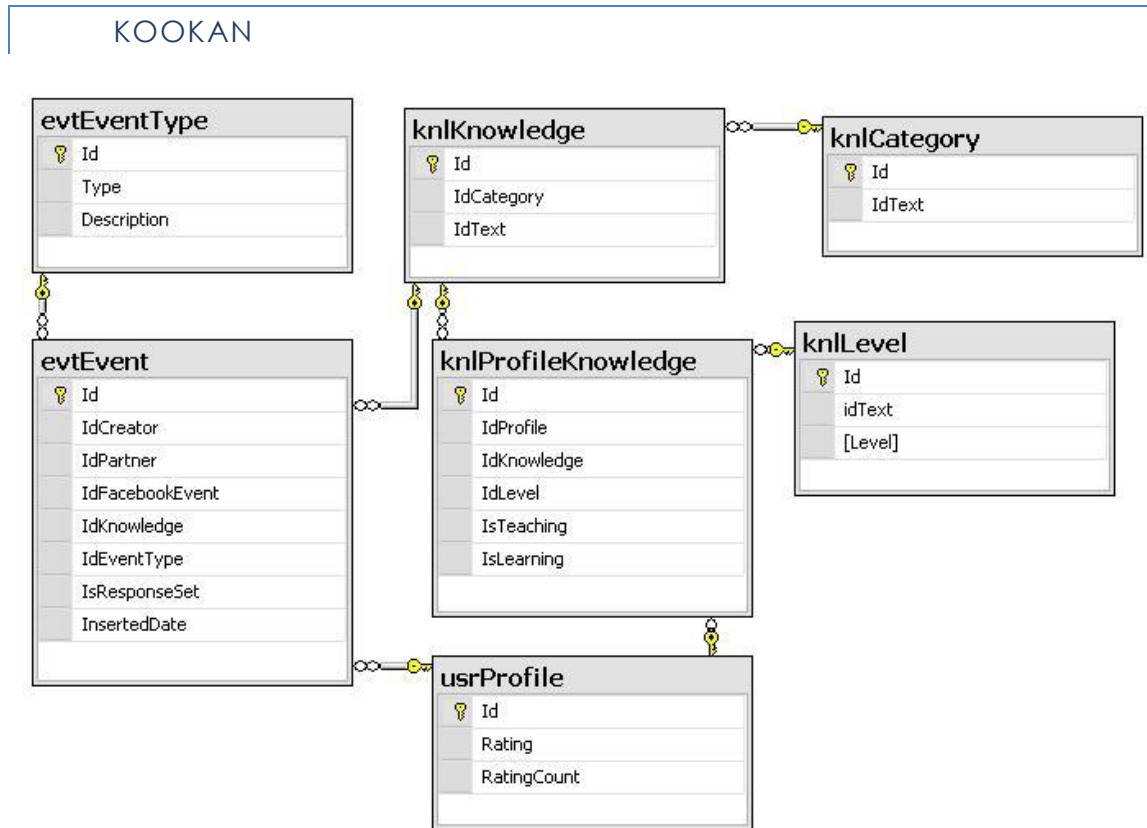


Figura 9 – Base de dades de Kookan

FACEBOOK

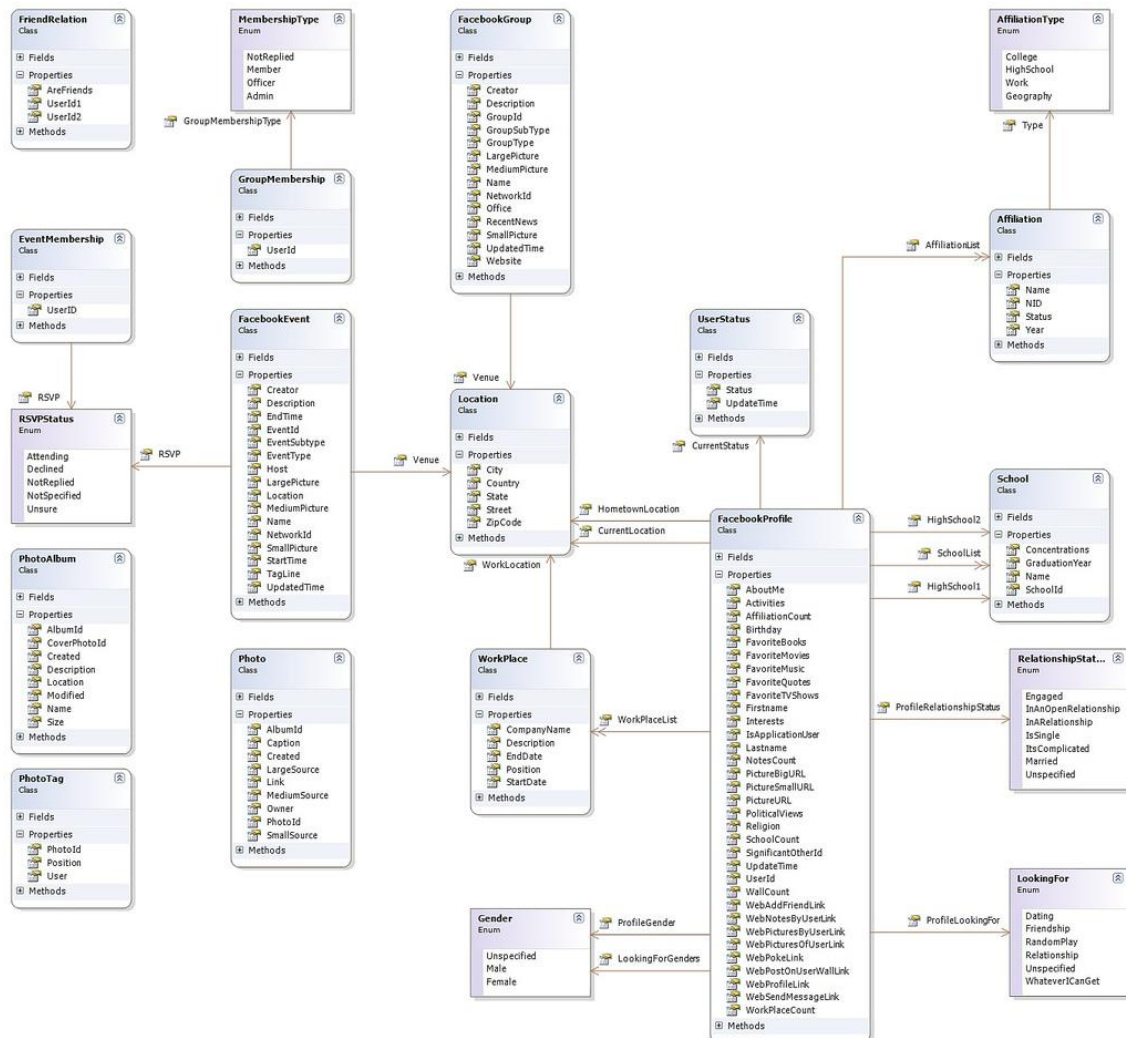


Figura 10 – Base de dades de Facebook

D'aquesta gran estructura de dades que té Facebook, a Kookan es fan servir les taules:

FacebookProfile, FacebookEvent, RSVPStatus, EventMembership i FriendRelation.

ESTRUCTURA DEL PROJECTE

Aquesta és l'estructura bàsica que tindrà el projecte per tal d'assolir les especificacions del client i per assegurar el desenvolupament en capes que, com hem vist amb anterioritat, tants beneficis aporta:

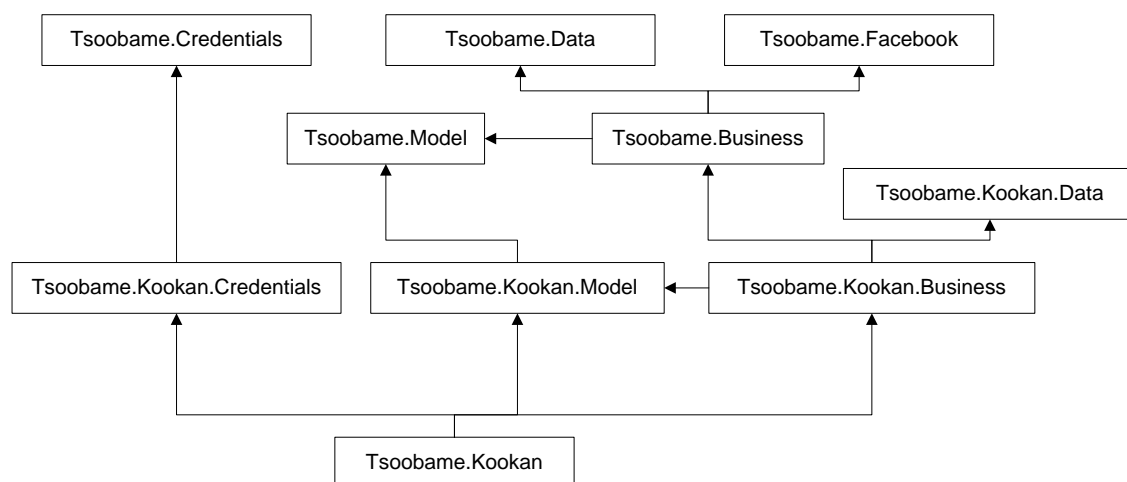


Figura 11 – Estructura del projecte

Quan diem bàsica vol dir que hi pot haver més mòduls afegits per poder ampliar funcionalitats (afegir utilitats, generació de logs, ...). Aquests mòduls però i la relació que hi ha entre ells haurà de seguir aquest diagrama. Ara passem a definir cadascun d'ells:

TSOOBAME.CREDENTIALS

És el mòdul encarregat de controlar els usuaris que accedeixen a la web principal. La resta d'aplicacions hauran de fer servir aquest mòdul directa o indirectament per a rebre les dades de l'usuari que ha fet login.

També controlarà que l'usuari que intenta realitzar una acció tingui els permisos necessaris per a fer-ho.

TSOOBAME.DATA

És la capa d'accés a dades de **Tsoobame**. Mitjançant aquest mòdul es podrà accedir a la Base de Dades.

TSOOBAME.FACEBOOK

Aquest mòdul es pot considerar una capa d'accés a dades, però no a la Base de dades de l'aplicació, sinó a la de **Facebook**.

TSOOBAME.MODEL

Aquí es definiran els objectes de transferència de dades o DTO (veure punt XXXXXXXX) de **Tsoobame**.

TSOOBAME.BUSINESS

Aquesta es la capa de negoci de **Tsoobame**. La resta d'aplicacions hauran d'accedir a les dades de **Tsoobame** i **Facebook** a través d'aquesta capa.

TSOOBAME.KOOKAN.DATA

Aquesta es la capa d'accés a dades específica de l'aplicació **Kookan**.

TSOOBAME.KOOKAN.BUSINESS

Amb aquest mòdul és farà l'accés a les dades necessàries per a fer funcionar **Kookan**. Tant accedirà a la capa de negoci de **Tsoobame** com a la capa de dades específica de **Kookan**.

TSOOBAME.KOOKAN.MODEL

En aquest mòdul es definiran els DTO de **Kookan**.

TSOOBAME.KOOKAN.CREDENTIALS

A partir d'aquest mòdul es faran les comprovacions d'usuaris extra per a **Kookan** (si són necessàries).

TSOOBAME.KOOKAN

Aquesta capa contindrà les capes de Controlador i de Vista de **Kookan**. Des d'aquesta capa només es podran fer servir els mòduls de credencials, de negoci de **Kookan** i els del Model també de **Kookan**.

CODIFICACIÓ

GESTIÓ DELS USUARIS

Els usuaris tenen tres nivell de dades.

Al nivell principal trobem les dades que **Facebook** ens aporta. No depenen de la nostra aplicació i sense un usuari de **Facebook** no es pot servir l'aplicació.

Al segon nivell trobem les dades que la plataforma **Tsoobame** aporta. Si un usuari que pertany a **Facebook** no té un usuari a **Tsoobame**, el crearem automàticament amb els valors per defecte.

Al tercer nivell es troba el usuari de l'aplicació secundària (**Kookan** en aquest cas). Les dades són les específiques per a aquesta aplicació. També si un usuari que pertany a **Facebook** i a **Tsoobame** no té usuari a l'aplicació específica es crearà un usuari automàticament amb els valors per defecte.

Les dades de **Facebook** i les de **Tsoobame** són accessibles des de qualsevol aplicació que pertanyi a la plataforma.

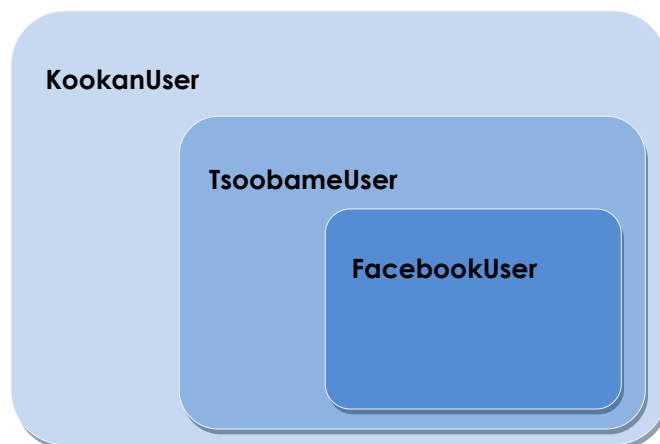


Figura 12 – Composició de l'usuari a Kookan

CONTROL D'EXCEPCIONS

Com que totes les execucions de l'aplicació web s'inicien amb els controlers i com que els controlers són la porta amb l'exterior, qualsevol Excepció del sistema produirà el mateix efecte:

En primer lloc s'afegirà una entrada al Log especificant l'usuari que està executant l'acció, la classe i el mètode on s'ha produït l'execució.

En segon lloc, es farà fluir l'excepció cap a la capa de presentació (sempre tornarà al controller que ha produït l'excepció indirectament).

En aquest punt es farà arribar a l'usuari un missatge d'excepció pertinent, en les diferents formes que s'espera el resultat de l'operació executada. Els resultats es poden esperar com a una pàgina HTML, com a un fragment d'HTML (com pot ser el cas d'una crida amb Ajax), com a una simple cadena de text (crida d'Ajax que esperi només un missatge de configuració).

Per a evitar repeticions de codis a cadascun dels mètodes, es fan servir uns atributs a cada capçalera. És una aproximació del que veiem als conceptes teòrics a l'apartat "Programació orientada a aspectes".

Posarem un exemple del codi LogOut, al controller Method:

```
[HandleViewError(APP, CLASS_NAME, "Logout")]
public ActionResult Logout()
{
    KookanUser user = KookanUser.Current;

    user.Logout();

    return View("Index");
}
```

L'atribut **HandleViewError** rep els paràmetres necessaris per a guardar els logs.

Existeixen els següents tipus de **HandleError**:

HandleAjaxError : retorna una part d'HTML informant de l'error.

HandleContentError :retorna un missatge informant de l'error.

HandleJsonError : retorna un objecte Json amb el missatge d'error.

HandleViewError : retorna una pàgina d'error per a informar a l'usuari.

CONTROL DE PERMISOS

Per al control de permisos existeixen dos nivells principals.

En primer lloc a nivell d'interfase. No s'ha de mostrar a l'usuari les opcions que no pot realitzar.

En segon lloc, s'ha de controlar que si un usuari intenta realitzar una operació no permesa, aquesta operació no tingui lloc.

En el control del primer nivell es fan servir condicions en funció del tipus d'usuari, de si les dades són del propi usuari o no, etc.

El segon nivell (més important), per tal d'evitar repetir i complicar el codi i donat que (com al punt anterior) totes les accions que du a terme un usuari és a través dels mètodes dels controlers, es faran servir atributs per tal de controlar els accessos a cadascun dels mètodes.

Posarem per cas l'accés al backoffice. En primer lloc, la opció no es mostrarà si l'usuari no és administrador:

```
<% RoleEnum rolActual = KookanUser.Current.IdRole; %>

<% if (rolActual == RoleEnum.Admin) { %>

<li class="ui-state-default ui-corner-top"><%=
Html.ActionLink(this.MessageResource("global.mainmenu.Backoffice"), "Index", "Backoffice")%></li>

<% } %>
```

En segon lloc, tanquem l'accés al controller que retorna la vista del backoffice als usuaris que no tinguin el nivell d'accés necessari:

```
[Authenticated(RoleEnum.Admin)]

[HandleViewError(APP, "BackofficeController", "Index")]

public ActionResult Index()

{

    return View();

}
```

SEGURETAT

Com a mètodes de seguretat (i a part del control de permisos) s'han realitzat les següents accions:

- 1) Validació de dades a servidor: així evitem que ens enviïn dades errònies per una via alternativa als formularis que es fan servir a la web.

- 2) Ús de LINQ per a les operacions de base de dades: mitjançant LINQ s'eviten molts dels possibles atacs fent servir injecció d'SQL o similars.
- 3) Mètodes de modificació de dades (creació, modificació o eliminació de dades) mitjançant enviament per POST: és un error típic posar un enllaç a una operació d'esborrar que funciona com la resta de links. Els buscadors com Google fan servir robots que naveguen per les webs per tal de capturar la informació dels continguts de cada web. És possible que (si no es té cura) aquest seguiment dels enllaços provoquin danys a les dades de l'aplicació.

CRIDES PER AJAX

Per a les accions on es fan servir Ajax (moltes a aquesta aplicació) s'ha creat una classe de javascript que ens ajuda a aquestes crides. Aquesta classe és la **tsoobame.controls.ajaxcontrol** i ofereix els següents mètodes:

exec (args, context, callback): executa una operació amb ajax, enviant els paràmetres mitjançant un objecte json definit al paràmetre args. Quan l'operació acaba, crida a la funció passada a través del paràmetre callback.

post (args, context, callback) : funciona de la mateixa manera que **exec** però enviant les dades per POST.

json (args, context, callback) : funciona de la mateixa manera que **exec** però el resultat que es retorna és una llista d'objectes json.

render (args, idcontainer) : és una drecera al mètode **exec** però en aquest cas, el resultat es renderitza a l'element del DOM amb id igual a idcontainer.

Així aconseguim que totes les crides passin per aquestes quatre funcions i tot el codi referent a l'execució de crides asíncrones estigui ben localitzat. D'aquesta manera és més fàcil i ràpid corregir errors o realitzar canvis si fos necessari a part de repetir el codi el mínim possible.

TEST

INTRODUCCIÓ

Al finalitzar el desenvolupament es realitzen els tests de l'aplicació. Hi ha diferents tipus de tests que s'han de fer per a assegurar el perfecte funcionament de l'aplicació desenvolupada.

- 1) Tests funcionals : es tracta de l'execució de tots els casos d'ús per a assegurar que l'aplicació no donarà errors als usuaris.
- 2) Tests de rendiment: a més d'assegurar que una aplicació no provoca errors, s'ha de comprovar que els resultats s'obtenen a una velocitat correcta, és a dir, que l'usuari no ha d'esperar més del compte per a obtenir els resultats que esperen.
- 3) Tests d'estres: aquests tipus de tests asseguren que l'aplicació no tindrà problemes o no provocarà errors quan un gran nombre d'usuaris la facin servir.

Ens els següents apartats veurem els resultats d'aquests tests.

RESULTATS

TESTS FUNCIONALS

Aquests són els tests funcionals que s'han portat a terme i els seus resultats:

Pàgina	Acció	Dades d'entrada	Sortida esperada	Estat
Home	Click a login		Pantalla de Facebook per a fer login.	Ok
Home	Click a registre		Pantalla de Facebook per a enregistrar-se	Ok
Home	Login a Facebook	Login i password de l'usuari de Facebook	Càrrega de la Home "Interna"	Ok
Home	Click al nom de l'usuari		l'usuari actual.	Ok
Home	Click a Profile		l'usuari actual.	Ok
Home	Click a Search		Càrrega de la pàgina de cerca	Ok
Home	Click a Schedule		Càrrega de la pàgina del calendari	Ok
Home	Click a Backoffice		Càrrega de la pàgina de backoffice	Ok
Profile	Click als settings de Facebook		Obertura de la pàgina de facebook per a gestionar els permisos de l'aplicació	Ok
Profile	Click al botó Facebook		Nova finestra amb el perfil de l'usuari de facebook	Ok
Profile	Click al botó edit		Càrrega del llistat de coneixements editable	Ok
Profile	Passar per sobre de les categories		queden destacats	Ok
Profile	Click a una categoria		Els coneixements queden filtrats per aquesta categoria	Ok
Profile	Click a netejar filtre		Es tornen a veure tots els coneixements	Ok
Profile	Canvi d'idioma		Canvia el idioma dels missatges de la web.	Ok
Profile	Votació a un usuari		Senyala el vot a l'usuari amb la puntuació seleccionada	Ok
Search	Click al botó de cerca		Càrrega d'usuaris i esdeveniments	Ok
Search	Click al botó de cerca	Coneixement seleccionat	oquest coneixement	Ok
Search	Click a crear esdeveniment public		Corregia el formulari de creació d'esdeveniment public	Ok
Search	Guardar esdeveniment públic		Error informant dels camps obligatoris	Ok
Search	Guardar esdeveniment públic	Coneixement seleccionat	Error informant dels camps obligatoris	Ok
Search	Guardar esdeveniment públic	Coneixement seleccionat	Error informant dels camps obligatoris	Ok
Search	Guardar esdeveniment públic	Data inici 5/07/2009	Error informant de rang de dates incorrecte.	Ok
Search	Guardar esdeveniment públic	Data FI 1/072009	Error informant de rang de dates incorrecte.	Ok

Search	Guardar esdeveniment públic	Coneixement seleccionat	Data inici 5/07/2009 9:00	Missatge informant de que l'esdeveniment s'ha guardat	Ok
Search	Click a Veure perfil d'un usuari	Data FI 5/072009 14:00		correctament.	Ok
Search	Click a crear cita			(usuari actual)	Ok
Search	Click a unir-se a un esdeveniment			és idem del públic)	Ok
Search	Click a veure esdeveniment			visible a l'schedule	Ok
Search	Click a veure perfil d'usuari a Facebook			Es mostra el detall de l'esdeveniment	Ok
Search	Click a esdeveniment a Facebook			Novo finestra amb el perfil de l'usuari de facebook	Ok
Schedule	Click a mes següent			Novo finestra amb la pàgina de l'esdeveniment de facebook	Ok
Schedule	Click a any següent			visualitzant	Ok
Schedule	Click a mes anterior			Carrega la planificació de l'any següent al que estem visualitzant	Ok
Schedule	Click a any anterior			Carrega la planificació del mes anterior al que estem visualitzant	Ok
Schedule	Click a mes actual			Carrega la planificació de l'any anterior al que estem visualitzant	Ok
Schedule	Click a "Veure tota la llista"			Carreta la planificació del mes i l'any actuals.	Ok
Schedule	Click a esdeveniment			Mostra tots els events per a aquest dia.	Ok
Schedule	Click a veure esdeveniment a			Carrega el detall de l'esdeveniment	Ok
Schedule	Click a perfil (assistents o creador de l'esdeveniment)			Novo finestra amb la pàgina de l'esdeveniment de facebook	Ok
Knowledges	Click a una categoria			Obre la pàgina del perfil en el que estem fent click	Ok
Knowledges	Click a esborrar categoria			Carrega els coneixements d'aquesta categoria	Ok
Knowledges	Click a afegir categoria			Prèvia confirmació per part de l'usuari esborra les categories i els coneixements associats	Ok
Knowledges	Click a guardar categoria			Mostra un formulari per a afegir una categoria	Ok
Knowledges	Click a guardar categoria			Missatge informant que els camps son obligatoris	Ok
Knowledges	Click a guardar categoria			Noms de categoria per a cada idioma	Ok
Knowledges	Click a guardar categoria			Guarda la categoria, tanca el formulari i recarrega la llista de categories	Ok

Knowledges	Click a esborrar coneixement		Prèvia confirmació per part de l'usuari esborra el coneixé	Ok
Knowledges	Click a afegir coneixement		Mostra un formulari per a afegir un coneixement	Ok
Knowledges	Click a guardar coneixement		Missatge informant que els camps son obligatoris	Ok
Knowledges	Click a guardar coneixement	Noms de coneixement per a codo	Guarda el coneixement, tanca el formulari i recorrejo la llista de coneixements	Ok
Translations	Click a cercar		Carrega els missatges de la web per als idiomes seleccio	Ok
Translations	Click a guardar tot		Guarda els canvis realitzats a la llista	Ok
Translations	Click a recarregar coche		Carrega els canvis realitzats per a que siguin visibles a la	Ok
Translations	Click a edit		Canvia el registre determinat per a posar-ho en mode ec	Ok
Translations	Click a guardar edició		els valor nous.	Ok
Translations	Click a cancelar edició		amb els valors antics.	Ok
Translations	Click a esborrar		Marca un registre com a eliminat	Ok
Translations	Click a recuperar		Cancela l'eliminació del registre.	Ok
Translations	Click a afegir		Afegeix un nou registre a la llista.	Ok
Logs	Click a cercar	Nivell : Info, Files : 100	o superior a Info	Ok
Logs	Click a cercar	Nivell : Debug, Files : 200	o superior a debug	Ok

TESTS DE RENDIMENT

Els tests de rendiment s'han anat fent mentre es provava l'aplicació. Normalment aquests tipus de tests es fan a servidors semblants al que serà el servidor de producció (un servidor a una intranet si el servidor final serà d'aquest tipus o un servidor a Internet si el servidor de producció estarà a Internet).

En el cas de **Kookan** encara que les proves s'han fet a una xarxa local quan el servidor de producció serà a Internet, es donen per correctes pel fet que la majoria de càrregues de dades fan referència a la base de dades de **Facebook** que sí es troba a Internet i per tant produeix la latència o temps d'espera que es trobarà a l'entorn de producció.

TESTS D'ESTRES

Els tests d'estres no s'han pogut portar a terme pel fet que no es poden crear usuaris de test per a afegir una gran quantitat de dades i per a simular l'accés simultani d'un gran nombre d'usuaris. Tampoc es poden crear esdeveniments a **Facebook** en un nombre suficientment alt com per a realitzar aquest tipus de proves pel perill de que tanquin l'aplicació **Kookan** per haver fet un ús indegut dels seus recursos.

CONCLUSIONS

OBJECTIUS

Per a aquesta entrega s'han cobert tots els objectius tret el de la localització mitjançant GoogleMaps. Degut a unes incompatibilitats, i per tal de permetre les proves amb el codi que s'entrega amb aquesta memòria s'ha desactivat aquesta funcionalitat, però no ha estat eliminada del codi.

LÍNIES OBERTES

A mesura que es desenvolupava l'aplicació i sobretot en la fase de test s'han anotat algunes millores que farien més atractiva l'aplicació. Passem a detallar-les:

-Fòrum : un fòrum per tal que els usuaris es puguin comunicar entre ells i publicar cursos o les xerrades que es facin en format text.

-Skins de la web: personalització del disseny de la web per a cada usuari. Com que el disseny de la web s'ha realitzat seguint els patrons JQueryUI es podria assolir aquesta tasca sense dificultat.

- Frame visible des de **Facebook** : es podria crear una pàgina per a ser carregada directament dins de la interfase de **Facebook** per fer més senzilla la difusió de l'aplicació.

CONCLUSIONS FINALS

El projecte ha presentat algunes dificultats degut a l'ús d'eines i d'APIs externes. Algunes tasques de càrrega de dades des de **Facebook** s'han complicat degut al tipus de mètodes que ofería la seva API i a la impossibilitat de recuperar dades de dues taules diferents en una sola crida.

Aquestes complexitats també han fet millorar el procés de desenvolupament i han ofert la possibilitat de crear unes eines i d'aprendre unes noves tècniques de conversió de dades.

BIBLIOGRAFIA

BIBLIOGRAFIA ESMENTADA A LA MEMÒRIA

- [1] <http://www.facebook.com/press/info.php?statistics>
- [2] <http://maps.google.com/>
- [3] <http://maps.google.es/help/maps/streetview/>
- [4] http://es.wikipedia.org/wiki/Language_Integrated_Query
- [5] <http://es.wikipedia.org/wiki/JQuery>
- [6] <http://en.wiktionary.org/wiki/rsvp>

BIBLIOGRAFIA CONSULTADA

ASP .NET

<http://www.asp.net/>

<http://www.asp.net/mvc/>

<http://weblogs.asp.net/scottgu/archive/2009/03/10/free-asp-net-mvc-ebook-tutorial.aspx>

JAVASCRIPT

<http://msdn.microsoft.com/en-us/magazine/cc163419.aspx>

JQUERY

http://docs.jquery.com/Main_Page

<http://jqueryui.com/demos/>

FACEBOOK API

<http://developers.facebook.com/>

<http://wiki.developers.facebook.com/index.php/API>

GOOGLEMAPS API

<http://code.google.com/intl/es-ES/apis/maps/documentation/>

AGRAÏMENTS

Volia agrair a la gent que conscient o inconscientment m'han ajudat a realitzar aquest projecte:

En primer lloc gràcies a la directora del projecte per donar-me alguns consells per a la realització de la memòria.

En segon lloc vull agrair als companys que he anat trobant a les companyies en les que he treballat i que m'han ajudat a aprendre com estructurar un projecte. Espero haver seguit bé els seus consells. En especial gràcies a Daniel Espanya i Ignacio Cougil.

Gràcies als amics que m'han animat quan passava per moments d'estres.

I gràcies a les persones que , m'han ajudat sense saber-ho creant manuals o responent a preguntes a fòrums i que sense ells m'hagués estat molt difícil superar algunes dificultats que he anat trobant durant el desenvolupament.

ANNEXOS

REQUERIMENTS DEL CLIENT

L'aplicació hauria d'aprofitar les tecnologies actuals d'Internet per tal de poder arribar al màxim nombre d'usuaris. D'aquesta manera es rebaixarien les despeses de publicitat als medis. Potser es podria integrar amb Facebook per aprofitar la popularitat d'aquesta xarxa social.

També per aquest motiu, l'aplicació hauria de ser multi idioma. L'original hauria d'estar en anglès (de les traduccions ens encarreguem nosaltres).

Per tal de que l'aplicació sigui atractiva pels usuaris, hauria de tenir una alta interactivitat. Una bona velocitat de resposta, y sensació de dinamisme. Quan una pàgina és molt estàtica l'usuari té la sensació de lentitud i fa que no facin servir tant l'aplicació. El disseny hauria de ser simple (no massa recarregat) per a que tot sigui senzill d'entendre i fàcil de manegar.

Els usuaris poden definir tant el que poden ensenyar com el que volen aprendre. Per tant, son professors i alumnes alhora. Per tal de cercar altres usuaris haurien de poder especificar també un horari de disponibilitat que fos definit a l'hora de donar-se d'alta. També s'hauria de poder especificar la localització (país, província, ciutat) per tal de buscar usuaris propers.

Com que es un projecte de voluntariat, els usuaris es poden donar de baixa en qualsevol moment.

La cerca d'usuaris ha de ser senzilla. Evitar en el possible l'ús de filtres de text i fer servir més l'ús de filtres per clic (tipus llista de categories).

Per poder fer més efectives les cerques, s'ha de poder puntuar els usuaris

Ha d'haver una secció on apareguin els usuaris que possiblement interessin a l'usuari que fa servir l'aplicació (tipus "People you may know" de Facebook).

Els coneixements que es poden demanar/oferir han d'estar categoritzats (no han de ser text lliure) per a facilitar la tasca de cerca. Per tant, s'ha d'oferir una eina de BackOffice per a mantenir aquests coneixements.

S'hauria de poder mantenir una agenda amb l'horari de l'usuari. Podria veure amb qui té cites, a quines hores té lliure, ...

Un usuari a part de demanar cites, pot oferir xerrades a les que els usuaris es poden apuntar lliurement. Aquestes xerrades han poder trobar-se pel tipus de coneixement que s'hi ofereix i per localització.

Aquesta és la primera aplicació de moltes que es volen fer en el futur. La gestió dels usuaris hauria de fer-se dins d'una part global que contindria la resta d'aplicacions. Cadascuna d'aquestes mantindria les seves dades concretes.

ANÀLISI I SÍNTESI DELS REQUERIMENTS DEL CLIENT

A partir del document que ens va enviar al client (cita del punt superior) vam sintetitzar les idees i vam aportar-ne de noves en un document que va ser acceptat pel client amb data 03/03/2009. Aquí es cita aquest document:

L'aplicació es desenvoluparà en dos mòduls diferenciats. Un tindrà un concepte global (gestió dels usuaris i les dades generals d'aquest) i l'altra serà la pròpia **Kookan**.

La gestió d'usuaris es farà fent servir la xarxa social **Facebook**. Per tant la web **Kookan** no tindrà un procés d'enregistrament pròpiament dit, sinó un Login amb usuari de Facebook i una pàgina de configuració de perfil. Això vol dir que qualsevol usuari de **Facebook** podrà accedir a **Kookan**.

La gestió interna de usuaris de **Kookan** (definir qui és administrador i qui no) es farà a una eina al BackOffice.

El sistema de localització, per tal de fer-lo més flexible i per tal de possibilitar la cerca per distància (usuaris/esdeveniments a 20kms com a màxim) és farà mitjançant latitud/longitud amb ajuda de la eina que ofereix **GoogleMaps**.

Els usuaris de **Kookan** podran especificar al seu perfil: coneixements que volen adquirir/ensenyar, horari de disponibilitat, localització, idioma en què volen veure la web.

La gestió de les cites i les classes és farà mitjançant els esdeveniments de **Facebook**. S'especificaran horari de la classe, localització, coneixement que es pretén intercanviar/presentar.

Els usuaris es podran donar de baixa en qualsevol moment.

A les pàgines de cerca els filtres es faran mitjançant llistes de selecció o enllaços sempre que sigui possible (evitant les caixes de text).

Possibilitat de puntuar els usuaris per tal de ordenar les cerques en funció dels usuaris més populars.

A la pàgina d'inici es presentaran les següents llistes, resultats de cerques automatitzades: usuaris més populars, peticions de cites, classes que et poden interessar, usuaris que et poden interessar.

Els coneixements estaran categoritzats i seran gestionats des d'una eina de BackOffice.

Hi haurà una pàgina d'agenda des d'on es podran consultar els Esdeveniments (classes i intercanvis) que té l'usuari. Es veuran en format calendari per tal de fer més senzilla la seva visualització.

