

Treball fi de carrera

Memòria del projecte

Gestió d'inscripcions del Centre Excursionista Olot

Data entrega: 09/01/2014

Estudiant: Xavier Terradellas

Consultor: Ignasi Lorente Puchades

Dedicat a tota la meva família i a la Laura per deixar-me el temps necessari per treballar en el projecte, i a tots els companys que m'han donat idees per la seva realització.

Resum

Actualment moltes de les aplicacions web es realitzen utilitzant tecnologies de codi obert. Una de les més famoses i àmpliament utilitzades a nivell de servidor és LAMP, que engloba Linux, Apache, Mysql i PHP. Algunes de les característiques més destacables del LAMP és el seu baix cost d'adquisició i la disponibilitat en múltiples plataformes. Entorn aquesta tecnologia, han aparegut diferents llibreries i *frameworks* per desenvolupar aplicacions de forma ràpida i senzilla.

La idea d'aquest projecte és doble. D'una banda desenvolupar una plataforma per permetre al Centre Excursionista Olot (CEO), centralitzar totes les inscripcions a les diferents excursions que organitzen en una sola plataforma. L'altre és estudiar quines possibilitats ofereix la tecnologia LAMP i estudiar més a fons les possibilitats que ofereix per desenvolupar software a nivell professional.

Pel que fa a la tecnologia, aquest projecte utilitza principalment el *framework* CodeIgniter, que permet una millor estructuració del codi gràcies al patró Model Vista Controlador (MVC) i aporta un conjunt de llibreries per fer més fàcils les tasques de desenvolupament d'una aplicació web. Tot i així s'han incorporat altres llibreries externes com ara Doctrine ORM, Pear, Bootstrap, entre d'altres, a fi d'incorporar noves funcionalitats, i millorar-ne algunes de les ja existents.

El sistema complert està pensat per desplegar-se en un servidor LAMP i està format per les següents parts:

- Web de "Frontend", és la part pública de la web. Es tracta d'una web senzilla i de fàcil utilització, per tal que els clients es puguin inscriure a les diferents excursions que organitza el CEO. Incorpora tot el procés de pagament bancari i la impressió de butlletes, així com una gestió d'usuaris completa. Una de les característiques més destacables d'aquesta part és que està implementada pensant amb els dispositius mòbils.
- Web de "Backend", és la part que gestionarà el CEO i permet entre d'altres, la gestió dels diferents tipus d'excursions, la gestió d'usuaris i la visualització de les inscripcions i els pagaments efectuats a través de la passarel·la de pagament bancari.

Índex de continguts

Introducció.....	6
Justificació i context del projecte.....	6
La planificació del projecte.....	6
Abast del projecte.....	6
Requisits i característiques del projecte.....	7
Criteris d'acceptació del projecte.....	7
Tasques del projecte.....	7
Anàlisi i planificació.....	7
Identificar i analitzar els requisits.....	7
Anàlisi i disseny.....	7
Construcció del sistema.....	8
Documentació.....	9
Planificació temporal.....	9
Fites importants.....	9
Pla de gestió de riscos.....	10
Recursos utilitzats i pressupost orientatiu.....	10
Situació del mercat sobre les webs d'inscripcions.....	11
Identificar i analitzar els requisits.....	12
Casos d'ús.....	13
Actors del sistema.....	14
Cas d'ús C01: Seleccionar excursió.....	14
Cas d'ús C02: Crear nou usuari.....	14
Cas d'ús C03: Entrar al sistema.....	14
Cas d'ús C04: Demanar paraula de pas oblidada.....	15
Cas d'ús C05: Inscriure's a una excursió.....	15
Cas d'ús C06: Modificar dades personals.....	15
Cas d'ús C07: Comprovar estat meves inscripcions.....	16
Cas d'ús C08: Realitzar pagament electrònic.....	16
Cas d'ús C09: Validar pagament electrònic.....	16
Cas d'ús GEA01: Gestionar excursions.....	17
Cas d'ús GEA02: Veure apuntats.....	17
Cas d'ús GEA03: Extreure llistat d'apuntats.....	18
Cas d'ús GEA04: Gestió d'usuaris.....	18
Prototip d'interfície de l'aplicació.....	19
Prototip de l'administració o Backend.....	19
Prototip del Frontend.....	22
Arquitectura inicial del sistema.....	25
Anàlisi i disseny	27
Diagrama estàtic de classes.....	27
Diagrama de disseny de l'aplicació en MVC.....	28
Diagrama model relacional.....	29
Diagrama de seqüència del pagament bancari.....	30
Anàlisi de l'arquitectura del sistema.....	31
Punt de vista físic.....	31
Punt de vista lògic.....	32
Diagrama de components.....	32
Decisions tecnològiques.....	33
Implementació.....	34
Consideracions generals.....	34
Fase 1.....	34
Seguiment.....	34
Seguiment dels riscos.....	35
Refinament de les decisions de disseny.....	35
Fase 2.....	36

Seguiment.....	36
Seguiment dels riscos.....	36
Refinament del diagrama estàtic de classes.....	37
Refinament del diagrama model relacional.....	38
Fase 3.....	39
Seguiment.....	39
Refinament del diagrama estàtic de classes.....	41
Detalls de la implementació	42
Controladors.....	42
Models i patrons de disseny.....	43
Objectes del domini.....	45
Vistes.....	45
Autenticació d'usuari en el Backend.....	49
Autenticació d'usuari en el Frontend.....	51
Recuperació de la paraula de pas.....	52
Pagament bancari.....	54
Evitant n° d'inscripció repetits i concurrència.....	55
Jocs de proves.....	56
Entorn de desenvolupament.....	57
Conclusions.....	59
Línies de futur.....	59
Glossari de termes.....	61
Bibliografia i referències.....	62
Programació de la part del servidor.....	62
Presentació part client.....	62
Disseny de software i metodologia de desenvolupament.....	62
Annex 1 – Manual de l'administrador de sistemes.....	64
Manual de desplegament de l'aplicació per primer cop.....	64
Execució dels jocs de proves.....	64
Manual de desplegament d'una nova versió.....	65
Annex 2 - Manual de l'administrador de la web.....	66
Part pública.....	66
Part privada.....	66
Gestió d'excursions.....	66
Gestió d'usuaris.....	68
Gestió d'inscripcions.....	68
Gestió d'entitats.....	70
Pagaments.....	71
Inscripció o pagament manual.....	72

Introducció

Justificació i context del projecte

El centre excursionista Olot (CEO) és una entitat dedicada a l'organització de diferents activitats relacionades amb el món de l'esport i el lleure a la muntanya. Des del CEO sempre s'ha apostat per les noves tecnologies, sobretot aplicades en la gestió de les inscripcions i a la promoció de les seves activitats com associació.

El CEO ja disposa d'una web d'informació basada en Wordpress, i realitza les seves inscripcions moltes vegades de forma manual, a través del telèfon o bé utilitzant formularis Google o tecnologies similars. El problema resulta de no poder efectuar pagaments bancaris a través d'aquests mètodes i la dificultat de no poder gestionar els usuaris de forma centralitzada.

La idea d'aquest projecte és doble. D'una banda desenvolupar una web per permetre al CEO, centralitzar totes les inscripcions a les diferents excursions que organitzen en una sola plataforma. L'altre és estudiar quines tecnologies existeixen en el món del desenvolupament web i posar-les en pràctica per desenvolupar aquesta plataforma.

La planificació del projecte

La planificació del projecte es durà a terme prenent com a base la metodologia **PMBOK**, ja que és la metodologia que s'estudia a l'assignatura de gestió de projectes. Tot i així no s'aplicaran tots els mecanismes descrits, sinó que es realitzaran només unes quantes de les activitats descrites, ja que es tracta d'un projecte unipersonal i no fa falta un formalisme tant gran en la comunicació de la informació.

Pel que fa a la metodologia de desenvolupament s'utilitzarà una metodologia iterativa, ja que permet realitzar entregables de forma ràpida i així reduir els riscos de cara al client. Tot i que es farà un disseny inicial acurat degut a les exigències d'entrega de les PAC de l'assignatura.

Donat que l'entrega del projecte es realitzarà en un temps relativament curt, només es podrà realitzar algunes iteracions o fases. Tot i així, com que es tracta d'un projecte real, les possibles millores que expressi el client es tindran en compte en nous cicles de desenvolupament dins el mateix projecte.

S'agafarà com a referència la metodologia de desenvolupament orientada a objectes **OpenUP**, basada en RUP de IBM però molt més simplificada i que és de naturalesa iterativa, aquesta s'adaptarà a les exigències del projecte.

Abast del projecte

L'objectiu general del projecte és la d'automatitzar la inscripció a les diferents excursions que organitza el centre excursionista Olot (CEO), a fi d'augmentar la participació en aquests events i facilitar-ne la gestió.

Que es pretén aconseguir:

- Facilitat en la inscripció del públic en general a les inscripcions.
- Menys gestió per part del personal del centre, que ara inscriuen a mà.
- Augmentar el nombre de participants a les excursions
- Fomentar la participació de la gent jove, més oberta en utilitzar les noves tecnologies.
- Aconseguir una solució adaptada al CEO però que en un futur pugui ser adaptada a altres centres excursionistes.

Com s'aconsegueix.

- Desenvolupant una web per realitzar les inscripcions

- Realitzant la web per tal que sigui accessible via telèfon mòbil, o en dispositius amb pantalles més petites com una "tablet".

Factors crítics d'èxit del projecte

- El temps i les dates d'entrega
- La disponibilitat del client, per realitzar reunions i respondre a diverses qüestions relacionades amb el projecte.

Requisits i característiques del projecte

- **R001** - Realitzar una web per permetre inscripcions a diferents excursions que organitza el centre.
- **R002** - Permetre el pagament a través de targeta bancària.
- **R003** - Permetre que la web, la part pública, sigui consultable a través de telèfons mòbils.

Criteris d'acceptació del projecte

- Es compleixen els dos primers requisits R001 i R002.
- No hi han errors crítics en el sistema
- S'ha provat l'aplicació en una excursió amb èxit.

Tasques del projecte

Donat que s'utilitzarà una metodologia iterativa, el projecte es dividirà en fases. En la primera iteració es planteja tot el projecte, i es realitza l'anàlisi general del sistema. En fases posteriors durant la construcció del sistema s'agafa com a partida l'anàlisi general i se'n fa un refinament si aquest no és del tot correcte.

Anàlisi i planificació

Activitat	Descripció
Pla de treball	Estudi del que s'ha de fer en el projecte i el temps que hi ha disponible. Documents generats - Pla de treball

Identificar i analitzar els requisits

En aquesta etapa es vol identificar els requisits que ha de complir l'aplicació. Un cop identificats s'elaboraran els casos d'ús.

Activitat	Descripció
Anàlisi de requisits	Estudi dels diferents processos del sistema, i obtenció dels requisits que ha de complir el sistema. Documents generats -Elaboració dels casos d'ús. -Elaboració d'un prototip d'interfície de l'aplicació.
Anàlisi de requisits tècnics	S'avaluaran les diferents alternatives tecnològiques disponibles i se'n seleccionarà una, atenent a l'estudi de requisits. Documents generats -Document de descripció de l'arquitectura del sistema

Anàlisi i disseny

En aquesta etapa es realitzarà l'anàlisi complet del sistema.

Activitat	Descripció
Anàlisi de les classes del sistema	S'analitzaran les classes que intervenen en el sistema i s'elaborarà el model de domini. Documents generats -Diagrama estàtic de classes o model de domini
Elaboració model de dades	S'analitzaran les necessitats d'informació del sistema, i es crearà un model de dades relacional. Documents generats -Model relacional
Elaboració del model de processos	S'analitzaran alguns dels processos més complicats i a més objectes implicats i s'elaboraran models que ajudin a descriure'ls. Documents generats -Diagrames d'estat
Entrega PAC2	Agrupació de documents de l'anàlisi i disseny de l'aplicació. Documents generats -PAC2

Construcció del sistema

Inicialment s'ha pensat en construir el sistema en 3 iteracions. Tot i així no es descarta que n'apareguin més, sobretot en les fases finals d'entrega, per acabar d'ajustar l'aplicació a les necessitats del client.

Activitat	Descripció
Iteració 1: Implementació "frontend"	Programació de la part pública de la web, inscripció bàsica d'un usuari a una excursió. - Gestió d'excursions. - Gestió d'usuaris. - Autenticació d'usuaris en el sistema. Inclou - Refinar requeriments. - Desenvolupament de la solució - Testeig de la solució Documents generats - (opcional) Millores en els documents de disseny. - Codi en repositori
Iteració 2: Implementació pagament bancari	Comunicació amb la passarel·la bancària de la Caixa (SERMEPA). - Realització de diferents tests de prova sobre la plataforma. Inclou - Refinar requeriments. - Desenvolupament de la solució - Testeig de la solució

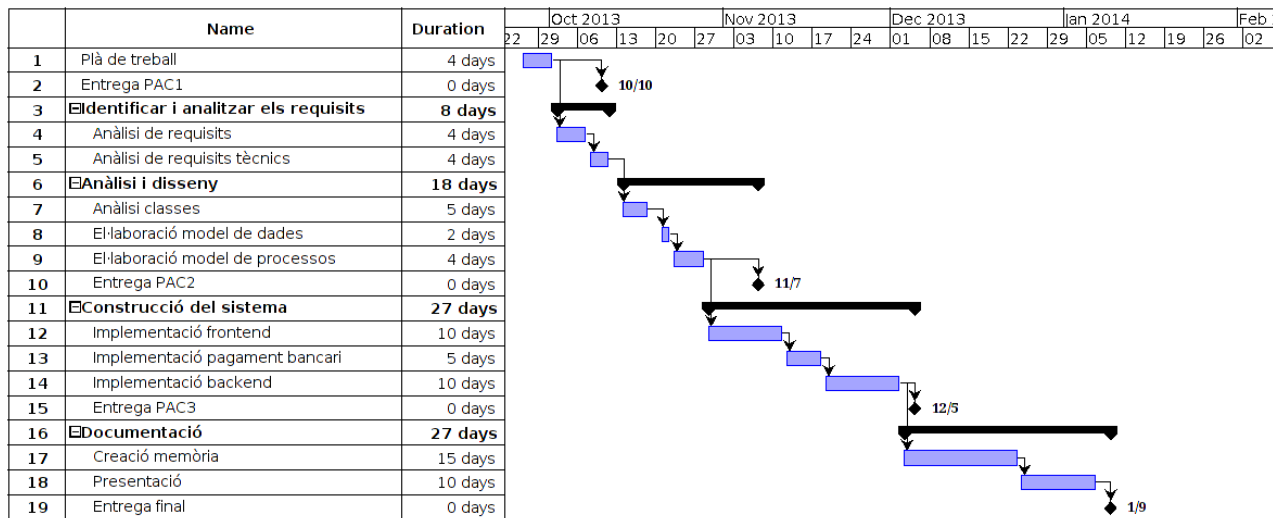
	Documents generats <ul style="list-style-type: none">- (opcional) Millores en els documents de disseny.- Codi en repositori- Codi de testeig per provar la solució de pagament amb la plataforma de proves.
Iteració 3: Implementació "backend"	Elaboració de la web d'administració, que permetrà gestionar les diferents excursions i obtenir llistats. Inclou <ul style="list-style-type: none">- Refinar requeriments.- Desenvolupament de la solució- Testeig de la solució Documents generats <ul style="list-style-type: none">- (opcional) Millores en els documents de disseny.- Codi en el repositori
Entrega PAC3	Entrega de la tercer pràctica del treball. Que inclou el codi de l'aplicació. Documents generats <ul style="list-style-type: none">-PAC3-Codi de l'aplicació

Documentació

Activitat	Descripció
Creació memòria final	Recopilació de tota la documentació del projecte i elaboració de la memòria.
Presentació	Creació del vídeo de presentació del projecte.
Entrega final	Entrega final del projecte Documents generats <ul style="list-style-type: none">- Memòria del projecte.- Vídeo de presentació.

Planificació temporal

La planificació temporal del projecte, és la següent:



Fites importants

- 10/10/2013 – Entrega PAC1.
- 7/11/2013 – Entrega PAC2.
- 11/11/2013 – Implementació “frontend”.
- 18/11/2013 – Implementació pagament bancari.
- 2/12/2013 – Implementació “backend”.
- 5/12/2013 – Entrega projecte.
- 9/01/2014 – Entrega memòria.

Pla de gestió de riscos

Codi	Nom	Causa	Conseqüència	Nivell impacte
R001	Problemes en el coneixement del llenguatge de programació	Falta d'experiència en el llenguatge seleccionat.	Pot fer augmentar el temps d'implementació.	Mitjà
R002	Problemes de temps	La persona encarregada de dur el projecte a terme li falti temps.	Impossibilitat de complir les fites marcades	Alt
R003	Problemes amb el pagament bancari	La plataforma bancària no estigui disponible o hi hagi problemes de posta en marxa	Realització del projecte sense una de les fites marcades. Implementació posterior.	Mitjà
R004	Falta de resposta ràpida del client	Les dades que es demanen al client no estan disponibles en un temps raonable.	S'haurà d'engegar el projecte sense les dades necessàries.	Baix
R005	Problemes amb el servidor	Es tracta d'un servidor de naturalesa compartida, aquest fet pot provocar problemes en la	Hauriem de contractar un servidor dedicat amb el consegüent augment de preu. O bé modificar alguna llibreria,	Baix

		configuració d'alguna llibreria. Fent que l'aplicació no arranqui.	per adaptar-lo al servidor.	
--	--	--	-----------------------------	--

Recursos utilitzats i pressupost orientatiu

Hores de d'analista-programador previstes per l'execució del projecte. El preu de l'hora el considerem a 30€ l'hora, donat que és la mateixa persona l'encarregada de realitzar el projecte.

Activitat	Hores/persona	Preu	Tipus
Planificació del projecte	10	300€	Hores analista.
Identificar i analitzar requisits	32	960€	Hores analista.
Anàlisi i disseny	72	2160€	Hores analista.
Construcció	108	3240€	Hores analista/programador
Documentació	108	3240€	Hores analista
TOTAL	330	9900€	

Altres costos atribuïbles al projecte:

Activitat	Preu/mensual
Cost <i>hosting</i> anual ARSYS.	96€
Cost del domini anual.	16.99€
Trucades i visites a casa el client	250€

Situació del mercat sobre les webs d'inscripcions

Des de fa un temps, han aparegut al mercat un bon munt de pàgines web d'empreses que es dediquen a la gestió d'inscripcions a esdeveniments esportius. Aquest tipus d'empreses facilita l'obtenció de les dades de l'usuari i de la inscripció, el pagament, i la generació del llistat d'inscrits.

Estudiant algunes d'aquestes webs s'ha determinat que n'hi ha de tres tipus diferents.

- Les que permeten que els organitzadors de l'event decideixin moltes de les opcions de la inscripció a través d'opcions configurables.
- Les que simplement mostren un bon munt de tarifes i permeten a l'usuari triar-ne una.
- Les que tenen opcions prefixades, normalment un sol preu.

Les més habituals són les del segon i tercer tipus, segurament per la seva major simplicitat pel que fa al manteniment, tot i que gradualment van apareixent webs més complexes i més especialitzades en un tipus d'esport.

Algunes de les webs estudiades sobre el tema es poden trobar en els següents enllaços:

http://www.albecam.com	
---	---

http://www.youevent.es	
http://www.runningcampus.com	

Val a dir que el CEO podria utilitzar una d'aquestes plataformes per la gestió de curses, però donat el nivell de detall que demanen en les inscripcions, seria complicat d'aconseguir. A banda d'això normalment aquest tipus de web cobren una comissió per inscripció.

Cal destacar també que algunes de les webs estudiades, no permeten la gestió d'usuaris, i cal introduir cada vegada que es fa una inscripció totes les dades personals.

Identificar i analitzar els requisits

En aquesta etapa es vol identificar els requisits que ha de complir l'aplicació. Un cop identificats s'elaboraran els casos d'ús tal i el prototip d'interfície de l'aplicació.

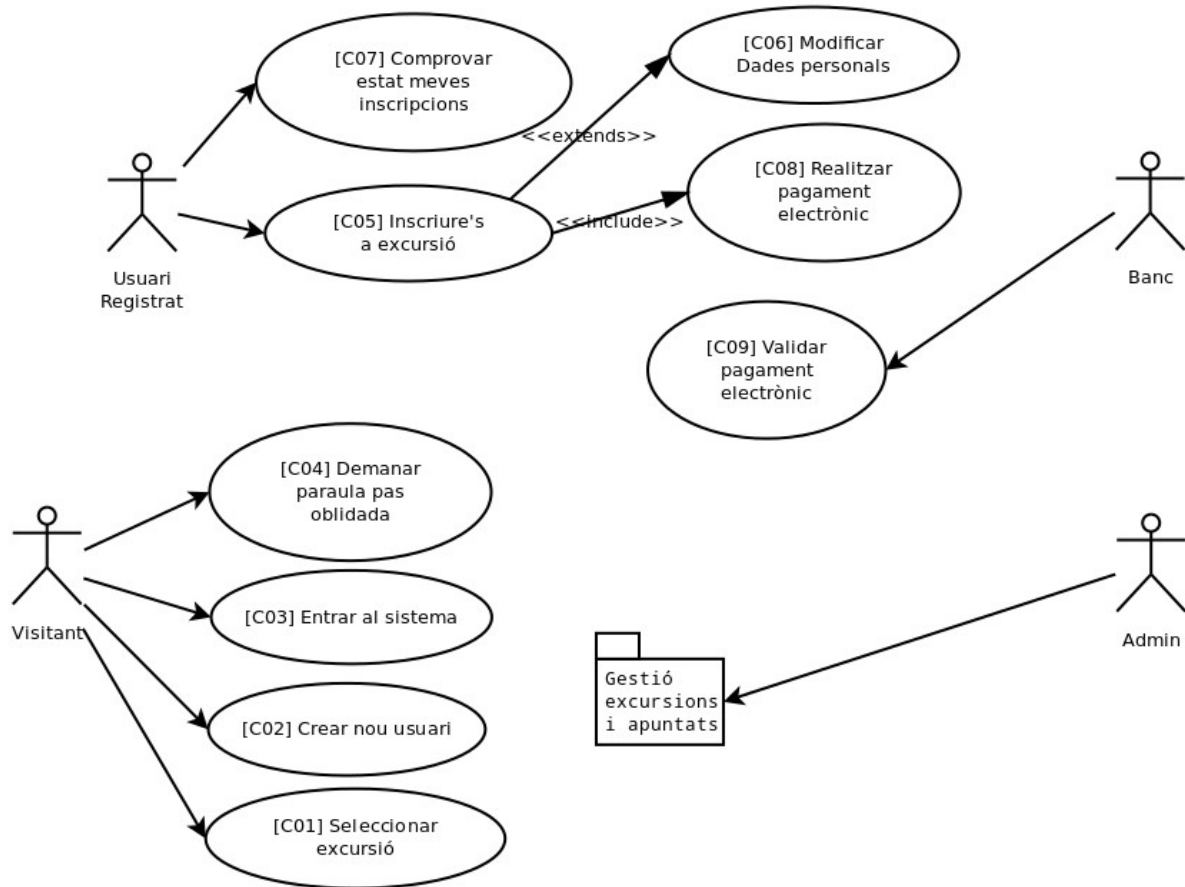
Per identificar els requisits s'ha seguit la metodologia de l'entrevista presencial amb el client i s'han realitzat nombrosos esbossos i dibuixos per arribar a copsar la idea, a partir d'aquí s'ha definit un prototip inicial que pot ser modificat en fases posteriors del projecte, però que conté l'essència del que es vol dur a terme.

Les figures que apareixen a continuació mostren alguns d'aquests esbossos inicials, que representen el FrontEnd de l'aplicació.

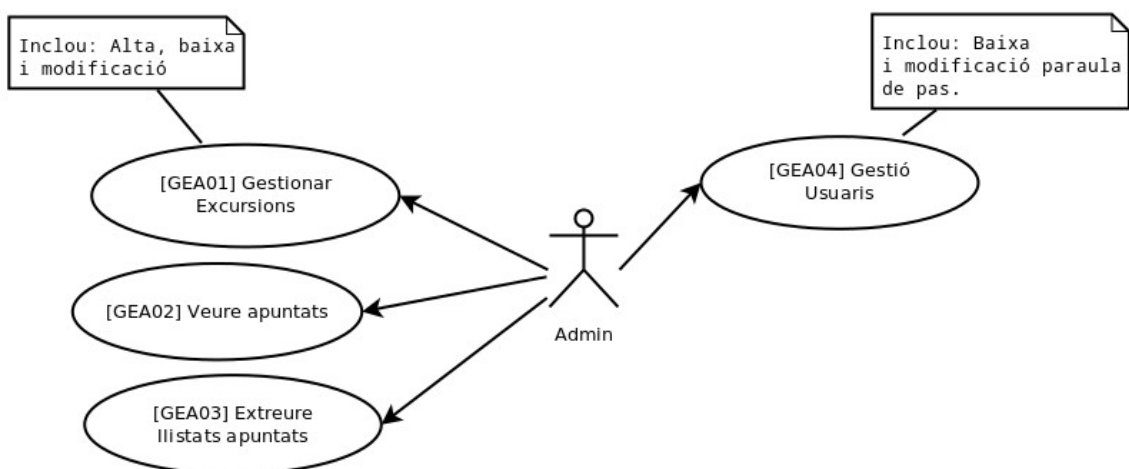


Casos d'ús

Diagrama principal



Gestió excursions i apuntats



Actors del sistema

El sistema constarà de 4 actors tal i com es mostra en el diagrama anterior.

Visitant: Representa una persona no identificada, que està interessada en apuntar-se a una excursió. Aquesta persona pot crear-se un nou usuari per passar a ser un usuari registrat.

Usuari registrat: És un usuari conegut pel centre, les seves principals accions son inscriure's a una excursió i obtenir les butlletes de les excursions on s'ha apuntat.

Usuari administrador: La feina de l'administrador correspon a crear excursions, i controlar les persones que s'apunten a les diferents excursions. També té la possibilitat de modificar les claus d'accés als usuaris, per tal de facilitar-los la entrada.

Banc: El banc és un sistema extern, i la seva funció és notificar a la web del CEO que una transacció ha finalitzat amb èxit o amb fracàs.

Cas d'ús C01: Seleccionar excursió

Actors implicats: Visitant

Flux bàsic:

1. Es mostra la pantalla de benvinguda amb les excursions on l'usuari es pot apuntar.
2. L'usuari selecciona una excursió entre les disponibles per efectuar-ne la inscripció.

Flux alternatiu:

Si no hi han excursions disponibles, el sistema notificarà sobre aquest fet.

Precondició: -

Postcondició: L'excursió seleccionada serà la que s'utilitzarà durant la inscripció.

Cas d'ús C02: Crear nou usuari

Actors implicats: Visitant

Flux bàsic:

1. El sistema mostrarà la pantalla de registre de l'usuari amb 3 opcions. Entrar al sistema, recuperar paraula de pas oblidada i donar d'alta un nou usuari.
2. Al fer un click sobre el botó de donar d'alta apareixerà un formulari on es demanaran les dades de l'usuari.
3. Es validaran les dades i es comprovarà que no hi hagi cap soci amb el mateix correu electrònic.
4. L'usuari fa click sobre el botó de crear i es crea l'usuari en el sistema.

Flux alternatiu:

1. L'usuari ja existeix al sistema. (veure punt 3 flux bàsic)
2. Es notifica d'aquest fet a l'usuari i es torna a repetir el punt.

Flux alternatiu 2:

1. Per algun motiu no es poden guardar les dades a la base de dades. Es notifica a l'usuari a través del mateix formulari d'alta.

Precondició: -

Postcondició: -

Cas d'ús C03: Entrar al sistema

Actors implicats: Visitant

Flux bàsic

1. L'usuari entra el nom i la paraula de pas a la pantalla de registre.
2. El sistema comprova que existeixi a la BD i que la paraula de pas entrada coincideixi amb la registrada.
3. El sistema dona l'usuari com a validat i redirigeix a la pàgina de pagament de l'excursió demanada.

Flux alternatiu: L'usuari no es troba al sistema

Després del pas 2, si l'usuari o la paraula de pas no coincideixen es torna al pas 1.

Precondicions: -

Postcondició: L'usuari ha iniciat sessió

Cas d'ús C04: Demanar paraula de pas oblidada.

Actors implicats: Visitant

Flux bàsic

1. El cas d'ús s'inicia a la pantalla de registre de l'usuari.
2. L'usuari entra el seu correu electrònic.
3. El sistema genera una URL amb la direcció de la pàgina web del canvi de paraula de pas juntament amb una sèrie de dades xifrades.
4. El sistema envia un correu al compte de l'usuari amb la URL anterior.
5. L'usuari accedeix a la URL marcada i entra una nova paraula de pas juntament amb la seva adreça de correu.

Flux alternatiu: El correu de l'usuari i les dades de l'usuari no coincideixen, el sistema generarà un error i el mostrarà a l'usuari sense donar masses detalls, ja que podria tractar-se d'un atacant.

Precondicions: L'usuari ha de tenir un compte d'usuari creat.

Postcondició: L'usuari pot tornar a entrar al sistema si s'ha seguit el flux normal.

Cas d'ús C05: Inscriure's a una excursió

Actors implicats: Usuari registrat

Flux bàsic

1. El cas d'ús s'inicia el moment d'iniciar sessió al sistema i havent seleccionat una excursió prèviament.
2. Es mostren les dades de l'usuari per si aquest vol realitzar algun canvi en les seves dades personals. A la mateixa pantalla es demanen les dades específiques del tipus d'excursió seleccionada.
3. Si les dades presentades son correctes l'usuari selecciona les opcions desitjades pel tipus d'excursió i entra les informacions demanades. Fent click al botó de continuar.
4. El sistema guarda les dades específiques de l'excursió.
5. El sistema dirigeix l'usuari a la pantalla de pagament en cas que l'excursió sigui de pagament.

Flux alternatiu:

- L'usuari vol modificar les dades personals, entrades en el procés de registre. Accedeix al cas d'ús de Modificar dades personals.
- L'usuari ja es troba inscrit a l'excursió. El sistema donarà error, notificant a l'usuari d'aquest fet.
- El sistema dona error al guardar les preferències de l'usuari, es notificarà de l'error en el mateix formulari.

- La sessió d'usuari ha caducat. El sistema detectarà l'error i redirigirà l'usuari a la pantalla de registre.

Precondicions: L'usuari ha de tenir un compte d'usuari creat i una excursió seleccionada prèviament.

Postcondició: -

Cas d'ús C06: Modificar dades personals

Actors implicats: Usuari registrat

Flux bàsic

1. El cas d'ús s'inicia el moment de demanar les dades específiques de l'excursió a l'usuari. Apareixerà un enllaç que permetrà modificar les dades de l'usuari.
2. L'usuari farà click sobre el botó de modificar dades i li apareixerà el formulari de modificació de dades de l'usuari. Ho podrà modificar tot a excepte el correu electrònic.
3. L'usuari acceptarà les modificacions i es tornarà al punt 1, per continuar introduint les dades de l'excursió.

Flux alternatiu:

- El sistema dona error al guardar les dades l'usuari, es notificarà de l'error en el mateix formulari.
- La sessió d'usuari ha caducat. El sistema detectarà l'error i redirigirà l'usuari a la pantalla de registre.

Precondicions: L'usuari ha de tenir un compte d'usuari creat i una excursió seleccionada prèviament.

Postcondició: -

Cas d'ús C07: Comprovar estat meves inscripcions.

Actors implicats: Usuari registrat

Flux bàsic:

1. El cas d'ús s'iniciarà a l'inici de l'aplicació. Es mostraran les excursions disponibles i un enllaç per veure les excursions on s'ha apuntat l'usuari anteriorment.
2. L'usuari accedeix al formulari d'autenticació i s'autentifica.
3. L'usuari accedeix a una pantalla on es mostren totes les inscripcions efectuades.
4. Accedeix a una inscripció i en pot visualitzar la seva butlleta d'inscripció.

Flux alternatiu:

- El sistema detecta que no s'ha realitzat cap inscripció, s'informarà d'aquest fet.
- L'usuari no es pot autenticar, donarà error i no deixarà accedir.

Precondicions: L'usuari ha d'estar prèviament registrat per poder accedir.

Postcondicions: -

Cas d'ús C08: Realitzar pagament electrònic

Actors implicats: Usuari registrat, banc

Flux bàsic:

1. El cas d'ús s'inicia després d'entrar les dades específiques d'una excursió, si aquesta és de pagament.
2. Apareix una pantalla on s'informa a l'usuari del preu total a pagar i les condicions de pagament.
3. Al fer "click" al botó pagar, s'accedirà a la plataforma de pagament bancari.
4. L'usuari treballarà amb la plataforma bancària.
5. Si el resultat de la transacció ha estat correcte, el banc redirigirà a l'usuari a una pàgina on s'informarà que el pagament s'ha efectuat correctament.
6. Posteriorment el sistema mostrarà la butlleta d'inscripció.

Flux alternatiu:

- El sistema detecta que no s'ha realitzat cap inscripció, s'informarà d'aquest fet.
- L'usuari no pot realitzar el pagament per algun motiu. El banc redirigirà l'usuari a una pàgina de pagament erroni que informarà d'aquesta situació.
- Es redirigirà a l'usuari a la pantalla principal per intentar de nou el pagament.

Precondicions: L'usuari ha d'estar prèviament registrat i autenticat per poder accedir.

Les dades que requereix l'excursió han d'estar entrades correctament.

Postcondicions: -

Cas d'ús C09: Validar pagament electrònic.

Actors implicats: banc

Flux bàsic:

1. La passarel·la de pagament del banc informarà a la web que s'ha realitzat un pagament correctament.
2. La web s'annotarà entre altres, el codi de pagament, el número de comanda i la data en que s'ha realitzat.
3. Posteriorment es passarà a marcar una inscripció com a vàlida i s'assignarà un número d'inscrit (codi pagament) a l'excursió.

Flux alternatiu:

- Si el pagament no correspon a cap inscripció. S'annotarà igualment els codis de pagament per que el client pugui veure la incidència. Però s'obviarà el pas de marcar una inscripció com a vàlida.

Precondicions: L'usuari ha de realitzar un pagament de forma correcta per tal que la passarel·la es posi en contacte amb la web per informar del pagament.

Postcondicions: La inscripció ha finalitzat correctament i l'usuari es troba inscrit amb un número únic a l'excursió.

Cas d'ús GEA01: Gestionar excursions.

Actors implicats: Administrador

Es tracta d'un manteniment bàsic i s'ha optat per agrupar tots els casos d'ús en un de sol per facilitar-ne la llegibilitat.

Flux bàsic:

1. L'usuari seleccionarà aquesta opció dins el menú d'administració on es mostrarà una entrada per gestionar cada tipus d'excursió.
2. Es mostraran les excursions del tipus seleccionat per l'usuari.
3. L'usuari podrà crear una nova excursió, modificar-la o bé eliminar-ne una de les existents (sempre i quan no hi hagin usuaris inscrits). També existirà la possibilitat d'activar-la o desactivar-la per fer-la accessible a públic. L'activació es realitzarà a través de la data de l'event, si es una data passada es considera desactivada.

Flux Crear:

1. Es demanaran les dades segons el tipus de l'excursió.
2. Es guardaran les dades, segons el tipus.

Flux Modificar:

1. El sistema deixarà modificar totes les dades excepte el codi de l'excursió. Algunes dades podran ser de modificació restringida si hi han usuaris apuntats a l'excursió.
2. Es modificaran les dades, segons el tipus d'excursió.

Flux Eliminar:

1. El sistema permetrà eliminar una excursió, sempre i quan no tingui usuaris inscrits.

Flux alternatiu:

- L'usuari només visualitza dades.
- L'usuari crea una excursió nova.
- L'usuari modifica una excursió.
- L'usuari elimina una excursió existent, sempre que no hi hagin usuaris inscrits.

Precondicions: L'usuari ha de ser administrador i està autenticat per accedir a aquest cas d'ús.

Postcondicions: S'actualitzarà la informació de les excursions.

Cas d'ús GEA02: Veure apuntats

Actors implicats: Administrador

Flux bàsic:

1. L'usuari seleccionarà aquesta opció dins el menú d'administració on es mostrarà una entrada per veure els apuntats de cada tipus d'excursió.
2. L'usuari seleccionarà la excursió de la qual vol veure els apuntats.
3. El sistema mostrarà una llista dels usuaris apuntats i les seves dades referents al pagament i al número d'inscripció.

Flux alternatiu:

- En cas que l'excursió no tingui cap usuari inscrit el sistema notificarà sobre aquesta incidència.

Precondicions: L'usuari ha de ser administrador i està autenticat per accedir a aquest cas d'ús.

Postcondicions: -

Cas d'ús GEA03: Extreure llistat d'apuntats

Actors implicats: Administrador

Flux bàsic:

1. Partint del cas d'ús de veure apuntats. Un cop l'usuari hagi elegit una excursió.
2. El sistema mostrarà els enllaços a les diferents possibilitats d'exportació de llistats. Que dependran directament del tipus d'excursió seleccionat.
3. L'usuari seleccionarà quin tipus d'exportació vol i es generarà un fitxer CSV (Comma Separated Value)

Flux alternatiu:

- En cas que l'excursió no tingui cap usuari inscrit el sistema notificarà sobre aquesta incidència i no en permetrà exportar cap mena de llistat.

Precondicions: L'usuari ha de ser administrador i està autenticat per accedir a aquest cas d'ús.

Postcondicions: -

Cas d'ús GEA04: Gestió d'usuaris

Actors implicats: Administrador

Flux bàsic:

1. L'usuari seleccionarà aquesta opció dins el menú d'administració on es mostrarà una entrada per gestionar cada tipus d'excursió.
2. Es mostraran els diferents usuaris del sistema.
3. L'administrador podrà modificar la paraula de pas d'un usuari o bé donar-lo de baixa, de forma que no es pugui autenticar més dins el sistema. Un usuari només podrà ser donat de baixa si no té inscripcions associades.

Flux Crear:

3. Es demanaran les dades de l'usuari i es realitzen les comprovacions oportunes.
4. Es guardaran les dades a la BD.

Flux Modificar:

1. El sistema deixarà modificar totes les dades excepte el mail de l'usuari.
2. Es modificaran les dades a la BD.

Flux Eliminar:

1. El sistema permetrà eliminar un usuari, sempre i quan no tingui inscripcions realitzades.

Flux alternatiu: -

Precondicions: L'usuari ha de ser administrador i està autenticat per accedir a aquest cas d'ús.

Postcondicions: S'actualitzarà la informació dels usuaris.

Prototip d'interfície de l'aplicació

Prototip de l'administració o Backend

El funcionament del Backend es basa en un menú principal on hi constaran com a mínim 3 opcions. Gestió d'excursions, gestió d'apuntats o inscripcions i gestió d'usuaris. L'usuari seleccionarà la opció que desitgi del menú principal.

En cas que la opció seleccionada sigui una excursió, apareixeran separades per tipus les diferents excursions disponibles. Segons el tipus d'excursió les dades a tractar i tarifes a aplicar seran diferents. A continuació es mostra el manteniment d'una excursió bàsica.

Administració

Menú	Contingut
Gestió excursions Tipus Embardissada Tipus 24 hores Altres Gestió apuntats Tipus Embardissada Tipus 24 hores Altres Gestió usuaris	

Administració

Menú	Gestió excursions
Gestió excursions Tipus Embardissada Tipus 24 hores Altres Gestió apuntats Tipus Embardissada Tipus 24 hores Altres Gestió usuaris	Embardissada 2014 Data: 14 Abril de 2014 Volta volcans Data: 11 Març de 2014 Nova excursió

Contingut - Nova excursió

Id	<input type="text"/>
Nom	<input type="text"/>
Data event	<input type="text"/>
Activada	<input type="checkbox"/>
Tipus de tarifa 1	<input type="text"/>
Tipus de tarifa 2	<input type="text"/>
....	
Modificar	Cancelar

Contingut - Editar excursió

Id	<input type="text" value="1"/>
Nom	<input type="text" value="Volta volcans"/>
Data event	<input type="text"/>
Activada	<input type="checkbox"/>
Tipus de tarifa 1	<input type="text"/>
Tipus de tarifa 2	<input type="text"/>
....	
Modificar	Cancelar

En cas que la opció seleccionada sigui la gestió d'apuntats o inscrits, es seleccionarà un tipus d'excursió per veure les persones que s'hi han inscrit.

A continuació pareixerà un llistat amb les diferents excursions disponibles. Fent click al damunt d'una excursió ens apareixerà el llistat d'inscrits amb les dades bàsiques d'una inscripció.

L'aplicació permetrà descarregar un document CSV amb les dades de les inscripcions.

Administració

Menú

- Gestió excursions
 - Tipus Embardissada
 - Tipus 24 hores
 - Altres
- Gestió apuntats
 - Tipus Embardissada**
 - Tipus 24 hores
 - Altres
- Gestió usuaris

Contingut - gestió apuntats

- Embardissada 2014**
Data: 14 Abril de 2014
- Volta volcans**
Data: 11 Març de 2014

Apuntats embardissada 2014

Mail	Nom	Cognoms	DataInscripcio	NComanda	NInscripció
xs@gm.com	Xevi	Serra	1/1/2014	123214232	1
jpi@ht.com	Josep	Pi	2/3/2014	123232111	2
jm@ht.com	Josep	Masó	3/3/2014	123323412	-

[Extreure llistats apuntats](#)

Apuntats embardissada 2014 - Extreure llistats

- [Tots els apuntats, totes les dades](#)
- [Tots els apuntats, dades bàsiques](#)




En cas de seleccionar la gestió d'usuaris, el sistema mostrarà una llista amb tots els usuaris. Les opcions bàsiques que es podran realitzar sobre un usuari seran eliminar-lo o bé modificar-ne la paraula de pas.

Administració

Menú

- Gestió excursions
 - Tipus Embardissada
 - Tipus 24 hores
 - Altres
- Gestió apuntats
 - Tipus Embardissada
 - Tipus 24 hores
 - Altres
- Gestió usuaris**

Gestió usuaris

Mail	Nom	Cognoms
 xs@gm.com	Xevi	Serra
 jpi@ht.com	Josep	Pi
 jm@ht.com	Josep	Masó

Gestió usuaris - Modificar paraula pas [xs@gm.com](#)

Nova paraula pas:

Repetir paraula pas:

[Modificar](#) [Cancelar](#)

Prototip del Frontend

El prototip del frontend s'ha realitzat directament amb HTML a fi d'utilitzar-ne algunes parts per el disseny definitiu de l'aplicació.

La idea inicial del Frontend és que funcioni com un assistent, de forma que si es segueixen tots els passos s'aconsegueixi una inscripció.

Inicialment l'usuari accedirà a la pàgina principal, on podrà seleccionar les excursions actives per apuntar-s'hi. També té la possibilitat de veure les inscripcions que té realitzades fins al moment.

Web inscripcions del CEO

Consultar les meves inscripcions

Click [aquí](#)

Apuntar-se a excursions

24 hores a muntanya

Es celebra el dia: 31/10/2013

Embardissada 2014

Es celebra el dia: 10/04/2014

Esmorzar a finestres

Es celebra el dia: 30/11/2013

Un cop hagi escollit una inscripció, li apareixerà la pantalla de registre. En aquesta pantalla hi haurà tres opcions.

Si l'usuari ja estava registrat a la web, simplement haurà d'introduir el seu compte de correu (identificador d'usuari) i paraula de pas.

També hi ha la possibilitat de crear un nou compte, si és el primer cop que l'usuari accedeix a la web d'inscripcions del CEO.

Finalment també hi ha la opció de recuperar una paraula de pas perduda i així restablir un compte ja existent.

Identifica't

Usuari ja registrat

Ja consto al registre i em vull inscriure a l'excursió.

Correu electrònic:	<input type="text"/>
Paraula de pas:	<input type="text"/>
<input type="button" value="Entrar"/>	

No recordo la paraula de pas

T'enviarem la teva contrasenya al compte de correu, si ja tens un compte.

Correu electrònic:	<input type="text"/>
<input type="button" value="Demandar el password"/>	

Nou usuari

Per realitzar una nova inscripció fes click [aquí](#)

En aquest punt es pot veure la pantalla d'inscripció a una excursió en particular.

La pantalla està formada per dues parts: Una primera part on l'usuari pot revisar i si ho desitja modificar les seves dades personals, i una segona part on l'usuari introdueix les dades referents a l'excursió.

Dades Embardissada 2014

* **Correu electrònic:**
xeviterr@gmail.com

Nom i cognoms:
Xevi Terradellas

Sexe:
H

Data de naixement (dd/mm/aaaa):
22/01/2014

Adreça:
C/ Proves

Població i codi postal:
Olot (17800)

Telèfons:
999999999/999888888

Entitat:
CEO

Número de soci:
Si les dades son incorrectes feu click [aquí](#) per modificar-les.

* **Sou soci?**
 Sí No

* **Adult o nen?**
 Adult Nen

* **Esteu federat?**
 Sí No

Número de federat

* **Voleu assegurança? (10 €)**
 Sí No

Dni (només si voleu assegurança):

Marqueu si voleu rebre informació sobre...

l'Embardissada

Esquí

Muntanya

Xerrades

A continuació es mostra la pantalla de pagament. On el sistema mostrarà el preu de l'excursió i donarà a l'usuari el número de comanda que se li ha assignat per realitzar el pagament.

Un cop l'usuari confirmi el pagament, el sistema es posarà en contacte amb la passarel·la bancària per fer-lo efectiu.

Pagament

La quantitat a pagar és: 10

Soci nen 10€

Ja heu entrat les dades d'inscripció a l'excursió.
Finalitzat el pagament rebreu un correu electrònic amb la butlleta d'inscripció .

MOLT IMPORTANT: Guardeu-vos aquest número de comanda, us servirà per obtenir el full d'inscripció un cop finalitzada la transferència bancària, si falla el correu electrònic.

1383734371

Podeu fer el pagament per Tarjeta segura o transferència.

Si opteu per el pagament amb targeta i per la vostra seguretat aquesta ha d'esser SEGURA (demana conformitat al mòbil o per altre sistema segur).

Finalment, si la inscripció s'ha efectuat correctament, apareixerà la butlleta d'inscripció, on hi constaran les dades d'usuari i les dades específiques de l'excursió. En la butlleta també hi apareixerà un codi de barres que identificarà el dorsal de cada corredor. Aquest codi de barres es calcularà per cada butlleta.

Instruccions Embardissada 2014

Us adjuntem butlleta d'inscripció que caldrà lliurar a la sortida i a cada un dels controls.
A la SORTIDA previa identificació amb qualsevol carnet us faran entrega del mapa i del llibret explicatiu del recorregut.

Per lliurar a la sortida

Número d'inscripció: **1383724119**



Nom:	Xevi		
Cognoms:	Terr		
Adreça:	33C/ de proves		
Població:	Olot		
Data inscripció:	06/11/2013		
Entitat:	G. E. Amics de l'Esport	Número comanda:	1383724119
Telèfon:	999999999	Telefon 2:	333333333
email:	xeviterr@gmail.com	Núm. inscripció:	1383724119

Vull rebre	L'Embardissada: si	Activitats d'esqui: si
Informació de:	Activitats de muntanya: no	Activitats sala xerrades: no

Assegurança

Nom:	Xevi	D.N.I:	11111111y
Cognoms:	Terr		
Data naixament:	07/11/1918		

Arquitectura inicial del sistema

Per desenvolupar el sistema s'ha pensat en utilitzar Php degut a que l'aplicació ha d'estar suportada en un entorn de *hosting* compartit. També s'han avaluat altres alternatives com ara el Python i el Java.

Pel que fa al Java s'ha descartat perquè no es podia aprofitar l'entorn de *hosting* del client, que ja hi té algunes aplicacions instal·lades, totes basades en php, i això hagués suposat una tasca addicional de migració, així com l'encariment de *hosting*, degut a la necessitat de disposar d'un servidor dedicat. Tot i així seria una opció interessant a tenir en compte en un futur, ja que proporciona unes eines excel·lents per el desenvolupament web.

Pel que fa al Python, tot i que es tracta d'un bon llenguatge, s'ha descartat per el temps excessiu que s'hauria d'haver invertit pel seu aprenentatge, fet que probablement hagués produït el no poder complir amb les fites marcades. Concretament s'ha estudiat la possibilitat de fer-ho en Python i Django.

Finalment s'ha optat per PHP donat que es tracta d'un llenguatge senzill i de ràpida implementació, tot i que té el problema de que es tracta d'un llenguatge sense tipus i que a vegades la seva orientació a objectes no és del tot funcional, com ara en l'apartat que fa referència al polimorfisme.

Per agilitzar la codificació s'ha pensat en utilitzar un *framework* que implementi el patró MVC (Model Vista Controlador), concretament s'ha estudiat la possibilitat d'utilitzar varis *frameworks* i finalment s'ha optat per Codeigniter, ja que és un *framework* senzill, ben documentat, i sobretot que s'adapta molt bé a entorns de *hosting* compartit.

Els *frameworks* estudiats han estat:

Zend Framework 2:

Molt interessant i bastant conegut al món empresarial, però amb una corva d'aprenentatge bastant gran. A més la documentació de l'API no és massa complerta i això fa que molt sovint calgui accedir al codi font. Un problema molt greu que s'ha detectat és que el servidor LAMP s'ha de configurar i això fa que no sigui una bona solució per un entorn de *hosting* compartit com en el que es vol desplegar l'aplicació.

Laravel 4:

És una opció molt bona i supera al Codeigniter en varis aspectes. Només s'han trobat tres inconvenients importants. El primer és que es necessita la línia de comandes per treballar, i això el fa més complicat de posar en un entorn de *hosting*. El segon inconvenient és la maduresa del framework respecte a Codeigniter, fa que no hi hagi tanta documentació. L'últim inconvenient és que demana la última versió de PHP i no sempre la tindrem disponible en el servidor de *hosting*. Tot i així és una opció de futur a tenir en molt en compte.

Anàlisi i disseny

En aquesta etapa es realitzarà l'anàlisi inicial del sistema. Aquest anàlisi pot ser modificat en fases posteriors, ja que es rebrà "feedback" del client i això pot provocar alguns canvis en l'anàlisi d'alguna de les parts de l'aplicació.

Aquest tipus de modificacions es poden dur a terme degut a la utilització d'una metodologia de tipus incremental per a la construcció d'aquest projecte (veure apartat planificació).

Diagrama estàtic de classes

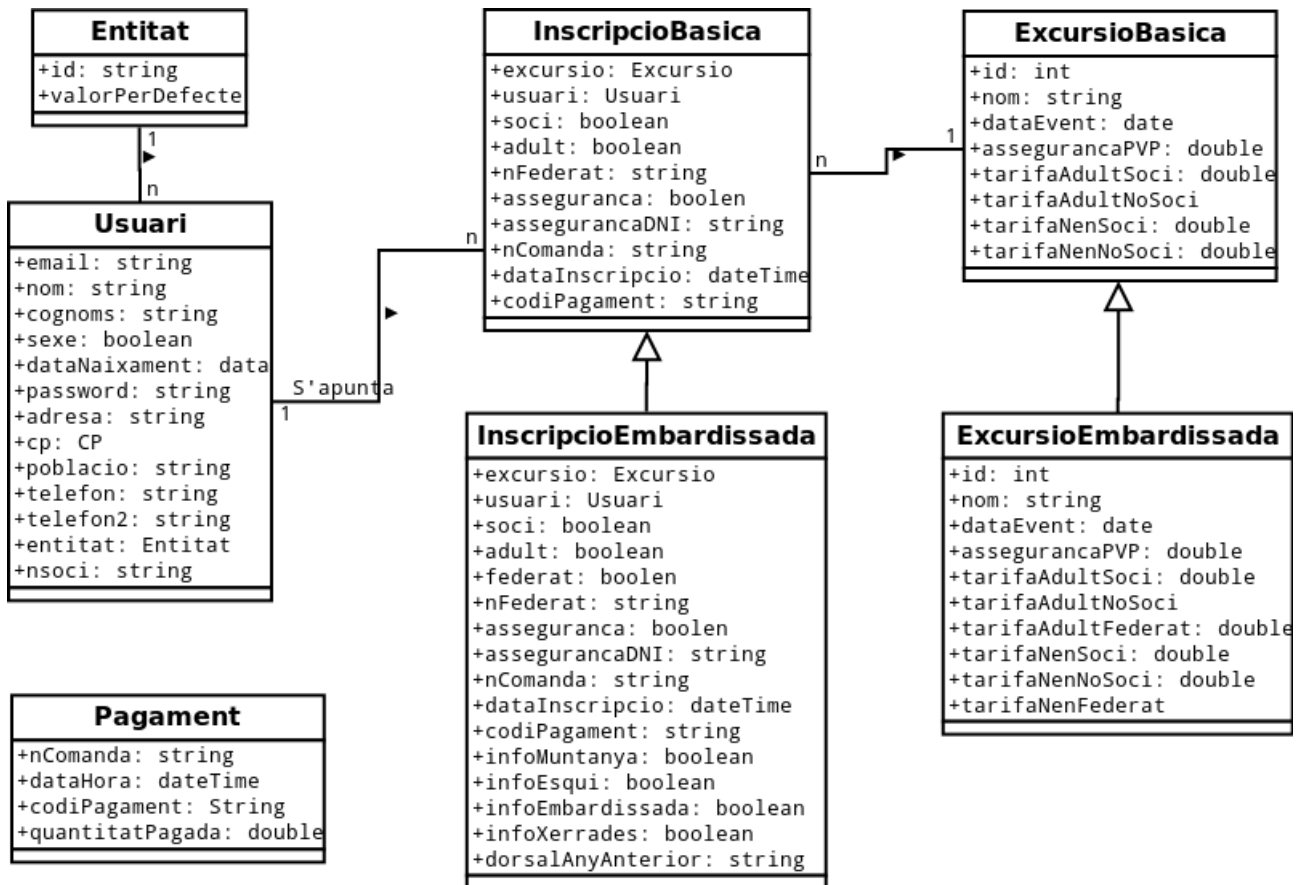


Diagrama d'estructura MVC

En aquest punt es mostra com està estructurada la web, seguint el patró MVC. Per fer-ho només es mostra el disseny de les classes que fan referència a usuaris.

En aquest diagrama es pot veure el controlador d'usuaris, amb totes les possibilitats que ofereix. Cada mètode equival a una crida HTTP. Un cop s'ha realitzat la crida el controlador carrega el model d'usuaris amb les dades corresponents, segons la crida que s'hagi realitzat.

Finalment el model realitza una instància de la vista i li transfereix el contingut del model per ser visualitzat. Cal destacar que en l'esquema apareixen dues vistes de suport, *header* i *footer*, que són dades comunes de la capçalera i el peu de pàgina, que es combinen amb la vista per formar la vista completa.

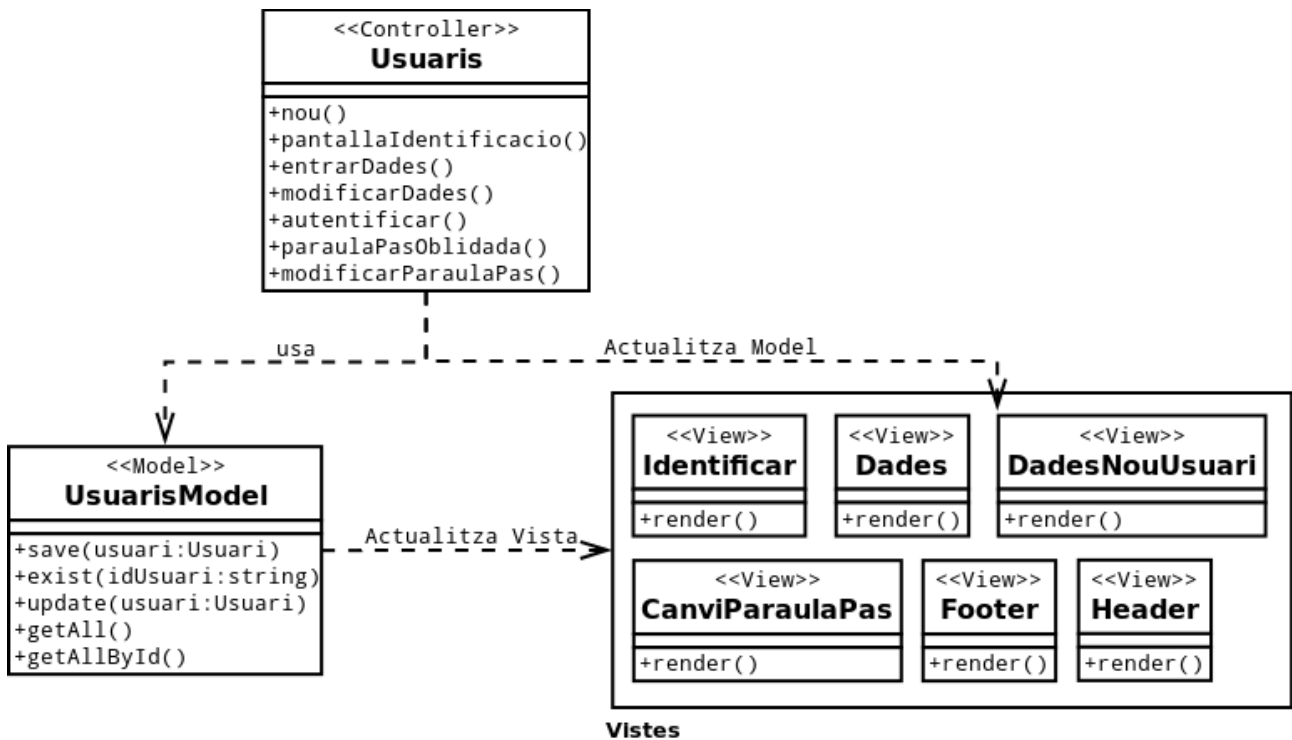
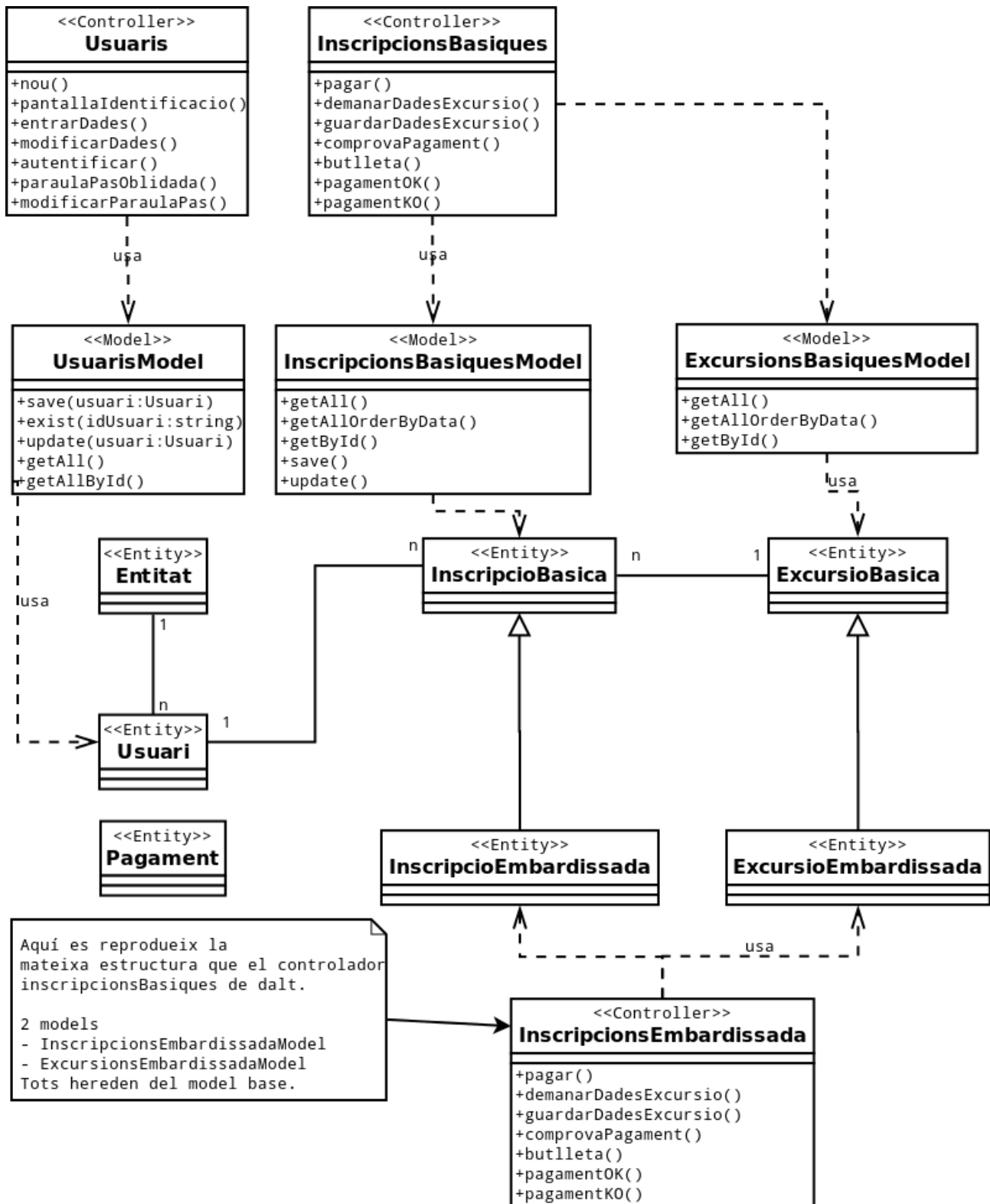


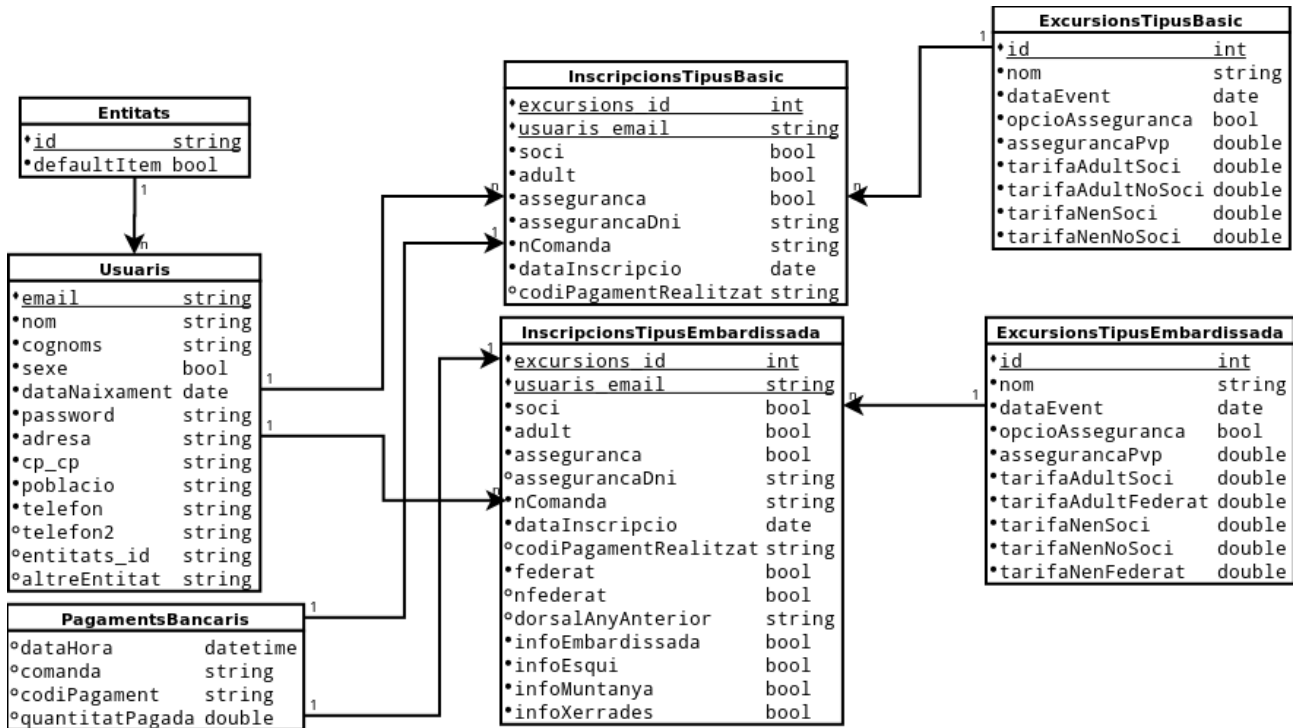
Diagrama de disseny de l'aplicació en MVC



L'aplicació segueix el patró de disseny MVC (Model Vista Controlador). S'ha intentat que els models siguin el més grans possibles en detriment dels controladors, per tal de disposar de controladors amb la mínima

lògica de negoci i que siguin fàcilment llegibles i models altament reutilitzables.

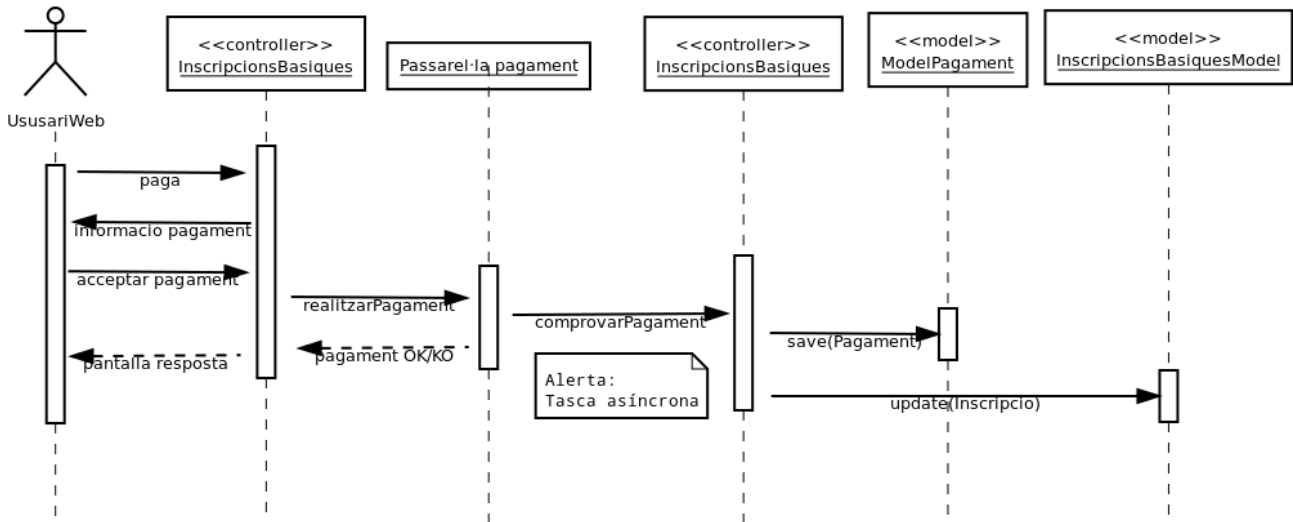
Diagrama model relacional



Inicialment s'ha plantejat l'estratègia de disposar d'una taula per cada classe de l'herència. Tot i que en un principi es volen tractar les dades per separat. Aquest fet podria canviar si es detecta que l'usuari demana un tractament unificat de les excursions.

Diagrama de seqüència del pagament bancari

L'aplicació no té punts massa complicats, el més rebuscat és sense dubte el pagament bancari, i per aclarir el seu funcionament s'ha realitzat el seu diagrama de seqüència.



El punt a destacar es el comprovaPagament entre la passarel·la bancària i el controlador d'inscripcions bàsiques. Es tracta d'una tasca asíncrona que el banc pot iniciar en qualsevol moment i que s'escapa al control de l'usuari.

Aquesta tasca notificarà a la web del CEO si un pagament s'ha realitzat o no. Si no s'ha realitzat el banc tornarà un codi d'error, indicant la causa que ha provocat que no s'hagi completat la transacció.

Anàlisi de l'arquitectura del sistema

A fi de facilitar el poder comprendre l'arquitectura del sistema s'han estudiat diferents punts de vista:

Punt de vista físic: Mostra com està estructurat el sistema, i com funcionarà un cop estigui posat en producció.

Punt de vista lògic: Mostra les diferents capes que formen el sistema.

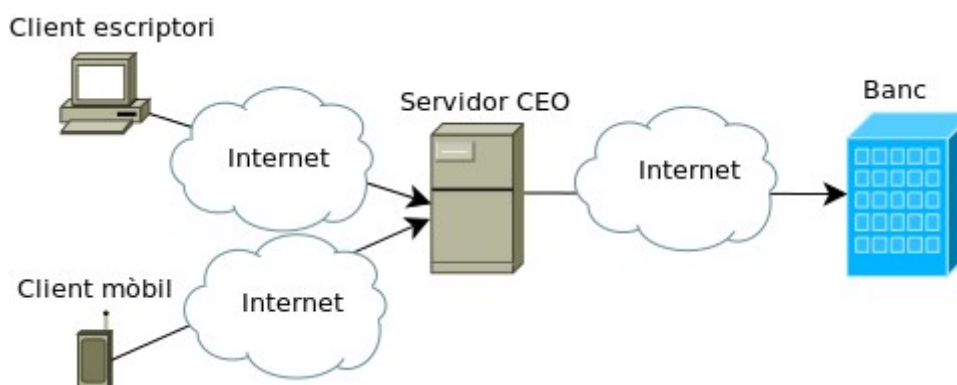
Punt de vista de components: Mostra els principals components que estructuren el sistema per tal d'ajudar a comprendre com està muntada l'arquitectura del mateix.

Punt de vista físic

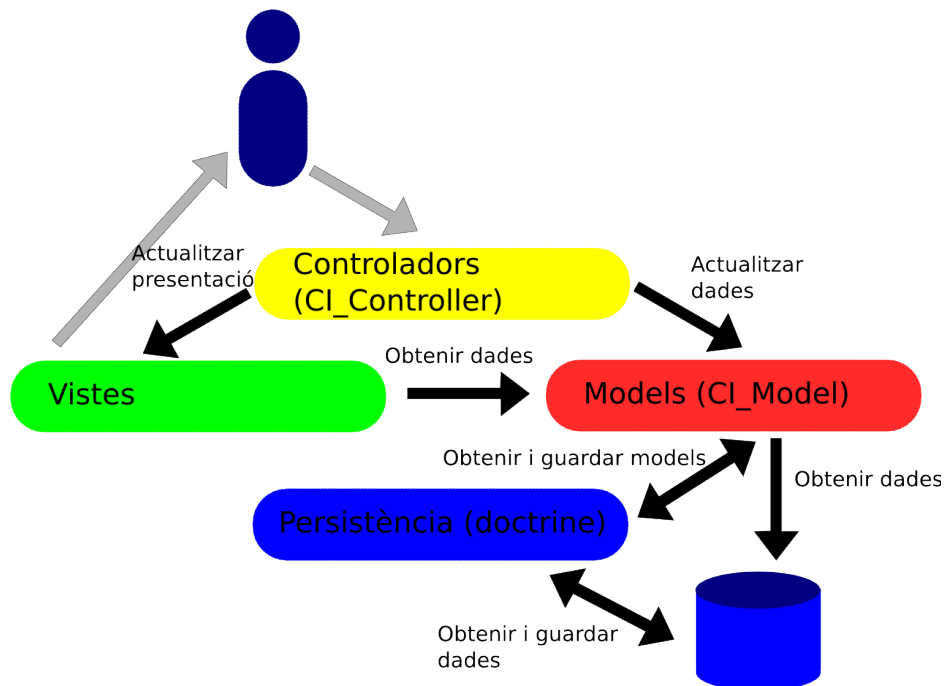
El sistema ha estat pensat per treballar en un entorn php de *hosting* compartit, i això vol dir moltes limitacions. Tot i així tampoc es descarta la futura idea d'exportar el sistema en un entorn més professional.

El sistema actualment consta d'un servidor compartit i l'accés a la passarel·la bancària de la Caixa.

El disseny de les interfícies del Frontend és simple, de forma que permeti l'accés des de qualsevol navegador, així com també des de la majoria de dispositius mòbils.



Punt de vista lògic



La vista lògica descriu els principals components i com es relacionen entre ells. L'esquema ve molt marcat per l'estructura de Model Vista Controlador que ofereix Codeigniter, tot i que s'han realitzat algunes variacions com la incorporació de Doctrine com a *framework* de persistència.

Controladors: S'encarreguen de controlar la interacció entre els models i les vistes. També són els encarregats de controlar la lògica de la navegació i la validació inicial de les dades.

Models: S'encarreguen d'obtenir les dades directament de la BD. Ho aconsegueixen de dues formes diferents:

- Utilitzant el l'Entity Manager de Doctrine que s'encarrega de fer transparent la gravació i recuperació de les dades.
- Utilitzant l'accés a dades que implementa el propi Codeigniter, que en certes ocasions resulta més ràpid, sobretot per l'extracció de llistats de la BD.

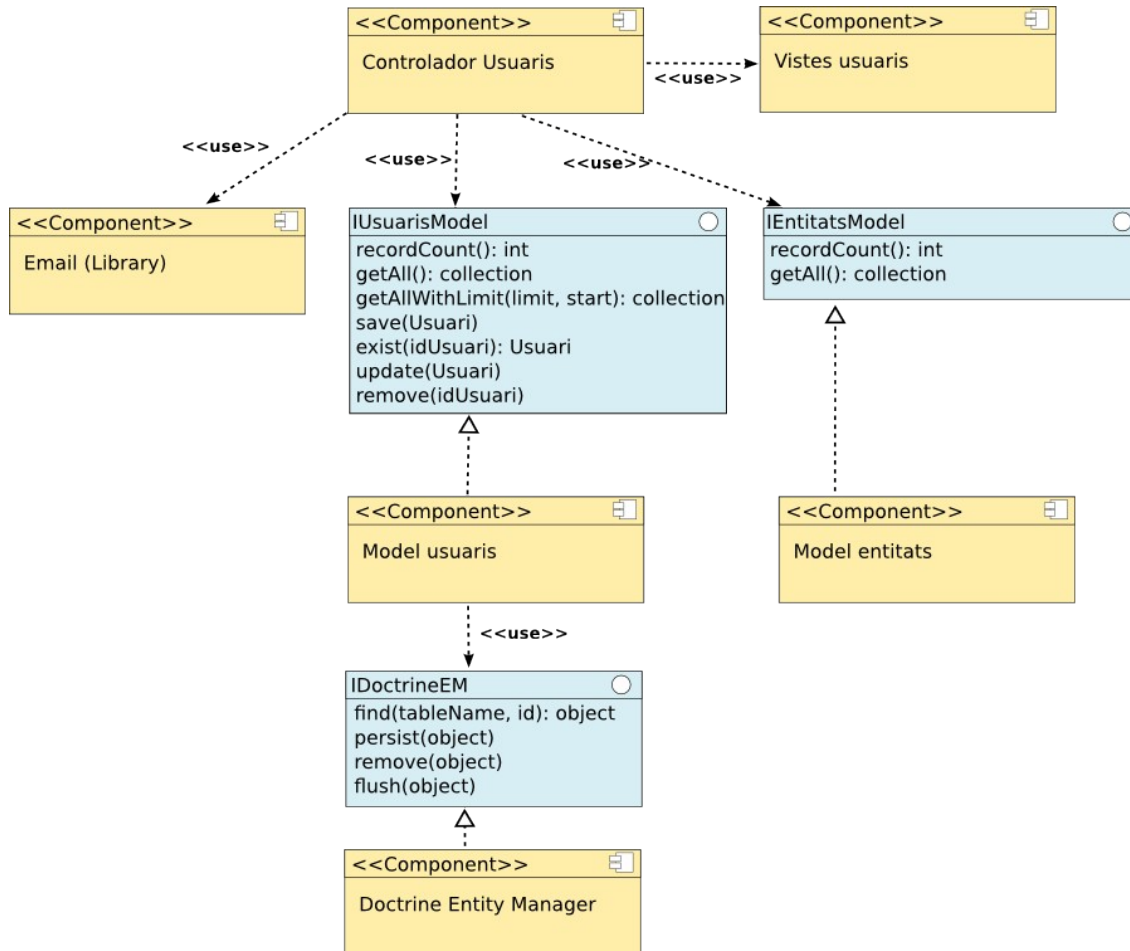
Vistes: S'encarreguen de mostrar les dades que extreu el model en forma d'entitats i estan governades pel controlador, que és qui passa les dades a mostrar. De fet no hi ha interacció directe entre la vista i el model, tot i que en el diagrama s'ha volgut mostrar que la vista extreu les dades del model en forma d'objectes d'entitat.

Persistència (doctrine): S'ha optat per incorporar dins el projecte un *framework* de persistència anomenat Doctrine, per millorar la manipulació de les dades i utilitzar el patró d'accés a dades DataMapper enlloc de usar l'ActiveRecord que pregona Codeigniter. Tot i així en alguns casos, principalment llistats, s'ha optat per utilitzar directament l'accés a dades de Codeigniter, ja que la complexitat de Doctrine i temps d'execució era massa elevada per la simplicitat del que es volia dur a terme.

Diagrama de components

En aquest diagrama es vol mostrar la interacció entre els diferents components de l'arquitectura utilitzant per fer-ho un exemple com és la gestió d'usuaris del FrontEnd.

Tot i que apareguin interfícies entre components, la crida es realitza de forma directa, i per tant d'interfície és inexistent, però s'ha cregut oportú descriure aquestes interaccions a través d'una interfície per facilitar-ne la comprensió.



Decisions tecnològiques

Llenguatge de desenvolupament

En principi l'elecció del llenguatge, ve molt condicionat per el servidor de *hosting*. La solució més extesa entre els servidors de *hosting* és el típic LAMP (Linux Apache MySQL i Php).

S'han plantejat altres alternatives com fer-ho amb Java i Tomcat, o una variant J2EE, però els costos del *hosting* eren més elevats.

Framework Web

S'han estudiat uns quants *frameworks* tal i com s'indica el l'apartat en que es descriu l'arquitectura inicial del sistema . En principi el Zend era un bon candidat, però la corva d'aprenentatge és gran i no es disposava de molt de temps material per crear el projecte. A més la documentació és una mica escassa, fet que sovint porta a revisar el codi font del propi *framework*.

Sistema gestor de BD

El sistema triat és el MySQL ja que es coneix que és un dels que millor s'integren amb PHP, tot i que es podria haver optat per una solució més professional com PostgreSQL, el volum de dades amb que s'ha de treballar no és extremadament elevat.

Persistència i accés a dades

S'ha plantejat la opció de fer servir alguna capa de persistència, i finalment s'ha elegit Doctrine, que és la

capa de persistència que implementa Symphony, un altre *framework* de PHP. S'ha seleccionat aquesta opció degut a la seva documentació excel·lent i la relativa facilitat amb la que es pot integrar amb CodeIgniter.

S'ha optat per utilitzar Doctrine en la implementació de l'accés a dades. Però s'ha limitat el seu ús només per la manipulació i consulta de registres individuals. No s'utilitzarà per tant per la selecció de registres variis com per exemple en el cas de llistats de dades. Aquesta decisió s'ha pres degut a la complexitat de les consultes via DQL (Doctrine Query Language), que tot i aportar una elegància important al codi resultant, impliquen també un cost de processament més gran.

Codis de barres de la butlleta d'inscripció

Donat que el CodeIgniter no té la opció de crear codis de barres a partir d'una cadena de text. S'ha optat per utilitzar una llibreria de les disponibles en la llibreria de classes Pear. Es tracta d'un sistema de repositori de classes amb diversos components que es poden instal·lar de forma selectiva i permeten realitzar diferents tasques. Concretament s'ha utilitzat el component Barcode2.

Presentació de l'aplicació

S'ha optat per una interfície de presentació senzilla. Les vistes contenen informació purament de presentació de dades i ho fan amb un HTML molt simple i fàcil de mantenir.

La idea consisteix en deixar tota la feina de presentació al CSS i realitzar una web que s'adapti bé tant al PC de sobretaula com als dispositius mòbils, i a poder ser sense implementar característiques addicionals per cadascun d'ells.

Implementació

Consideracions generals

S'ha decidit que el codi es trobi sempre disponible dins un repositori **GIT**, per tal de realitzar un correcte control de versions. El control de versions ajudarà a disposar sempre de les diferents versions de codi i a portar el control de les diferents fases del projecte.

Concretament s'ha creat un compte a Bitbucket per fer accessible el codi via internet.

Per descarregar-lo, simplement cal disposar del client de GIT i executar la següent ordre a través de la línia de comandes:

```
git clone https://xeviter@bitbucket.org/xeviterr/cewebinscripciones.git
```

Fase 1

Seguiment

S'ha començat la implementació del Frontend, actualment hi ha un prototip disponible.

Per descarregar el codi a partir del control de versions:

```
git clone https://xeviter@bitbucket.org/xeviterr/cewebinscripciones.git
```

Per veure la fase1 en el moment d'escriure aquest document.

```
git checkout 5503f54cc3004055bba241f9e11475f37e4433ff
```

Així mateix s'ha publicat una primera versió de la web al servidor del CEO, per tal que el client pugui fer les revisions pertinents i comenci a revisar si la solució s'adapta a les seves expectatives. Per tal d'accedir a la

web, es pot fer a través de la URL:

<http://www.ceolot.cat/embaridissada/cewebinscripcions/>

Seguiment dels riscos

El projecte ha aconseguit eliminar alguns riscos, marcats anteriorment a la taula de riscos.

Codi	Nom	Causa	Conseqüència	Nivell impacte
R001	Problemes en el coneixement del llenguatge de programació	Falta d'experiència en el llenguatge seleccionat.	Pot fer augmentar el temps d'implementació.	Resolt, s'ha aconseguit un domini suficient del llenguatge.
R002	Problemes de temps	La persona encarregada de dur el projecte a terme li falti temps.	Impossibilitat de complir les fites marcades	Mitjà
R003	Problemes amb el pagament bancari	La plataforma bancària no estigui disponible o hi hagi problemes de posta en marxa	Realització del projecte sense una de les fites marcades. Implementació posterior.	Mitjà
R004	Falta de resposta ràpida del client	Les dades que es demanen al client no estan disponibles en un temps raonable.	S'haurà d'engegar el projecte sense les dades necessàries.	Baix
R005	Problemes amb el servidor	Es tracta d'un servidor de naturalesa compartida, aquest fet pot provocar problemes en la configuració d'alguna llibreria. Fent que l'aplicació no arranqui.	Hauríem de contractar un servidor dedicat amb el conseqüent augment de preu. O bé modificar alguna llibreria, per adaptar-lo al servidor.	Resolt, s'ha instal·lat l'aplicació en el servidor i el comportament és correcte.

Fase 2

Seguiment

En la fase 2 han aparegut molts canvis, degut al feedback rebut per part del client. Aquest fet en si es bo, ja que minimitzem el risc, però ha obligat a fer més canvis dels previstos inicialment.

S'ha parlat amb el client, i s'ha acordat que les excursions de tipus Embardissada han de tenir la possibilitat de gestionar diferents etapes. Aquest fet ha provocat que algunes parts de l'anàlisi hagin variat per adaptar-se a aquests nous requisits. Les etapes poden ser totes obligatòries o bé es pot participar només en unes quantes.

També s'ha comentat que totes les excursions, haurien de tenir alguns paràmetres opcionals, a fi de ser més flexibles:

- **Opció no pagament:** S'hauria de poder realitzar excursions que no necessitessin un pagament previ i simplement fos una forma ràpida de recollir les dades de l'usuari.
- **Opció assegurança:** Caldria que l'assegurança fos opcional i que no en totes les excursions hi aparegués.
- **Opció només federats:** Per inscriure's a una excursió és necessari disposar de carnet de la federació.
- **Opció tarifes anticipades:** Caldria que les tarifes a abans d'una determinada tinguessin un preu diferent, habitualment més barat.

Seguiment dels riscos

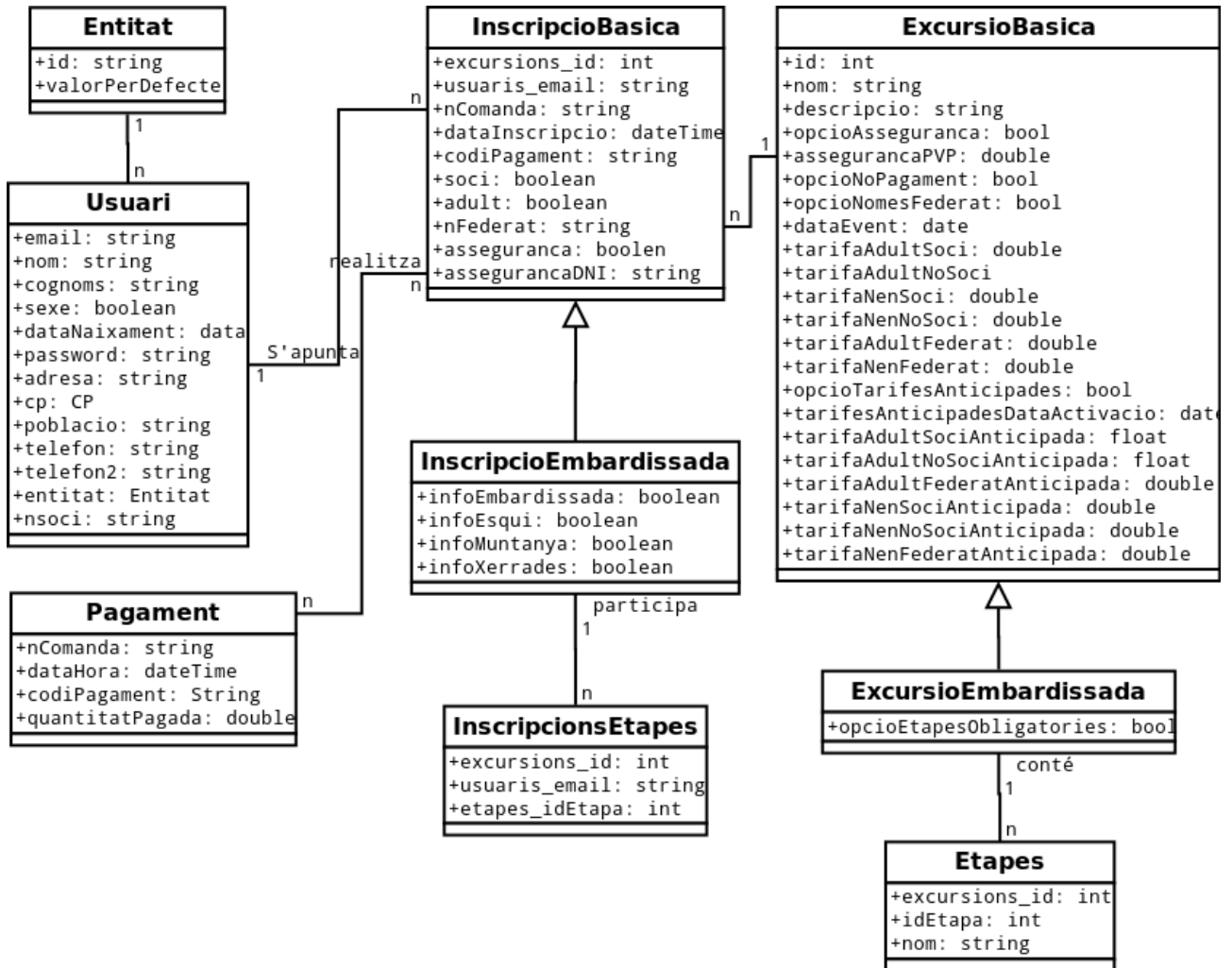
El projecte ha aconseguit eliminar alguns riscos, marcats anteriorment a la taula de riscos.

- Cal destacar que el client ha tardat bastant en comprovar el prototip de la fase 1 i a aportar informació de "feedback", per tant el risc R004, s'incrementa.

Codi	Nom	Causa	Conseqüència	Nivell impacte
R001	Problemes en el coneixement del llenguatge de programació	Falta d'experiència en el llenguatge seleccionat.	Pot fer augmentar el temps d'implementació.	Resolt, s'ha aconseguit un domini suficient del llenguatge.
R002	Problemes de temps	La persona encarregada de dur el projecte a terme li falti temps.	Impossibilitat de complir les fites marcades	Mitjà
R003	Problemes amb el pagament bancari	La plataforma bancària no estigui disponible o hi hagi problemes de posta en marxa	Realització del projecte sense una de les fites marcades. Implementació posterior.	Resolt, la plataforma de proves funciona correctament.
R004	Falta de resposta ràpida del client	Les dades que es demanen al client no estan disponibles en un temps raonable.	S'haurà d'engegar el projecte sense les dades necessàries.	Mitjà
R005	Problemes amb el servidor	Es tracta d'un servidor de naturalesa compartida, aquest fet pot provocar problemes en la configuració d'alguna llibreria. Fent que l'aplicació no arranqui.	Hauríem de contractar un servidor dedicat amb el conseqüent augment de preu. O bé modificar alguna llibreria, per adaptar-lo al servidor.	Resolt, s'ha instal·lat l'aplicació en el servidor i el comportament és correcte.

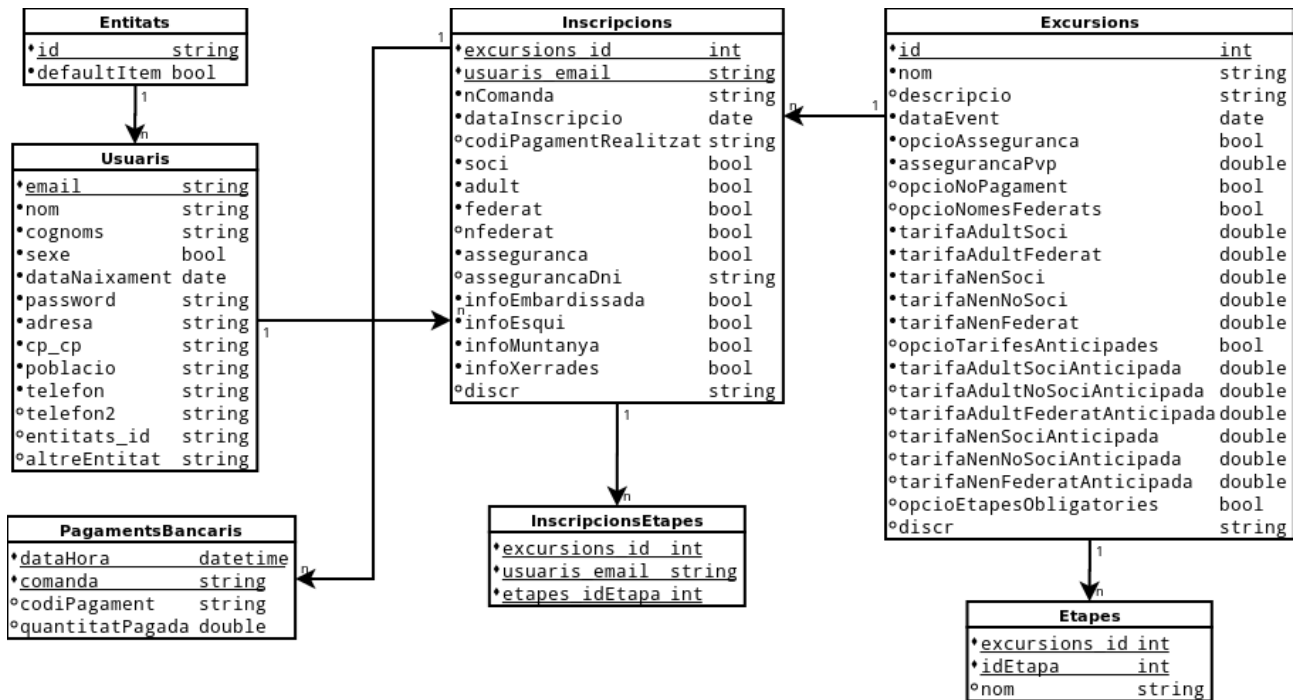
Refinament del diagrama estàtic de classes

Com a resultat dels canvis s'han modificat els següents diagrames per adaptar-los a la nova situació.



Un fet destacable és que s'han afegit els nous atributs de cada classe dins el model estàtic de classes. També s'han afegit dues noves entitats InscripcionsEtapes i Etapes, que afecten només als objectes de tipus ExcursioEmbardissada i InscripcioEmbardissada.

Refinament del diagrama model relacional



Aquest diagrama ha sofert canvis importants:

El més destacable es que s'ha reduït el nombre de taules, i s'ha optat per un model d'herència d'una sola taula, degut a les moltes característiques comunes que comparteixen els dos tipus d'excursions. S'ha optat per unir les taules InscricionsTipusBasiques i inscricionsTipusEmbardissada en una taula inscricions i el mateix per les dues taules excursions. S'ha afegit un camp anomenat disc (discriminador) que determina el tipus d'herència d'un determinat registre dins la taula inscricions i excursions.

S'han afegit les noves taules InscricionsEtapes i Etapes, que controlen les etapes disponibles d'una excursió de tipus Embardissada i en quines etapes s'han inscrit els diferents usuaris.

Finalment s'han afegit els camps corresponents als canvis demanats per l'usuari i que ja es trobaven reflexats al diagrama estàtic de classes.

Fase 3

Seguiment

En la fase3 han aparegut diverses millores sobretot pel que fa a la part visual. També s'ha aprofitat per fer una refactorització de codi per preparar l'aplicació i fer-la més fàcil als canvis futurs.

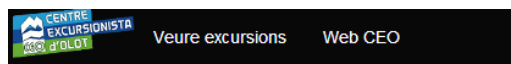
Millora presentació visual

Tot i que inicialment s'havia acordat que la web de Frontend havia de ser simple per tal de mantenir la compatibilitat amb els dispositius mòbils. El client ha demanat si és possible millorar-ne la presentació visual, a fi de millorar-ne l'atractiu i la facilitat d'ús en el mòbil.

En aquesta fase s'han produït bàsicament alguns canvis en el disseny de les vistes de la web, s'ha intentat adaptar-la al disseny mòbil, però de forma més atractiva a la vista. Per fer-ho s'ha seguit la filosofia de disseny "responsive", que consisteix en adaptar el contingut de la web al dispositiu de visualització a través de CSS i Javascript. Per fer l'adaptació més simple s'ha utilitzat un *framework* de presentació anomenat Bootstrap, que simplifica enormement la feina i permet aconseguir uns resultats increïbles amb molt poc codi.

Alguns dels punts que s'han millorat considerablement han estat:

- Gestió de formularis, s'han adaptat al model que ofereix Bootstrap i que fa els controls més grans i accessibles quan es fa click en un dispositiu mòbil.
- Millora de les validacions de formulari, s'han col·locat els missatges d'error en les validacions del sistema en un requadre de color vermell a la part superior del formulari, fent-los molt més visibles.
- Creació de menús de l'aplicació a la part superior dels formularis, tant en la part de Frontend com en la part de Backend. Aquesta millora ha comportat un augment en l'accessibilitat de la web.



Introdueix les teves dades

El camp email és obligatori.
El camp telefon és obligatori.

* Correu electrònic:

* Nom:

* Cognoms:

Generalització en els tipus d'excursions

El client, ha demanat la possibilitat de treballar amb tipus d'excursions diferents al tipus base. Concretament ha demanat la possibilitat de realitzar inscripcions molt simples que puguin inscriure una persona sense demanar dades addicionals i sense incloure pagament.

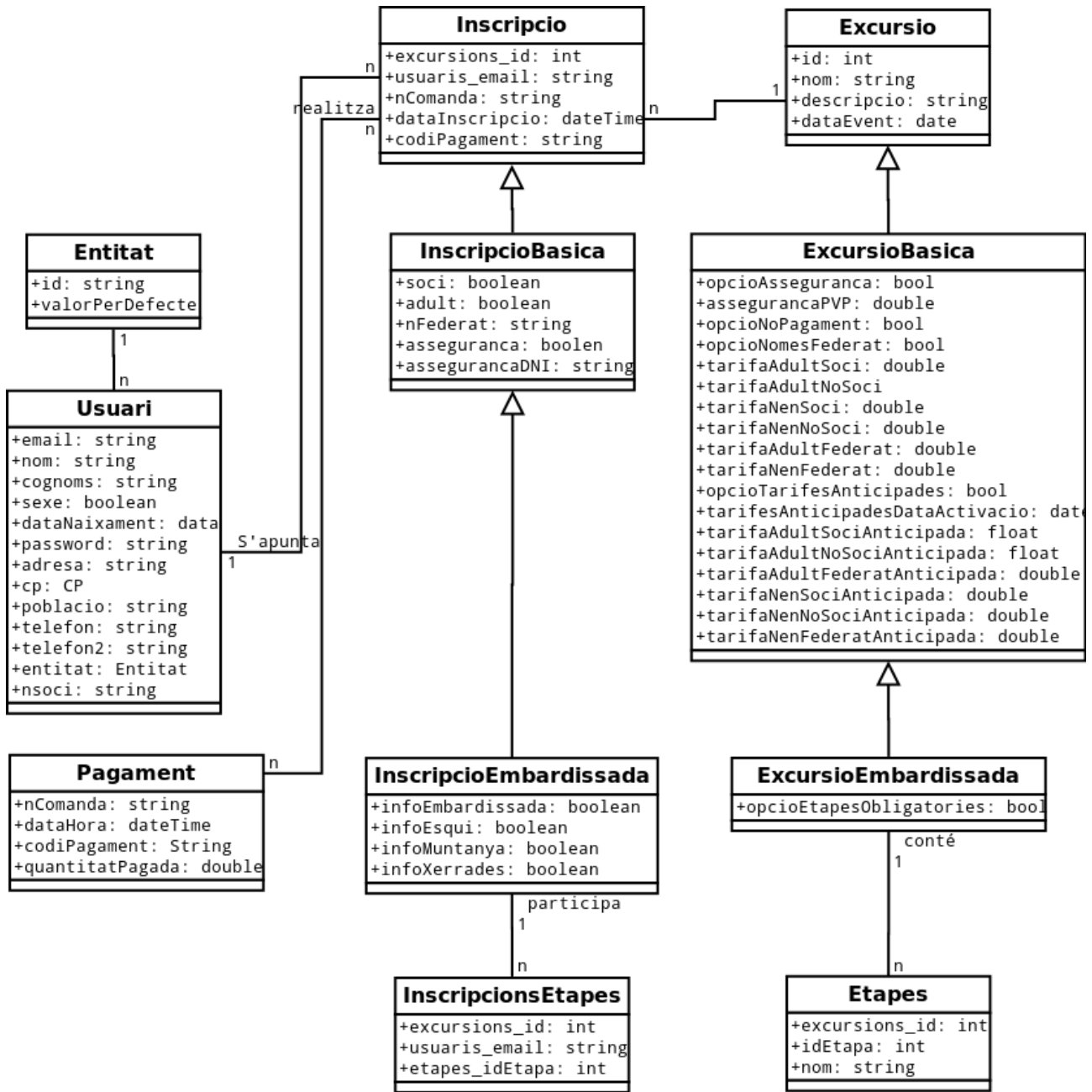
També ha demanat si seria possible en un futur utilitzar la web per a realitzar inscripcions d'altres seccions

del centre excursionista amb necessitats més específiques, com podrien ser curses d'orientació o bé la secció de mainada.

Inicialment s'havia plantejat un tipus d'excursió bàsica massa especialitzat a les necessitats inicials del centre, amb moltes opcions configurables i aquest fet ha provocat que s'hagi hagut de modificar la jerarquia d'herència dels diferents tipus d'excursions creant una excursió molt simple amb les característiques que tota excursió ha de tenir. Concretament el nom, la descripció i la data de l'event. Aquest fet ha suposat una refactorització de codi important, que ha afectat també a les inscripcions, però no ha afectat massa la funcionalitat.

La creació del tipus excursió permetrà desenvolupar nous tipus d'excursió a petició del client, d'una forma molt més simple i aprofitant al màxim el codi, gràcies als mecanismes de l'herència.

Refinament del diagrama estàtic de classes

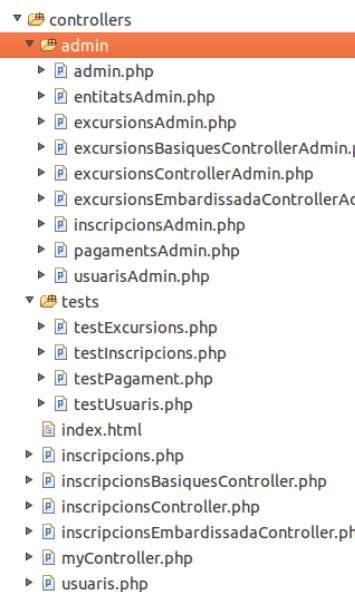


Detalls de la implementació

Controladors

Tot el codi de l'aplicació es troba distribuït seguint el model vista controlador seguint l'estructura que mostra la documentació de CodeIgniter.

Pel que fa als controladors, es troben distribuïts en tres directoris diferents. Els que es troben dins "admin", fan referència a l'administració o "backend" de l'aplicació. Els que es troben dins "tests", fan referència als diferents tests unitaris i d'integració de l'aplicació. Finalment els controladors que es troben a l'arrel corresponen als controladors de "frontend".



Els diferents controladors s'han agrupat segons la funcionalitat. Així per exemple el controlador "usuarios.php" incorpora totes les operacions que impliquen treballar amb l'usuari. Com son:

```
class Usuarios extends MyController {  
...  
    public function identificar($urlContinuacio)  
    public function nou($urlContinuacio)  
    public function autenticar()  
    public function paraulapasoblidada()  
    public function modificarparaulapas($idEnllac)  
    public function modificarDades($urlContinuacio, $email)  
    public function tancarsessio()  
}
```

Pel que fa a l'herència, la majoria de controladors hereden de **MyController**, aquest controlador hereta directament del controlador **CI_Controller**, que és el que proporciona CI, però hi afegeix la documentació dels seus objectes per fer-ne més fàcil el tractament.

Cal destacar que un controlador en CodeIgniter conté a dins seu un *singleton* o super objecte que conté referències a totes les classes instanciades per CI. Aquest super objecte té com a objectiu eliminar els problemes de *scoping* o visibilitat de les variables en tot el programa, a part d'evitar duplicar certs objectes. Per tant per treballar directament amb les diferents llibreries de CI, sempre caldrà tractar-les com a propietats d'un objecte (\$this).

Models i patrons de disseny

- ▼ 📁 models
 - ▶ 📁 Entity
 - ▶ 📄 entitatsmodel.php
 - ▶ 📄 excursionsbasiquesmodel.php
 - ▶ 📄 excursionsembardissadamodel.php
 - ▶ 📄 excursionsmodel.php
 - ▶ 📄 index.html
 - ▶ 📄 inscripcionsbasiquesmodel.php
 - ▶ 📄 inscripcionsembardissadamodel.php
 - ▶ 📄 inscripcionsmodel.php
 - ▶ 📄 mymodel.php
 - ▶ 📄 pagamentsmodel.php
 - ▶ 📄 selectorexcursionsmodel.php
 - ▶ 📄 selectorinscripcionsmodel.php
 - ▶ 📄 usuariamodel.php

Els models de l'aplicació es troben tots dins la mateixa carpeta i son utilitzats per els diferents controladors, tant els de Backend" com els de Frontend.

Per construir els models de l'aplicació s'ha utilitzat dos patrons de disseny diferents:

El primer patró de disseny és Active Record i representa la filosofia de CI, per extreure la informació dels llistats de la BD. Té com avantatge la seva rapidesa ja que treballa molt pròxim a la BD. Aquest patró s'ha utilitzat en tota mena de llistats i consultes que implicaven extreure més d'un registre de la BD.

El segon patró de disseny el Data Mapper, que implementa Doctrine. Aquest patró s'ha utilitzat per extreure informació, crear, modificar i eliminar objectes individuals, com ara: usuaris, inscripcions, excursions, entre d'altres. Té l'avantatge de ser molt flexible a l'hora de ser utilitzat i permet una millor separació del codi de persistència i l'objecte en memòria. Tot i així té una penalització pel que fa a la rapidesa, com la tenen la majoria de ORM's.

Un exemple de model d'usuari, en que es pot veure la utilització dels dos patrons:

```
class UsuarisModel extends MyModel {
...
//Funció que retorna totes els registres de la taula d'usuari
//segueix el model Active Record de CI.

function get_all(){
    $this->db->select('usuaris.*');
    $this->db->from('usuaris');

    return $this->db->get()->result();
}

//Funció que donat un objecte usuari en memòria, l'actualitza
//dins la base de dades utilitzant EntityManager (em).
//segueix el model Data Mapper de Doctrine2.
function save($usuari){
    $this->em->persist($usuari);
    $this->em->flush();
}
}
```

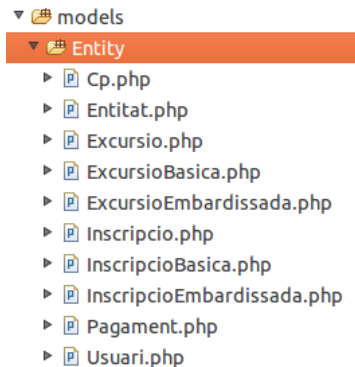
Cal destacar també que s'ha implementat el mecanisme d'herència que ofereix Doctrine, per la creació dels diferents tipus d'excursions i inscripcions. Aquest mecanisme fa que la implementació de l'herència sigui transparent pel programador a l'hora de guardar i modificar les dades.

En el següent exemple es pot veure com es declara l'herència dins Doctrine per l'entitat excursió i totes les seves classes derivades ExcursioBasica i ExcursioEmbardissada.

```
/**
 * @Entity
 * @Table(name="excursions")
 * @InheritanceType("SINGLE_TABLE")
 * @DiscriminatorColumn(name="discr", type="string")
 * @DiscriminatorMap({"Excursio" = "Excursio", "ExcursioBasica" =
"ExcursioBasica", "ExcursioEmbardissada" = "ExcursioEmbardissada"})
 */
class Excursio
{
    /** @Id @Column(type="integer", length=11)
     * @GeneratedValue(strategy="AUTO")
     */
    private $id;

    /** @Column(type="string", length=50) */
    private $nom;
    /** @Column(type="text", nullable=true) */
    private $descripcio;
    ...
}
```

Objectes del domini



Els objectes de domini, es troben dins el directori dels models. Son simplement representacions dels objectes de la BD en memòria. Aquests objectes s'han definit manualment, però els *setters* i *getters* s'han generat gràcies a Doctrine2 de forma automàtica a través d'un assistent.

Aquí podem veure un exemple de part de la classe Usuari. Cal destacar que les definicions que utilitza doctrine per fer el mapeig amb la BD es troben declarats en un format específic dins els comentaris de PHPDoc.

```
/**
 * @Entity @Table(name="usuaris")
 */
class Usuari
{
    /** @Id @Column(type="string", length=50) */
    protected $email;
    /** @Column(type="string", length=25) */
    protected $nom;
    /** @Column(type="string", length=50) */
    protected $cognoms;
    /** @Column(type="string", length=1) */
    protected $sexe;
    /** @Column(type="date") */
    protected $datanaixament;
    /** @Column(type="string", length=25, nullable=TRUE) */
    protected $altreentitat;
}
```

Vistes

Les vistes segueixen el funcionament detallat en la guia del programador de CI. És a dir son simples fitxers de PHP que incorporen petites porcions de codi HTML, tot i així no implementen cap mena de lògica de negoci i es centren només en la presentació de les dades.

Les vistes es troben distribuïdes en tres directoris diferents tots dins el directori view:

- **Directori arrel:** Conté totes les vistes que fan referència al front-end.
- **Admin:** conté totes les vistes que fan referència exclusiva al back-end.
- **Comú:** conté aquelles vistes o subvistes compartides entre el front-end i el back-end.

En CodeIgniter, realment no existeix una instància com a tal de l'objecte vista. Segons la seva pròpia documentació el que es fa és un "include" d'un fitxer amb la informació que volem passar-li a la vista.

Aquest model a vegades resulta complicat de tractar, sobretot si les dades que s'han de traslladar a la vista són complexes o formades a la vegada per subvistes. La solució que s'ha adoptat finalment, després de moltes proves, ha estat crear la vista complexa a partir del controlador.

A continuació podem veure un exemple la crida d'una vista formada per subvistes. Concretament existeix la vista **header** i la vista **footer**, ambdues són subvistes que es poden substituir per altres de forma ràpida.

Aquest codi es situaria dins en el controlador d'usuaris.

```
public function helper_editar($email, $errors='')
{
    //Preparació de les dades de la vista principal.
    $data['usuari'] = $this->usrModel->get_by_id($email);
    $data['url'] = 'admin/usuarisAdmin/modificarPassword';
    $data['url_tornar'] = 'admin/usuarisAdmin';
    $data['errors'] = $errors;

    //Preparació de la vista de capçalera
    $dataHeader['mnuUsuarisAdmin'] = true;
    $data['header'] = $this->load->view('admin/header', $dataHeader, true);

    //Preparació de la vista del peu
    $data['footer'] = $this->load->view('admin/footer', null, true);
    $this->load->view('admin/usuaris_editar.php', $data);
}
```

Aquí es pot veure una part de la vista cridada en l'apartat anterior. Cal destacar sobretot les variables **\$header** i **\$footer**, que estan formades per les subvistes renderitzades.

```
<?php
echo $header;
?>
<h1>CEWebExcursions Backend - Gestió d'usuaris</h1>
<form method="post" class="form well" action="<?php echo base_url( $url );?>">
...
    <label for="email">* Correu electrònic:</label>
    <div><?php echo $usuari->getEmail() ?></div>
    <br><label for="nom">Nom i cognoms:</label>
    <div><?php echo ($usuari->getNom() . " " . $usuari->getCognoms());?></div>
    <br> <label for="datanaixament">Data de naixament (dd/mm/aaaa):</label>
    <div><?php echo $usuari->getDataNaixament()->format('d/m/Y'); ?></div>
...
</form>
<?php
echo $footer;
?>
```

Una altre solució, que s'ha adoptat en les fases finals del projecte i que resulta útil, si la vista és extremadament complexa. Consisteix en crear una classe que s'encarregui de generar un objecte que s'encarregui d'instanciar els diferents elements que incorpora la vista. D'aquesta forma es possible reutilitzar la vista d'una manera senzilla passant només uns quants paràmetres.

La crida des del controlador queda molt simplificada.

```
$data['pv'] = new PVInscripcionsLlistatExcursio(
    $this, $excursio->getNom(),
    $inscripcions, $paginacio, $urlBase, $urlExportacio);
$this->load->view ( 'admin/inscripcions_llistat_excursio.php', $data );
```

Aquí podem veure com seria l'objecte PVInscripcionsLlistatExcursio

```
class PVInscripcionsLlistatExcursio
{
    public $nomExcursio;
    public $paginacio;
    public $urlExportacio;
    public $urlBase;
    public $header;
    public $footer;
    public $llistatCapcaleresInscripcions;
    public $llistatValorsInscripcions;

    public function __construct($controlador, $nomExcursio,
        $inscripcions, $paginacio, $urlBase, $urlExportacio)
    {
        $this->nomExcursio = $nomExcursio;
        $this->paginacio = $paginacio;
        $this->urlExportacio = $urlExportacio;
        $this->urlBase = $urlBase;

        $dataHeader['mnuInscripcionsAdmin'] = true;
        $this->header = $controlador->load->view(
            'admin/header', $dataHeader, TRUE);
        $this->footer = $controlador->load->view(
            'admin/footer', null, TRUE);

        $dataLlistatCapcaleres['url'] = $this->urlBase;
        $this->llistatCapcaleresInscripcions =
            $controlador->load->view(
                'admin/parts/inscripcions_llistat_capcaleres.php',
                $dataLlistatCapcaleres, true);

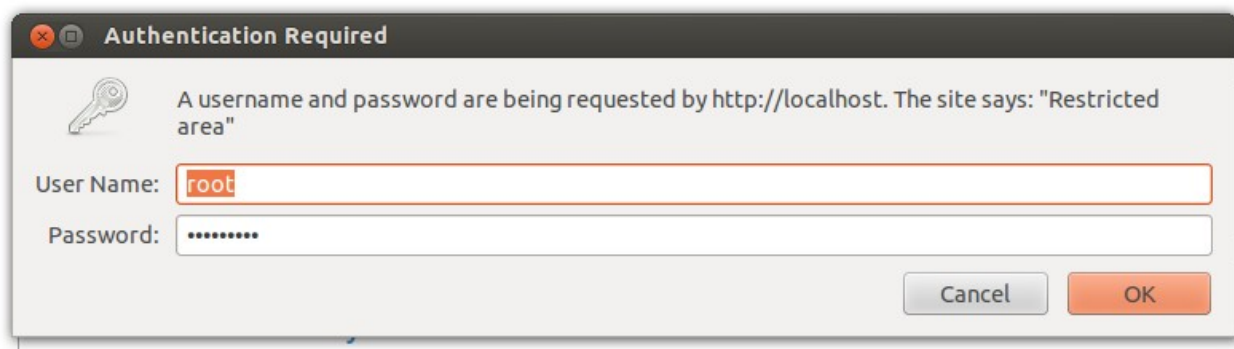
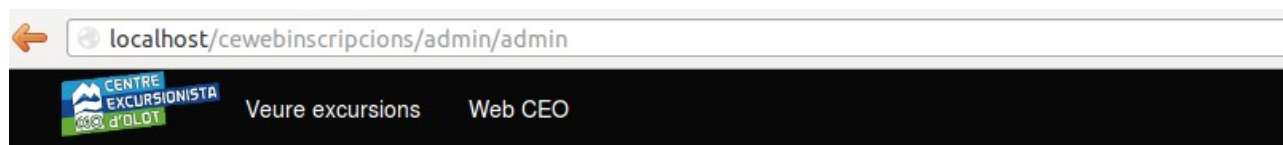
        $campsLlistatInscripcions = array();
        foreach ($inscripcions as $inscripcio)
        {
            $tmp['inscripcio'] = $inscripcio;
            $this->llistatValorsInscripcions[] =
                $controlador->load->view(
                    'admin/parts/inscripcions_llistat_camps', $tmp, true);
        }
    }
}
```

L'ús del model dins la vista també queda molt més simplificat i més clar, ja que tots els accessos es fan a través d'un sol objecte.

```
<?php
/* @var $pv PVInscripcionsLlistatExcursioBasica */
echo $pv->header;
```

```
?>
<h1>Inscrits excursió <?php echo $pv->nomExcursio; ?></h1>
<div id="contingut">
...
<p><?php echo $pv->paginacio; ?></p>
<table class="table table-condensed">
  <tr>
    <?php echo $pv->llistatCapcaleresInscripcions; ?>
  </tr>
  <?php
  foreach ($pv->llistatValorsInscripcions as $campsInscripcio)
  {
    echo '<tr>';
    echo $campsInscripcio;
    echo '</tr>';
  }
  ?>
</table>
...
</div>
<?php echo $pv->footer;?>
```

Autenticació d'usuari en el Backend

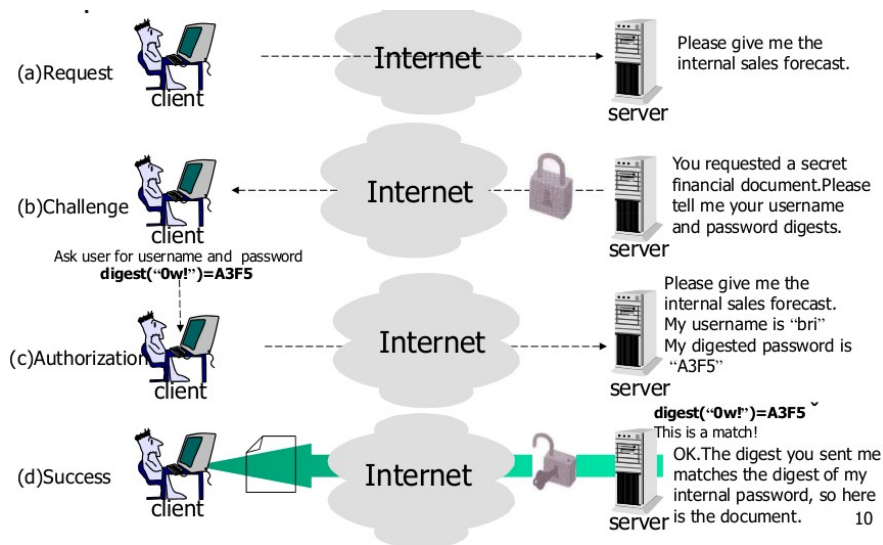


L'autenticació d'usuari en el Backend es realitza a través del mètode DIGEST que proporciona el servidor web. Inicialment s'havia pensat en utilitzar HTTPS/TLS i l'autenticació bàsica, passant totes les dades xifrades, però degut al tipus de servidor *hosting* contractat no ha estat possible.

Una solució viable, ha estat crear una autenticació basada en digest. Que no soluciona que la informació viatgi en pla, però almenys minimitza el risc que qualsevol usuari a través d'un "sniffer" tingui accés als passwords d'administració de la web de forma instantània i per tant obtenir accés a tota l'administració de l'aplicació.

L'autenticació mitjançant digest té com a idea fonamental no enviar el password a través de la xarxa. En el seu lloc el client envia un resum o digest del password, de forma que a partir del resum no es pot obtenir la seva clau original. El servidor per la seva banda coneix el password i l'únic que fa és fer un resum del mateix i comparar els resultats, si els dos resums son idèntics, hi ha un % molt elevat que sigui el mateix password i la possibilitat de col·lisió és tant baixa que és pràcticament tant difícil com obtenir el password a través de la força bruta.

En la següent imatge, podem veure la representació d'una comunicació realitzada amb digest. Extreptes de les transparències de Heng-Yow Chen que parlen del mateix tema.



El mètode digest a part de no enviar el password directament, tampoc envia el resum directament ja que un possible atacant podria obtenir aquest resum i reproduir-lo tantes vegades com fes falta per entrar al servidor.

La forma de prevenir aquest tipus d'atac és mitjançant l'enviament d'un *token* anomenat *nonce* (un numero únic) i que es combina amb el password per fer el resum que s'enviarà al servidor. D'aquesta forma si l'atacant captura el resum serà un resum només vàlid per una determinada connexió, i per tant s'evita la possibilitat de replicació del resum del password tantes vegades com faci falta.

La implementació de la connexió a través de digest està encapsulada dins una llibreria de CodeIgniter. La funció més important és la de `tryLogin`, que intenta validar un usuari en el sistema.

Si tot funciona correctament la classe realitza els càlculs del MD5 i els compara amb la informació enviada per el client. Si quelcom va malament torna a enviar una petició de connexió (funció `sendHeaders`).

```
class DigestAuthentication
{
    ...

    function tryLogin($users, $realm = 'Restricted area')
    {
        if (empty($_SERVER['PHP_AUTH_DIGEST'])) {
            $this->sendHeaders($realm);
            die('Authentication required, cancel button pressed');
        }

        // Analitza la variable PHP_AUTH_DIGEST, que conté la informació
        // que ha enviat l'usuari.
        if (!$data =
            $this->http_digest_parse($_SERVER['PHP_AUTH_DIGEST'])) ||
            !isset($users[$data['username']]))
        {
            $this->sendHeaders($realm);
            die('Wrong Credentials!');
        }

        // Genera una resposta vàlida, realitzant resums MD5
    }
}
```

```
// a partir de certes dades de l'usuari.
$A1 = md5($data['username'] . ':' . $realm . ':' .
    $users[$data['username']]);
$A2 = md5($_SERVER['REQUEST_METHOD'] . ':' . $data['uri']);
$valid_response = md5($A1 . ':' . $data['nonce'] . ':' .
    $data['nc'] . ':' . $data['cnonce'] . ':' . $data['qop'] . ':' . $A2);

//Comprovem que el que s'ha enviat xifrat amb MD5
//coincideixi amb la informació local.
if ($data['response'] != $valid_response) {
    $this->sendHeaders($realm);
    die('Wrong Credentials!');
}
return $data;
}
```

Per fer encara més senzilla la validació de l'usuari, s'ha optat per la creació d'un *helper*, que amb una sola línia de codi realitza aquesta funció en tots els controladors de l'administració.

```
/**
 * Valida l'accés a les pàgines d'administració.
 */
function adminValidation() {
    $CI =& get_instance();
    $CI->load->library('DigestAuthentication');

    $auth = new DigestAuthentication ();
    $auth->tryLogin ( array (
        $CI->config->item('admin_user') =>
        $CI->config->item('admin_password')
    ) );
}
```

Autenticació d'usuari en el Frontend

Per autenticar l'usuari en la part del Frontend s'ha utilitzat l'autenticació bàsica a través de formularis HTML. Aquest mètode a nivell d'usabilitat és millor que el comentat a l'apartat anterior, ja que s'integra en l'estructura de la web. En futures versions quan es contracti un servidor amb capacitat HTTPS, aquest problema quedarà resolt. Ja que HTTPS xifra tota la connexió i evita escoltes indesitjades.



Identifica't

Usuari ja registrat

Ja consto al registre i em vull inscriure a l'excursió.

Correu electrònic: <input type="text" value="Correu electrònic"/>
Paraula de pas: <input type="text" value="Paraula de pas"/>
<input type="button" value="Entrar"/>

Per crear un nou compte fes click [aquí](#)

El client va demanar de forma explícita en els requisits, que la part del Frontend, s'havia de comportar com un assistent per registrar-se en una excursió, i així es va expressar en el prototip.

El problema d'aquest comportament és que un cop seleccionada l'excursió en la pantalla inicial, si l'usuari no està registrat, es demana l'autenticació. Aquest fet és un problema per continuar després des del punt on s'havia deixat. El problema a més quedava agreujat amb el fet que hi hagi diferents controladors per diferents tipus d'excursió.

Per solucionar-ho s'ha optat per passar la URL de continuació al formulari d'autenticació codificada en base64 i d'aquesta forma simular el comportament d'un assistent.

Així per exemple la URL:

```
http://localhost/cewebinscripcions/inscripcionsEmbardissadaController/dadesexcursio/1032
```

Passa a formar part de la crida a l'autenticació, codificada en base64.

```
http://localhost/cewebinscripcions/index.php/usuaris/identificar/aHR0cDovL2xvY2FsaG9zdC9jZXd1Ym1uc2NyaXBjaW9ucy9pbmNjcmlwY21vbWVhbnNFbWJhcjR3ZGFDb250cm9sbGVyL2RlZGVzZXhjZaW8vMTAzMg%3D%3D
```

Posteriorment i un cop l'usuari hagi estat autenticat, aquesta URL es descodificada i s'envia l'usuari a la inscripció de l'excursió que havia demanat.

Recuperació de la paraula de pas

El mecanisme per recuperar la paraula de pas, és bastant habitual consisteix en 3 passos.

L'usuari introdueix el seu correu electrònic, que al mateix temps és l'identificador d'usuari.

He oblidat la paraula de pas

Si ja tens un compte, rebràs informació de com recuperar-la al teu compte de correu.

Correu electrònic:

A continuació s'envia un correu al compte de l'usuari, sempre i quan estigui prèviament registrat. Amb l'enllaç per modificar la paraula de pas. Aquí es pot veure un exemple del correu rebut a la bústia d'entrada.

Recuperació de paraula de pas  Safata d'entrada x

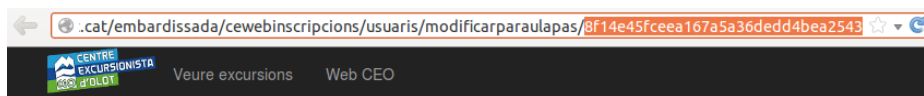
 **ceo@ceolot.org** <ceo@ceolot.org> 11.55 (Fa 0 minuts)
per a usuari ▾

Fes click al següent [enllaç](#) per modificar la paraula de pas.

Si es clica a l'enllaç, apareix la pàgina de canvi de claus. Cal destacar sobretot el número del final de l'adreça. El número és el resultat d'aplicar una funció de hash MD5 a les següents dades.

```
md5($usuari->getEmail() +  
'passchange' +  
$usuari->getDataNaixament()->format('d/m/Y'))
```

Abans de canviar la paraula de pas, el sistema comprovarà que el resultat de la funció de hash enviada concordi amb el resultat calculat pel mateix sistema. Si és així es permetrà el canvi d'adreça. Altrament el sistema donarà un error. Aquest mètode és útil per prevenir possibles intents de canvi de password no autoritzats.



Introdueix les noves dades

* Correu electrònic:

* Nova paraula de pas:

* Repetiu la nova paraula de pas:

Pagament bancari

El pagament bancari és executat per el controlador d'inscripcions (InscripcionsBasiquesController). Algunes de les funcions de pagament es troben encapsulades dins una llibreria (Payments).

Així per exemple per enviar les dades del pagament a l'entitat bancària es genera un conjunt de camps de tipus "hidden" dins un formulari que han de ser enviats a la passarel·la de pagament.

A continuació es pot veure l'exemple de crida a la llibreria Payments. Cal destacar que és necessari enviar a la funció la quantitat a pagar, el correu de l'usuari i el número de comanda. La resta de valors corresponen a les claus de xifratge de l'algoritme de xifrat de clau pública SHA i les URL's de comprovació del pagament, encert i error.

```
Payments::process_button (
    $quantitat, $email, $nComanda,
    $this->config->item ( 'module_payment_caixa_sha_id_primaria' ),
    $this->config->item ( 'module_payment_caixa_sha_id_com' ),
    base_url ( '/' . $this->getControllerName() . '/comprovaPagament' ),
    base_url ( '/' . $this->getControllerName() . '/pagamentOk' ),
    base_url ( '/' . $this->getControllerName() . '/pagamentKO' ));
```

Cal destacar com ja s'ha comentat en apartats anteriors, que el resultat del pagament serà enviat per part del banc de forma asíncrona, és a dir, en una connexió diferent a la de l'usuari. Aquest pagament ha de ser capturat per la pròpia web i processat.

A continuació es mostra un fragment de les funcions que s'encarreguen d'aquesta tasca. És important remarcar que s'analitzen els codis d'error segons el manual de "La Caixa", que indica que si la transacció cau entre 0 i 99 es considera aprovada. Tot i així, en tots els casos s'anota el resultat de la transacció per fer front a possibles reclamacions per part del client.

Finalment si la transacció no ha resultat en error, s'assigna un número de dorsal (helper_inscriure) i s'envia un correu al client notificant el resultat de la transacció.

```
public function comprovaPagament()
{
    ...

    $this->helper_comprovaPagament($_POST['Ds_Response'],
        $_POST['Ds_Order'], $_POST['Ds_Amount']);
}

protected function helper_comprovaPagament(
    $ds_response, $ds_order, $ds_amount)
{
    //Pagament bancari.
    $error = false;

    if($ds_response == "") {
        $error = true;
    }
    else
    {
        $codiError = (int) $ds_response;
        if (!$codiError >= 0 && $codiError <= 99 )
            $error = true;
    }
}
```

```
//Alerta el pagament pot ser repetit.
$dataHoraPagament = new DateTime();
$dataHoraPagament->format('Y/m/d');
$pagament = $this->pagaModel->createNew($ds_order, dataHoraPagament);
$pagament->setCodipagament($ds_response);
$pagament->setQuantitatpagada($ds_amount);

$this->pagaModel->save($pagament);
}

$inscripcio = $this->getInscripcionsModel()->get_by_ncomanda($ds_order);
if ($inscripcio == null)
    $error = true;

if (!$error)
{
    $this->helper_inscriure($ds_order);
    $this->payments->enviarCorreuPagament($inscripcio);
}
return (!$error);
}
```

Evitant nº d'inscripció repetits i concurrència

Per tal d'evitar duplicitats en el número d'inscripció o de dorsal tal i com s'havia detectat en fases inicials, s'ha decidit utilitzar una zona d'exclusió mútua a través d'un *Mutex*, que garanteix que no s'executarà un mateix fragment de codi simultàniament.

```
public function anotarPagamentInscripcio($objInscripcio,
$modelInscripcio, $db)
{
    if ($objInscripcio == null)
        throw new Exception("Error en el pagament");

    // Només actualitzem el número d'inscripció si anteriorment no existia.
    if ($objInscripcio->getCodiPagamentRealitzat () == null) {

        // Actualització del número d'inscripció ha de fer-se en exclusió
mútua o mutex.
        $mutex = new Mutex($db);
        $mutex->init ( "obtenirNInscripcio", 5000 );
        $mutex->lock ();

        $codiPagament =
            $modelInscripcio->get_next_codiPagament (
                $objInscripcio->getExcursionsId () );
        if ($codiPagament == null)
            $codiPagament = 0;
        $codiPagament = $codiPagament + 1;
        $objInscripcio->setCodiPagamentRealitzat ( $codiPagament );
        $modelInscripcio->updateAttached ( $objInscripcio );
        $mutex->release ();
    }
    else
        throw new Exception("El pagament ja s'ha efectuat");
}
```

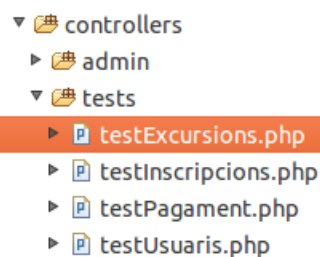
Malauradament PHP no disposa de la funció *Mutex* i s'ha implementat a través de la BD Mysql que si que permet el bloqueig. Aquest codi s'ha encapsulat en una classe *Mutex* que simplement rep com a paràmetre la BD.

El constructor permet definir el nom del bloqueig i el *timeout* del *Mutex*. Aquesta implementació no és òptima, però donat que no hi ha solució nativa del llenguatge, és una aproximació que duu a terme el seu propòsit. El *Mutex* s'utilitza en altres parts de la web com ara el mecanisme per generar un nou codi de comanda.

Jocs de proves

Els jocs de proves del sistema estan dissenyats per tal de provar alguns models del sistema i assegurar que algunes operacions com el pagament es realitzin de forma que no es produeixin repetits.

Els tests es troben en el directori **controllers/tests**, i son controladors que executen directament el test i mostren els resultats.



Tots els tests compten amb una funció índex, que és la que per defecte executa el controlador. La funció és responsable d'executar el conjunt de jocs de proves i mostrar els resultats de l'execució.

```
public function index()
{
    $this->setUp();
    $this->unit->run($this->testCrearUsuari(),
        true, 'testCrearUsuari');
    $this->unit->run($this->testEliminarUsuari(),
        true, 'testEliminarUsuari');
    echo $this->unit->report();
    $this->tearDown();
}
```

En l'exemple anterior es pot veure l'inicialització (*setUp*), comuna per tots els tests i la finalització (*tearDown*). A part es pot veure la crida a l'execució de dos tests, *crearUsuari* i *modificarUsuari*.

A continuació es pot veure el resultat de l'execució dels tests anterior, utilitzant la classe de tests que integra CodeIgniter:

← localhost/cewebinscripcions/tests/testUsuaris	
Nom del test	testCrearUsuari
Típus de dades del test	Boolean
Típus de dades esperat	Boolean
Resultat	Correcte
Nom del fitxer	/var/www/cewebinscripcions/application/controllers/tests/testUsuaris.php
Número de línia	34
Notes	
Nom del test	testEliminarUsuari
Típus de dades del test	Boolean
Típus de dades esperat	Boolean
Resultat	Correcte
Nom del fitxer	/var/www/cewebinscripcions/application/controllers/tests/testUsuaris.php
Número de línia	35
Notes	

El joc de proves testPagament

El joc de proves testPagament és un cas especial, ja que es vol comprovar que la generació de varies comandes una darrere l'altre és realitza sense duplicats, també es vol comprovar que el pagament es realitza de forma correcte sense tenir números d'inscripció a les excursions repetits.

Per simular aquest pagament s'executen a la vegada, o dins el mateix bucle el pagament a totes les inscripcions que marca el vector \$this->codisComandes, ja creades anteriorment. Per tant les comandes es paguen amb mili-segons de diferència.

Aquí podem veure la funció que realitza el pagament de totes les excursions en sèrie:

```
public function testPagarVariesComandes()
{
    foreach ($this->codisComandes as $codiComanda)
    {
        $this->unit->run(
            $this->helper_comprovaPagament(
                TestPagament::CODI_RETORN_BANCARI, $codiComanda,
                TestPagament::QUANTITAT_PAGADA),
            true,
            "comprovaPagamentComanda_" . $codiComanda);
    }
}
```

Entorn de desenvolupament

Les eines que s'han utilitzat per la planificació, l'anàlisi i la implementació del projecte, han estat basades en software lliure en la seva gran majoria. Aquest no era un objectiu del projecte, però si que es tractava d'un objectiu personal.

A continuació es realitza un inventari de les tecnologies implicades en el desenvolupament del projecte:

<p>Entorn de treball: Ubuntu Linux 12.04.</p>	
<p>Planificació del projecte: Project Libre, és una eina que ens permet planificar les tasques del projecte i generar un diagrama de Gantt a través d'aquesta planificació.</p>	
<p>Generació de diagrames i models UML: Dia, un programa per dibuixar diagrames estructurats.</p>	
<p>Servidor Web: Apache 2.2 configurat per PHP 5.3</p>	
<p>Servidor de base de dades: Mysql 5.5</p>	
<p>Client gràfic per Mysql: PHPMyAdmin 4.3</p>	
<p>Entorn de desenvolupament: Zend Eclipse for PHPDevelopers, entorn de desenvolupament basat en Eclipse, i personalitzat per el treball amb PHP. Suporta funcions molt interessants com ara l'autocompletat de codi, el ressaltat de sintaxi i la possibilitat de veure documentació sobre les classes al mateix temps que es programa.</p>	
<p>Gestió i control del repositori: GIT, i el portal de repositoris Bitbucket.</p>	
<p>Framework de desenvolupament MVC: CodeIgniter 2.1.3.</p>	
<p>Framework per la persistència de dades: Doctrine versió 2.2.2</p>	
<p>Framework per la generació de codis de barres en temps d'execució: PEAR 1.9.4</p>	

Conclusions

El projecte ha aconseguit els seus objectius inicials de forma satisfactòria. De fet el projecte ja es troba en producció i s'han realitzat les primeres proves de pagament.

Cal destacar sobretot el nombroses tecnologies que s'han hagut d'estudiar per dur a terme el projecte. Començant per la més important CodeIgniter i el patró de disseny (MVC) Model Vista Controlador, que ha permès estructurar el codi de forma més llegible i separant la responsabilitat de cadascun dels elements que el componen. També s'ha treballat amb Doctrine, un gestor ORM que proporciona la part de persistència de l'aplicació i PEAR per la generació en temps d'execució de codis de barres.

Pel que fa a la presentació s'han estudiat tècniques de disseny web *responsive* per fer compatible la part del Frontend amb la visualització a través de dispositius mòbils. Finalment per fer-ho tot més atractiu s'ha treballat amb un *framework* de presentació com és Bootstrap que combina CSS i Javascript.

A més s'ha hagut d'estudiar la comunicació amb la part bancària (SERMEPA), per realitzar el pagament. Aquesta part ha estat especialment complicada, ja que el banc proporciona la informació de pagament de forma asíncrona i xifrada per evitar-ne la seva manipulació.

Pel que fa a la metodologia seleccionada OpenRUP, ha resultat molt interessant, ja que ha permès adaptar-se als canvis demandats pel client en les diferents fases i entregar en cada fase un producte funcional que ha pogut ser avaluat. D'aquesta manera s'ha obtingut un Feedback molt valuós que ha permès prendre decisions de disseny i corregir errors durant el projecte, minimitzant el risc que el producte final no s'adaptés a les necessitats del client, fet que podria haver passat si s'hagués utilitzat una metodologia tradicional.

Com a punt crític, és important destacar l'elevada casuística que es dona en les diverses excursions que organitza el CEO (Centre Excursionista Olot), que fan que l'aplicació sempre pugui incorporar-ne de noves i sigui difícil a vegades agrupar diferents tipus d'excursions sota un mateix nom. L'aplicació s'ha adaptat a aquests canvis creant diferents tipus d'excursió i permetent que aquests tipus siguin ampliables de forma fàcil de cares a possibles noves activitats.

Finalment i com a opinió personal. Fer aquest projecte m'ha servit per entendre amb més detall el món del desenvolupament web ja que he hagut de familiaritzar-me amb un bon munt de tecnologies que treballen conjuntament per realitzar una aplicació web i que s'utilitzen en el món laboral en l'actualitat. D'altra banda estic satisfet d'haver portat el projecte endavant utilitzant només solucions basades en software lliure, ja que inicialment ho veia un xic complicat.

Línies de futur

El projecte ara cobreix les necessitats bàsiques del CEO, és a dir les excursions que organitza de forma més freqüent: Embardissada, PocaTraça, Volta Volcans, 24H, entre d'altres. Tot i així, s'ha detectat en les últimes reunions mantingudes amb el client, la necessitat d'incorporar noves excursions, com les que organitza la secció de mainada o bé la secció d'esquí. Amb necessitats molt més específiques.

Val a dir que el sistema ha estat plantejat per permetre aquesta ampliació de forma fàcil, ja que simplement cal afegir nous tipus d'excursions, entre les que ja es troben disponibles actualment, i re-implementar algunes parts com ara la butlleta o la vista de demanar dades de l'excursió.

Un altra possible ampliació que s'està estudiant, és la possibilitat d'unificar les dades del programa de gestió de socis del centre, amb la web d'inscripcions desenvolupada en aquest projecte. D'aquesta forma es facilitaria enormement la gestió integrada de les dades.

Per últim cal dir que inicialment s'havia plantejat com a línia de futur la possibilitat de portar el control de les persones inscrites, de forma que es pogués saber quins controls han passat i si s'ha perdut algun excursionista a temps real. Tot i així sembla que aquest seria un sistema autònom i la única connexió que tindria amb la web d'inscripcions seria per extreure dades dels participants i usuaris inscrits.

Glossari de termes

ActiveRecord	Patró d'arquitectura del software que s'encarrega de la persistència de dades.
Apache	Servidor web.
Bootstrap	<i>Framework</i> per el desenvolupament d'aplicacions web adaptades als dispositius mòbils.
CodeIgniter	<i>Framework</i> desenvolupat sobre PHP que aporta llibreries pel desenvolupament d'una aplicació.
CSS	Cascading StyleSheets, fulls d'estil en cascada, llenguatge declaratiu per definir l'estil de presentació d'una pàgina web.
DataMapper	Patró d'arquitectura del sotware que inclou funcions com crear, llegir i modificar, que operen sobre els objectes entitats de domini en una base de dades.
Doctrine ORM	Doctrine Object Role Modeling, Motor de persistència basat en PHP, que s'encarrega de negociar amb la BD, implementa el patró DataMapper.
HTML	HyperText Markup Language , llenguatge de descripció dels continguts d'una pàgina web.
HTTP	HyperText Transfer Protocol, protocol usat per la negociació amb un servidor web.
HTTPS	Hypertext Transfer Protocol Secure, protocol que defineix la transmissió d'informació HTTP segura.
LAMP	Conjunt de tecnologies pel desenvolupament d'aplicacions web. Son les sigles de Linux, Apache, Mysql i PHP.
MVC	Model-View-Controller, patró de desenvolupament de software que proposa la separació de la presentació, la lògica de negoci i l'accés a dades.
Mysql	Servidor de base de dades.
NONCE	Number used only Once, en una comunicació segura el nonce és un número que només s'utilitza una sola vegada.
PEAR	PHP Extension and Application Repository, consisteix en un conjunt de llibreries que es gestionen en un sol repositori.
PHP	PHP Hypertext Pre-processor, llenguatge de programació script interpretat i no tipat, molt usat en el desenvolupament d'entorns web.
PHPDoc	Estàndar oficial per a comentar codi en PHP.
SGBD	Sistema Gestor de Base de Dades especialitzat en la gestió de la informació en una base de dades, normalment relacional.
SHA	Secure Hash Algorithm, sistema de funcions hash criptogràfiques.
SQL	Structured Query Language, llenguatge de consulta utilitzat en la manipulació de la informació en un SGBD.
TLS	Transport Layer Security, implementa protocols criptogràfics per permetre connexions segures a través d'Internet.
URL	Uniform Resource Locator, serveix per a referenciar continguts a través d'Internet.

Bibliografia i referències

Per l'elaboració d'aquest document s'han consultat varies referències que s'han agrupat segons la seva finalitat en el desenvolupament de l'aplicació:

Programació de la part del servidor

Codeigniter for Rapid PHP Application Development
Editorial PACKT
Autor: David Upton

Web amb la documentació de Codeigniter

<http://ellislab.com/codeigniter>

Web de Doctrine per treballar la persistència de dades

<http://docs.doctrine-project.org/projects/doctrine-orm/en/latest/>

Diferències entre el patró active record i datamapper

<http://russellscottwalker.blogspot.com.es/2013/10/active-record-vs-data-mapper.html>

Autenticació en Digest segons el Dr. Heng-Yow Chen

http://solomon.ipv6.club.tw/Course/Web_Technology/Slides/13.%20Digest%20Authentication.ppt

Presentació part client

Introducció al responsive design o com crear aplicacions per mòbils

<http://blog.ikhuerta.com/transforma-tu-web-en-responsive-design>

Més informació sobre el CSS i mòbils

<http://www.webdesignerdepot.com/2013/03/how-to-make-your-contact-forms-mobile-friendly/>

Pàgina web de les llibreries de presentació Bootstrap

<http://getbootstrap.com/>

Disseny de software i metodologia de desenvolupament

Apunts Dia per disseny UML

<http://www.eg.bucknell.edu/~cs475/F04-S05/UMLTutorial.pdf>

Disseny de casos d'ús

http://www.karonaconsulting.com/downloads/UseCases_IncludesAndExtends.pdf

Metodologia de desenvolupament de software OpenUP:

<http://epf.eclipse.org/wikis/openup/>

Apunts assignatura MGPI

Editorial: UOC

**Materials de l'assignatura de Enginyeria del Programari de Components i Sistemes
Distribuïts**

Editorial: UOC

Materials diversos de l'assignatura de projecte (UOC)

Annex 1 – Manual de l'administrador de sistemes

Manual de desplegament de l'aplicació per primer cop

Per realitzar el desplegament de l'aplicació simplement cal seguir els següents passos:

- Cal descarregar la última versió del codi del repositori GIT, o bé obtenir un zip amb la última revisió de codi.

Per descarregar la última versió de codi a través de BitBucket, cal executar la següent comanda:

```
git clone https://xeviter@bitbucket.org/xeviterr/cewebinscripciones.git
```

- Crear la BD a partir del fitxer existent dins el codi: **dades_bd/estructurabd.sql**
- Col·locar el codi de l'aplicació dins un directori a elecció de l'administrador del servidor web.
- Configurar el fitxer **application/config/database.php** els paràmetres de la BD, segons el tipus de servidor del qual es disposi.

```
$db['default']['hostname'] = 'localhost';  
$db['default']['username'] = 'root';  
$db['default']['password'] = '...';  
$db['default']['database'] = 'cewebinscripciones';  
$db['default']['dbdriver'] = 'mysql';
```

- Donar permisos d'escriptura i lectura al directori **tmp**
- Finalment s'haurà de configurar el següent fitxer **application/config/config.php** aquí es configuren la resta de valors de l'aplicació que no son la BD.

```
$config['mail_from'] = 'ceo@ceolot.org';  
  
$config['tmp_dir'] = realpath(APPPATH . '/../tmp/');  
  
//Directorio temporal per guardar codis de barres i fitxers descarregables.  
$config['tmp_dir_url'] = '/tmp';  
  
//Valors de pagament pel desenvolupament.  
$config['url_pagament'] = 'https://sis-  
t.sermepa.es:25443/sis/realizarPago';  
$config['module_payment_caixa_sha_id_com']='xxxxxxx';  
$config['module_payment_caixa_sha_id_primaria']='xxxxxxx';  
  
//Valors de configuració del password d'administració.  
$config['admin_user'] = 'root';  
$config['admin_password'] = 'xxx';
```

Execució dels jocs de proves

Per comprovar que el sistema funciona correctament es pot executar els tests d'integració, que comproven el funcionament bàsic d'alguns mòduls de l'aplicació.

ALERTA: Per executar els tests, és molt recomanable fer-ho sobre una BD de proves i mai directament sobre el servidor de producció si aquest conté dades del client per prevenir possibles pèrdues de dades. Les url's per accedir-hi son:

<http://servidor/tests/testExcursions.php>
<http://servidor/tests/testInscripcions.php>
<http://servidor/tests/testPagament.php>

<http://servidor/tests/testUsuaris.php>

Si aquests testos passen correctament voldrà dir que l'aplicació està apunt per treballar.

Manual de desplegament d'una nova versió

Per desplegar una nova versió del programa en el servidor de producció, abans cal provar-la i executar tots els tests en l'entorn de proves per assegurar que funcioni correctament. La publicació en el servidor és un procés pensat per ser ràpid i no s'hauria de prolongar durant massa temps.

Per publicar de forma correcta, cal realitzar els següents passos:

- **Parar la web** per evitar possibles accessos no desitjats. Si no es pot parar la web, sempre es pot renombrar el fitxer index.php amb un altre nom. El fitxer index.php, és el punt d'entrada principal a l'aplicació i és l'encarregat de cridar al diferents controladors, per tant si es desactiva es desactiva tota possibilitat d'accedir tant al Frontend com a Backend.
- **Realitzar una còpia de seguretat de la BD**, per evitar possibles pèrdues de dades dels usuaris. També hi ha la possibilitat de realitzar una còpia de seguretat dels fitxers del servidor web, tot i que aquest últim punt no és del tot necessari, ja que les diferents versions publicades es troben dins el repositori GIT i sempre son recuperables.
- **Executar els canvis efectuats a la BD respecte versions anteriors**. Els canvis a la BD es troben en el fitxer **dades_bd.sql**, per executar-los simplement cal accedir a Phpmyadmin del servidor de producció i executar tot el que hi hagi dins el fitxer. Segons la versió que es descarregui del repositori aquests canvis poden variar. Per tant el repositori fa la feina de controlar les modificacions a la BD.
- **Publicar els fitxers en el servidor web**. Per publicar de forma automàtica i transmetre només els canvis realitzats, es pot utilitzar l'script **publicar.bash**

```
administrador@Inspiron1525:/var/www/cewebinscripciones$ bash publicar.bash  
Password: █
```

La tasca que duu a terme l'script és accedir a l'FTP del servidor web de producció i sincronitzar els canvis dels fitxers en local amb els fitxers remots. Per fer-ho utilitza un programa LFTP, que actua com a client ftp, i al mateix temps permet fer un MIRROR o mirall entre el contingut del servidor local i el contingut del servidor remot. Aquesta opció permet que només es publiquin els fitxers que han estat modificats.

Abans d'utilitzar l'script cal personalitzar-lo. Concretament cal modificar l'adreça del servidor a publicar, el nom d'usuari, el directori d'origen de publicació i el de destí.

A continuació es pot veure el contingut del fitxer, que es necessari personalitzar:

```
lftp -u xevit ceolot.cat -e "mirror --exclude=system --exclude=user_guide  
--exclude=tmp --exclude=docs --exclude=.git --exclude=.settings  
--exclude=application/config/ --exclude=application/libraries/Doctrine  
--exclude=application/libraries/Pear --reverse  
/var/www/cewebinscripciones /embardissada/cewebinscripciones -v -c"
```

També hi ha la possibilitat de configurar els directoris que no seran publicats. Molts d'aquests directoris no es troben ignorats, degut a que es tracta de llibreries que normalment no varien massa en el temps i no és necessari publicar-les cada vegada.

Annex 2 - Manual de l'administrador de la web

El sistema consta de dues parts la part oberta als excursionistes (pública o frontend) i la part privada, on només hi pot entrar el personal de centre excursionista (backend).

Part pública

Per accedir a la part pública es pot fer a través de la URL:

<http://www.ceolot.cat/embaridissada/cewebinscripcions/>

La part pública de la web, és molt intuïtiva i no s'ha considerat la realització d'un manual de funcionament, ja que un dels objectius és que sigui simple en el seu ús i oberta a tothom.

Part privada

Per accedir a la zona d'administració es pot fer a través de la URL:

<http://www.ceolot.cat/embaridissada/cewebinscripcions/admin/admin/>

Des d'aquí es poden crear excursions, veure els usuaris i les inscripcions a les diferents excursions creades. La paraula de pas provisional per accedir a la part privada és:

- Nom usuari: root
- Paraula de pas: President

Aquesta paraula de pas es pot modificar en el fitxer de configuració, tal i com descriu el manual de l'administrador de sistemes.

Gestió d'excursions

Acció	Codi	Nom	Data Event	Tipus
	1032	24 hores a muntanya	2014-01-10	ExcursioEmbaridissada
	1031	Embaridissada 2014	2014-03-01	ExcursioEmbaridissada
	1051	Embaridissada de proves	2013-12-10	ExcursioEmbaridissada
	1053	Embaridissada de proves	2013-12-10	ExcursioEmbaridissada
	1054	Embaridissada de proves	2013-12-10	ExcursioEmbaridissada

[Crear nova excursió](#)
[Crear nova excursió bàsica](#)
[Crear nova excursió embaridissada](#)

En aquesta pantalla es poden gestionar les diferents excursions que organitza el CEO. Actualment hi han tres tipus d'excursions:

- **Excursió:** És un tipus excursió genèric, que només té una descripció, permet que un usuari ja registrat es pugui inscriure. No es realitza pagament a través del banc, ni admet cap mena d'opció. Aquest tipus d'excursió resulta útil per inscriure els usuaris habituals a les excursions que organitza el centre de forma periòdica o simplement per confirmar l'assistència a una matinal o excursió no oberta al públic.
- **Excursió bàsica:** És el tipus d'excursió més freqüent en el centre. Es pot parametritzar a través de diferents opcions:
 - Opció assegurança: Permet indicar que volem demanar a l'usuari si vol assegurança i ens permet detallar-ne el preu.
 - Opció no pagament: Permet indicar si l'excursió és de pagament o simplement es vol inscriure als usuaris sense passar per la passarel·la bancària.
 - Opció només federats: Indica que l'usuari que s'inscriu ha d'especificar un número de federat de forma obligatòria. Aquesta opció té sentit sobretot en curses.
 - Opció tarifes anticipades: Indica que es poden aplicar les tarifes anticipades, sempre i quan la data actual sigui inferior a la data indicada per aquest tipus de tarifa.
- **Excursió embardissada:** És una excursió idèntica al tipus bàsic, però admet etapes. Un usuari podrà inscriure's en una o varies etapes. Existeix una opció addicional respecte a les opcions de l'excursió bàsica:
 - Opció etapes obligatòries: Permet definir si totes les etapes, s'han de realitzar de forma obligatòria, o bé l'usuari pot seleccionar en quin tipus d'etapa participará.

Creació de l'excursió

Per crear una excursió simplement cal accedir als enllaços que hi han a la part inferior i seleccionar el tipus d'excursió desitjat. Segons el tipus d'excursió apareixeran dades diferents a omplir.

Modificar excursió

La modificació de l'excursió pot resultar un problema si ja hi han usuaris inscrits, donat que poden variar les condicions en que un usuari s'apunta a l'excursió. El sistema podria controlar aquest punt, no deixant editar, o només deixant editar alguns camps, si hi ha un sol usuari apuntat, però a petició del CEO s'ha demanat que sempre es pugui modificar sota la seva responsabilitat.

Eliminar excursió

Permet l'eliminació d'una excursió sempre que no hi hagin inscrits, si aquest és el cas, l'excursió no es pot eliminar i el sistema ho notificarà amb un error.

Gestió d'usuaris

Administració CEO	Excursions	Inscrits	Usuaris	Entitats	Pagaments	Pagament manual
-------------------	------------	----------	---------	----------	-----------	-----------------

Gestió d'usuaris

Accions	Email	Nom	Cognoms
<input type="button" value="Eliminar"/> <input type="button" value="Modificar"/>	albert@gmail.com	Albert	Terr
<input type="button" value="Eliminar"/> <input type="button" value="Modificar"/>	albertcorreu@gmail.com	Albert	Terra
<input type="button" value="Eliminar"/> <input type="button" value="Modificar"/>	paquito@gmail.com	Paquito	Perez
<input type="button" value="Eliminar"/> <input type="button" value="Modificar"/>	pere@gmail.com	Pereo	Prova

[Crear nou usuari](#)

Els usuaris son creats pels excursionistes en el moment d'inscriure's a la seva primera excursió i les dades es poden modificar a posteriori per ells mateixos. De totes maneres des de l'administració es possible realitzar el manteniment complet d'un usuari.

Es important destacar que no es possible eliminar un usuari amb inscripcions realitzades. Si s'intenta la operació el sistema notificarà de l'error.

Administració CEO	Excursions	Inscrits	Usuaris	Entitats	Pagaments	Pagament manual
-------------------	------------	----------	---------	----------	-----------	-----------------

Gestió d'usuaris

* Correu electrònic:	<input type="text" value="albert@gmail.com"/>
* Nom:	<input type="text" value="Albert"/>
* Cognoms:	<input type="text" value="Terr"/>
* Sexe:	<input type="text" value="Home"/>
* Data de naixament (dd/mm/aaaa):	<input type="text" value="08/08/2000"/>

Gestió d'inscripcions

La funció de la gestió d'inscripcions, és veure els diferents usuaris que hi ha inscrits a una excursió determinada i permetre extreure llistats per tal que el personal del CEO pugui disposar de les dades.

Inicialment apareixerà un llistat amb totes les excursions organitzades pel centre, així per veure els inscrits en una excursió concreta només caldrà fer click al damunt.

Administració CEO	Excursions	Inscrits	Usuaris	Entitats	Pagaments	Pagament manual
-------------------	------------	----------	---------	----------	-----------	-----------------

Gestió d'inscrits

1	2	>
---	---	---

Nom	Data Event	Tipus
24 hores a muntanya	2014-01-10	ExcursioEmbardissada
Embardissada 2014	2014-03-01	ExcursioEmbardissada
Excursió apuntat i prou	2014-01-07	Excursio
Passejada del diumenge al matí	2014-01-31	ExcursioBasica

A continuació apareixeran els inscrits en una determinada excursió. Aquí es pot veure el número de comanda assignat pel pagament, la data en que s'ha fet la inscripció i el codi de pagament realitzat. El codi de pagament realitzat només apareixerà si el pagament s'ha realitzat amb èxit.

Administració CEO	Excursions	Inscrits	Usuaris	Entitats	Pagaments	Pagament manual
-------------------	------------	----------	---------	----------	-----------	-----------------

Inscrits excursió Passejada del diumenge al matí

Usuari	Num. Comanda	Data inscripció	Codi pagament realitzat
albert@gmail.com	1389085874	2014-01-07 10:11:00	2
paquito@gmail.com	1388772531	2014-01-03 18:07:41	1

[Exportar inscrits](#)

A la part inferior del llistat apareixerà sempre el botó exportar inscrits, que permetrà obtenir un llistat en format CSV de tots els usuaris inscrits en una excursió. El llistat inclou, algunes de les dades d'inscripció, i d'usuari relacionades, tot i així els camps a exportar varien segons el tipus d'excursió.

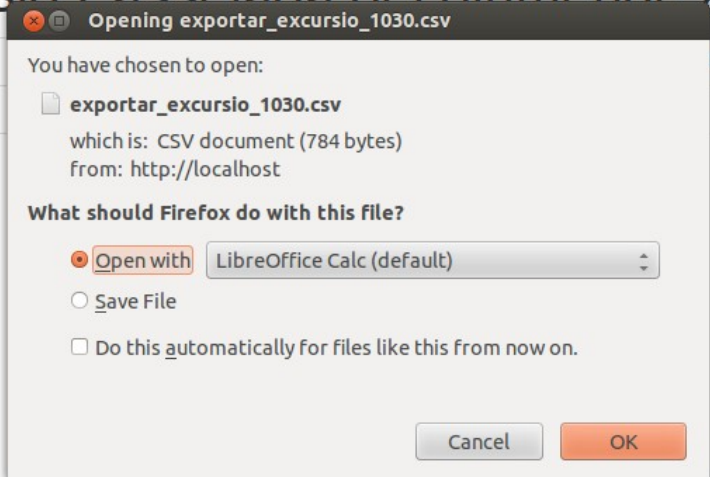
Administració CEO Excursions Inscrits Usuaris Entitats Pagaments Pagament manual

Inscrits excursió Passejada del diumenge al matí

[Usuari](#)
albert@gmail.com
paquito@gmail.com

[Exportar inscrits](#)

[i pagament realitzat](#)



Segons el tipus d'excursió poden aparèixer opcions d'exportació addicionals. Com en el cas d'una excursió de tipus Embardissada, on es poden obtenir llistats parcials amb les persones que s'han apuntat a una etapa en concret. A continuació es pot veure un exemple.

Administració CEO Excursions Inscrits Usuaris Entitats Pagaments Pagament manual

Inscrits excursió 24 hores a muntanya

Usuari	Num. Comanda	Data inscripció	Codi pagament realitzat
albertcorreu@gmail.com	1388606780	2014-01-01 21:06:20	1
paquito@gmail.com	1388772713	2014-01-03 17:40:43	3
pere@gmail.com	1388608246	2014-01-01 21:30:46	2

[Exportar inscrits](#)
[Exportar inscrits etapa Etapa1](#)
[Exportar inscrits etapa Etapa2](#)
[Exportar inscrits etapa Etapa3](#)
[Exportar inscrits etapa Etapa4](#)

Gestió d'entitats

Les entitats apareixen en el moment de crear un usuari i representen els diferents clubs excursionistes de Catalunya o les diferents entitats que habitualment s'inscriuen a les excursions que organitza el CEO. De les entitats es pot fer un manteniment complert. Cal destacar però que a l'hora d'eliminar una entitat només serà possible si no hi han usuaris que disposin de la mateixa. El camp "Per defecte", indica quina és l'entitat que apareixerà per defecte en la creació d'un usuari. El valor

per defecte hauria de ser sempre, "Altre entitat" i no hi hauria d'haver més d'un valor per defecte seleccionat.

Administració CEO	Excursions	Inscrits	Usuaris	Entitats	Pagaments	Pagament manual
-------------------	------------	----------	---------	----------	-----------	-----------------

Gestió d'entitats

Accions	Entitat	Per defecte
<input type="button" value="Eliminar"/> <input type="button" value="Modificar"/>	[Altre entitat]	1
<input type="button" value="Eliminar"/> <input type="button" value="Modificar"/>	CEO	0
<input type="button" value="Eliminar"/> <input type="button" value="Modificar"/>	Girona	0
<input type="button" value="Eliminar"/> <input type="button" value="Modificar"/>	La Selva	0
<input type="button" value="Eliminar"/> <input type="button" value="Modificar"/>	UE Banyoles	0

[Crear nova entitat](#)

Pagaments

En aquest apartat es mostra un llistat paginat amb els pagaments que s'han realitzat a través de la passarel·la de pagament de la Caixa. La utilitat d'aquest apartat és bàsicament de consulta, i l'ús principal és a partir d'una comanda veure quin ha estat el resultat del seu pagament, la data i la quantitat pagada.

El resultat del pagament indicarà sempre el codi d'error retornat i contrastant amb el manual de la Caixa es pot deduir ràpidament quin problema s'ha produït en la transacció. Hi ha un bon munt de codis d'error que poden indicar casuístiques com ara: targeta fora de servei, no hi ha diners en compte, targeta no segura, entre d'altres.

Per facilitar la cerca entre tants registres, s'ha col·locat un filtre en aquesta pantalla que permet filtrar les comandes que continguin un determinat codi, o bé introduir un codi de comanda i obtenir-ne les dades.

Administració CEO	Excursions	Inscrits	Usuaris	Entitats	Pagaments	Pagament manual
-------------------	------------	----------	---------	----------	-----------	-----------------

Veure pagaments

Filtrar per número de comanda:

> Últim

Comanda	Data i hora pagament	Codi pagament	Quantitat pagada
1388437290	2013-12-30 22:01:34	8	10
1388437291	2013-12-30 22:01:35	8	10

Inscripció o pagament manual

Per inscriure un usuari de forma manual, és a dir, sense passar pel banc. Només cal obtenir el seu número de comanda i introduir-lo en aquest apartat. El numero de comanda es genera per qualsevol inscripció que es realitza a través de la web, per tant es possible realitzar aquesta operació a partir d'una inscripció on el pagament no s'hagi pogut efectuar o s'hagi efectuat de forma errònia.

Si el codi de comanda és correcte i no s'ha realitzat el pagament de la inscripció el sistema apuntarà l'usuari a l'excursió sol·licitada en el número de comanda i li assignarà un codi d'inscripció o dorsal. El sistema notificarà via mail a l'usuari de la nova inscripció realitzada.

Administració CEO	Excursions	Inscrits	Usuaris	Entitats	Pagaments	Pagament manual
-------------------	------------	----------	---------	----------	-----------	-----------------

Pagament manual d'una inscripció

Entreu el número de comanda per realitzar-ne el pagament

Pagament realitzat correctament