



Aplicatiu web per la confecció de Plans de Seguretat i Salut en obres de construcció.

Mateu Oliver Monserrat
Grau d'Enginyeria Informàtica

Oriol Martí Girona

11/06/2016

© Mateu Oliver Monserrat

Reservats tots els drets. Està prohibit la reproducció total o parcial d'aquesta obra per qualsevol mitjà o procediment, compresos la impressió, la reprografia, el microfilm, el tractament informàtic o qualsevol altre sistema, així com la distribució d'exemplars mitjançant lloguer i préstec, sense l'autorització escrita de l'autor o dels límits que autoritzi la Llei de Propietat Intel·lectual.

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Aplicatiu web per la confecció de Plans de Seguretat i Salut en obres de construcció.</i>
Nom de l'autor:	<i>Mateu Oliver Monserrat</i>
Nom del consultor:	<i>Oriol Martí Girona</i>
Data de lliurament (mm/aaaa):	<i>06/2014</i>
Àrea del Treball Final:	<i>Enginyeria del Programari</i>
Titulació:	<i>Grau d'Enginyeria Informàtica</i>
Resum del Treball (màxim 250 paraules):	
<p>Aplicatiu web per redacció de plans de seguretat i salut a les obres de construcció.</p> <p>L'aplicatiu consta d'una gestió d'usuaris, una gestió de cobrament i el procés de redacció pròpiament dit. Defineix usuaris freemium i usuaris de pagament, els primers podran redactar dos plans a l'any gratuïts i els segons no tindran aquesta limitació i comptaran amb eines per l'edició del contingut estàndard</p>	
Abstract (in English, 250 words or less):	
<p>Web application for redact Health and Safety plans in works.</p> <p>The application includes a manager of users, a manager for payment and the process for doing plans. It defines two kind of users: freemium and payment users. The firsts can do two plans in a year and the seconds doesn't have this limitation and besides they have editing tools.</p>	
Paraules clau (entre 4 i 8):	
Web, python, Django, Amazon EC2,	

Índex

Grau d'Enginyeria Informàtica.....	1
Grau d'Enginyeria Informàtica.....	i
1. Introducció.....	1
1.1 Context i justificació del Treball.....	1
1.2 Objectius del Treball.....	2
1.3 Enfocament i mètode seguit.....	3
1.4 Planificació del Treball.....	3
1.5 Breu sumari de productes obtinguts.....	4
1.6 Breu descripció dels altres capítols de la memòria.....	4
2. Recollida de requisits.....	5
2.1 Antecedents legals e introducció al pla de seguretat i salut.....	5
2.2 Entrevistes per la recollida de requisits.....	8
2.3 Diagrames identificatius dels casos d'ús.....	11
2.3.1 Login.....	11
2.3.2 Pagament.....	11
2.3.3 Selecció de partides.....	12
2.3.4 Cas d'ús entrades de dades.....	13
3. Anàlisi.....	15
3.1 Adaptació de la solució al «framework».....	15
3.2 Diagrames de seqüència.....	17
3.2.1 Login.....	17
3.2.2 Pagament.....	17
3.2.3 Selecció de partides.....	18
3.2.4 Cas d'ús entrades de dades.....	18
3.3 El model de dades.....	19
4. Disseny Tècnic.....	21
4.1 Introducció: bases de l'arquitectura de desenvolupament.....	21
4.2 Python.....	21
4.3 Framework Django.....	21
4.3.1 El gestor de BBDD de Django.....	25
4.4. Diagrama entitat relació de les dades.....	27
4.6. Diagrama final de les dades.....	28
4.7 Esbos de les pantalles.....	28
4.8. Pantalles proposades.....	31
5. Implementació.....	37
5.1 Eclipse amb plugin pydev.....	37
5.2 Pycharm de jetbrains.....	37
5.3 Mysql.....	37
5.4 Bitbucket.org repositori Git remot.....	38
5.5 EC2 d'Amazon.....	39
5.5.1 ElasticBeanstalk.....	39
5.6 Manual d'ús i Joc de proves.....	40
5.6.1 Dinàmica d'ús de l'aplicatiu.....	40
5.6.2 Introducció dades necessàries per fer un pla.....	40
5.6.3 Generació del pla i pagament.....	48
6. Estat actual del projecte.....	50
7. Pressupost econòmic.....	51
8. Glossari.....	52

9. Bibliografia.....	53
----------------------	----

Llista de Figures

Illustration 1: Planificació del projecte.....	3
Illustration 2: Marc Normatiu Preventiu.....	5
Illustration 3: Procediment per començar una obra de construcció.....	6
Illustration 4: Diagrama d'estats del cas d'ús de login.....	11
Illustration 5: Diagrama d'estats per la selecció de partides.....	12
Illustration 6: Diagrama cas d'ús de la selecció de plantilles.....	12
Illustration 7: Diagrama d'estats genèric per l'entrada de dades a l'aplicatiu.....	13
Illustration 8: Diagrama de cas d'ús per l'entrada de dades a l'aplicatiu.....	14
Illustration 9: Diagrama de seqüència de login.....	17
Illustration 10: Diagrama de seqüència de pagament.....	18
Illustration 11: Diagrama de seqüència de selecció de partides.....	18
Illustration 12: Diagrama de seqüència d'entrada de dades.....	19
Illustration 13: Diagrama de fluxe del framework.....	24
Illustration 14: Diagrama entitat relacio.....	27
Illustration 15: Diagrama final de Taules.....	28
Illustration 16: Pantalla d'obra.....	29
Illustration 17: Pantalla d'obra retorna resultats cerca.....	29
Illustration 18: Proposta Pantalla Partides.....	30
Illustration 19: Pantalla definitiva tècnics.....	31

1. Introducció

1.1 Context i justificació del Treball

L'empresa cima20, és un servei de prevenció de riscos laborals amb la autorització IB-011 i àmbit d'actuació dins les Illes Balears.

Un servei de prevenció és tal com defineix el REAL DECRETO 39/1997, de 17 de enero, por el que se aprueba el Reglamento de los Servicios de Prevención. BOE no 27 31-01-1997.

És una de les modalitats que pot triar un empresari per implantar la prevenció de riscos laborals dins el seu negoci i mantenir-la.

Aquesta necessitat ve definida per la Llei de prevenció de riscos laborals 31/95 fruit de la transposició de la Directiva Marc89/391 de les Comunitats Europees.

El tipus de clients que tenen un servei de prevenció aliè, en endavant SPA pot ésser qualsevol des de una empresa de 1 treballador fins empreses de 250 treballadors empreses de risc o fins 500 empreses sense risc. A partir d'aquesta mida han de constituir un servei de prevenció propi.

Les empreses es veuen obligades a avaluar els seus riscos laborals i plantejar mesures preventives amb la finalitat última d'evitar accidents laborals i malalties professionals.

Llavors, si es segueixen tots els preceptes preventius s'ha d'aconseguir aquest fet.

En concret, per empreses que entren dins d'obres amb la qual cosa afegiríem tots els oficis com electricistes, llanterners, guixaires, pintors ,etc a més de constructores i picapedrers. Es veuen obligades a complir amb un requisit més, el Pla de Seguretat i Salut dins d'obres de construcció.

Aquest tipus d'empresa per mitja és una empresa petita amb pocs coneixements tant preventius i òbviament pocs coneixents informàtics. Per afegir més contratemps el pla de seguretat que han de confeccionar ells és una contesta a un document generat per un tècnic (arquitecte, enginyer industrial, arquitecte tècnic, enginyer tècnic) anomenat estudi de seguretat i salut, llavors la problemàtica s'incrementa.

Els clients no estan capacitats per generar aquesta documentació d'una manera correcta.

A cima20 on treballa, es planteja: l'increment dels seus clients a través del web, la monetització del seu coneixement del sector de prevenció de riscos laborals, així com fer un experiment per traslladar tot el seu coneixement amb el mateix tipus d'eina i d'aquesta manera assegurar la supervivència de l'empresa en temps convulsos.

La rellevància del treball, radica tant en la finalitat del projecte com en que també és una manera d'experimentar amb solucions al núvol, ja que l'empresa no vol tenir cap maquinari a les seves instal·lacions per donar el servei ofert.

El problema de moment es resol amb l'ús d'un programari comercial instal·lat en local destinat a fer amidaments d'obres, usat d'una manera per el que no ha estat pensat i s'utilitza amb una persona de l'empresa.

Per tant amb el nou programari al núvol es pretén alliberar a cima20 del manteniment de servidors i a la vegada alliberar a la persona que en la actualitat fa els plans.

És pretén que siguin els clients qui directament facin els seus propis plans.

D'aquesta manera s'aconsegueix un compromís del client més alt amb la prevenció al involucrar-se amb el desenvolupament del seu propi Pla de seguretat i salut; en endavant PSS.

A més de facilitar la problemàtica als clients; ja que per fer un pla han d'omplir un formulari tan complex i exhaustiu que amb el mateix esforç ja tindrien la feina feta.

1.2 Objectius del Treball

Els objectius que es pretenen assolir es poden dividir en dos grans grups els primers acadèmics i els segons empresarials:

Acadèmics:

- Demostrar els coneixements obtinguts durant el grau.
- Capacitat de gestió i planificació d'un projecte.
- Capacitat de concreció d'un problema en unes eines de feina que permetin l'estudi del mateix amb garanties en els productes obtinguts.
- Capacitat d'aprenentatge d'un llenguatge nou cas del Python.
- Capacitat d'aprenentatge d'un framework cas del Django.
- Capacitat d'aprenentatge de les eines cloud que ofereix EC2 d'Amazon

Empresarials:

- Aplicació que compleixi els requisits de l'empresa on treballa, per realitzar plans.
- Plenament operativa, capaç de permetre el pagament de la documentació generada.
- Incrementar els ingressos del servei de prevenció, així com augmentar la seva base de clients, ja que el servei té àmbit regional i d'aquesta manera quasi sense cost incrementaria la seva presència a nivell nacional.

- Permetre ,en un futur poder llançar més programari amb la mateixa filosofia; si el mercat respon.

1.3 Enfocament i mètode seguit

L'estratègia triada ha estat el desenvolupament conceptual d'un producte nou encara que s'emprarà el coneixement que ja s'utilitza a l'empresa a l'actualitat.

El producte ha d'ésser nou perquè el que es pretén :

- Treballar al núvol.
- Arribar a clients de tota la península en lloc de només els de l'illa de Mallorca, que és on ara està radicat el servei de prevenció.
- Fer que els clients s'autoabasteixin, és a dir que emprin tot sols l'eina.
- Cobrar a través d'un sistema àmpliament establert a Internet, en aquest cas "Paypal". Aquest sistema permet el pagament amb targeta (passarel·la de pagament) o directament amb un compte de Paypal (té una targeta de crèdit lligada).

Fa que el producte hagi d'ésser radicalment nou. Ara bé el contingut utilitzat no cal que sigui nou ja que es tracta de refinar uns continguts existents.

1.4 Planificació del Treball

El treball es planifica amb l'eina web <http://www.tomsplanner.com/> ,definint tres grans grups:

- Pressa de requisits 26d
- Especificació de requisits 33d
- Programació 30d

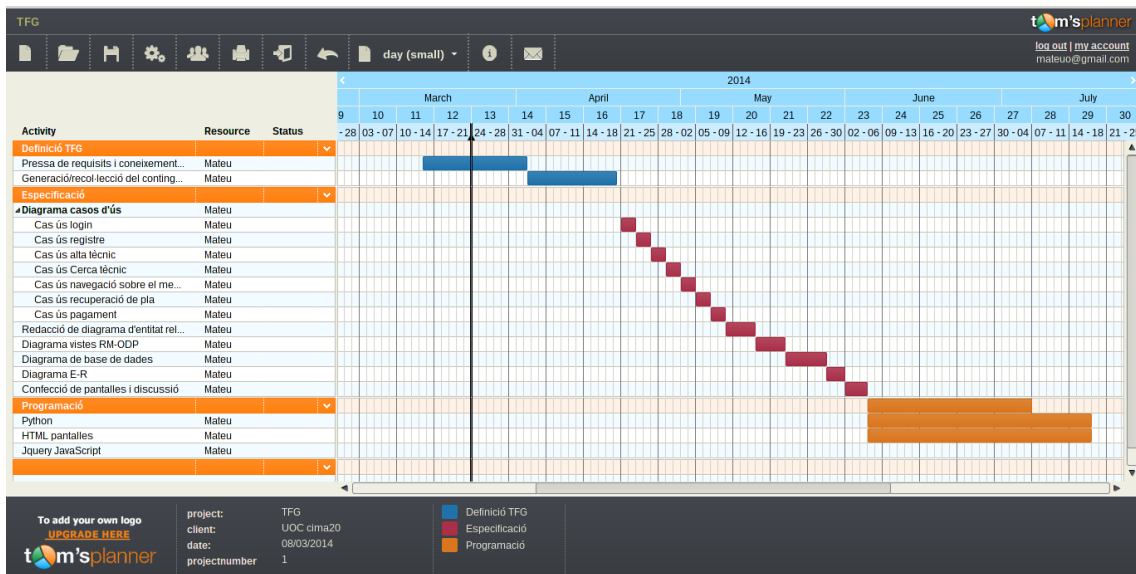


Illustration 1: Planificació del projecte

1.5 Breu sumari de productes obtinguts

Els productes obtinguts són:

- Planificació del treball planificat.
- Diagrama de diferents casos d'ús.
- Diagrames RM-ODP
- Diagrama entitat relació de les classes del projecte.
- Diagrama de la base de dades.
- Pantalles i navegació a través del aplicatiu.
- Producte final aplicatiu instal·lat al núvol EC2 en funcionament

1.6 Breu descripció dels altres capítols de la memòria

A la resta de capítols es tractaran una explicació dels productes obtinguts, així com una explicació de l'ús de l'eina, com una instal·lació en un entorn núvol, així com un pressupost i cost.

2. Recollida de requisits.

2.1 Antecedents legals e introducció al pla de seguretat i salut

Arrel de l'entrada a la Unió Europea tota la normativa es va haver d'adaptar i modificar a directrius europees. En concret, la normativa de prevenció laboral es va haver d'adaptar i de molt poc cos normatiu essencialment :

ORDEN de 9 de marzo de 1971 por la que se aprueba la Ordenanza General de Seguridad e Higiene en el Trabajo.

A tot un desenvolupament que amb aquest esquema es veurà clarament:

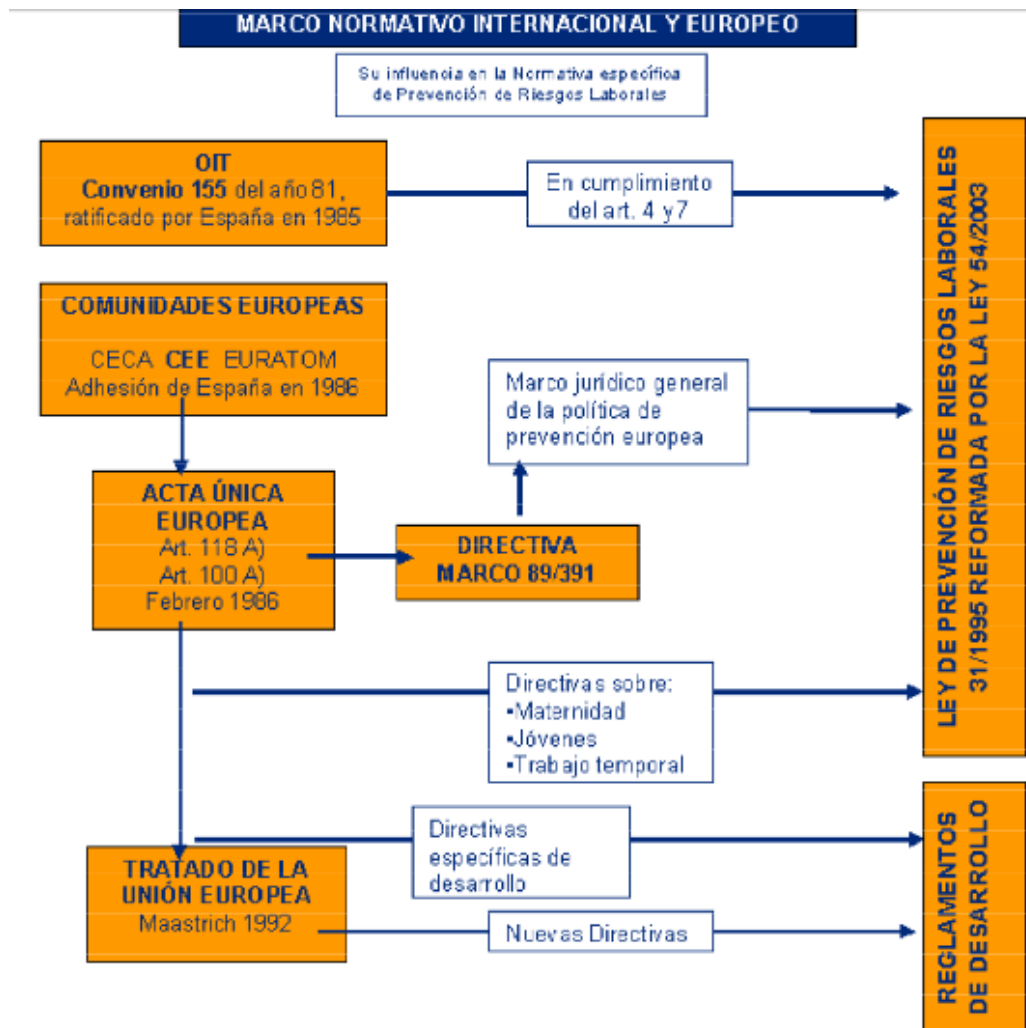


Illustration 2: Marc Normatiu Preventiu

fa que el cos normatiu experimenti un creixement exponencial.

Dins aquests reglaments de desenvolupaments a partir de la llei de prevenció de riscos laborals hi ha tots els relatius a obres de construcció.

Essencialment és el REAL DECRETO 1627/1997, de 24 de octubre, por el que se establecen las disposiciones mínimas de seguridad y salud en las obras de construcción. (BOE nº 256, de 25 de octubre).

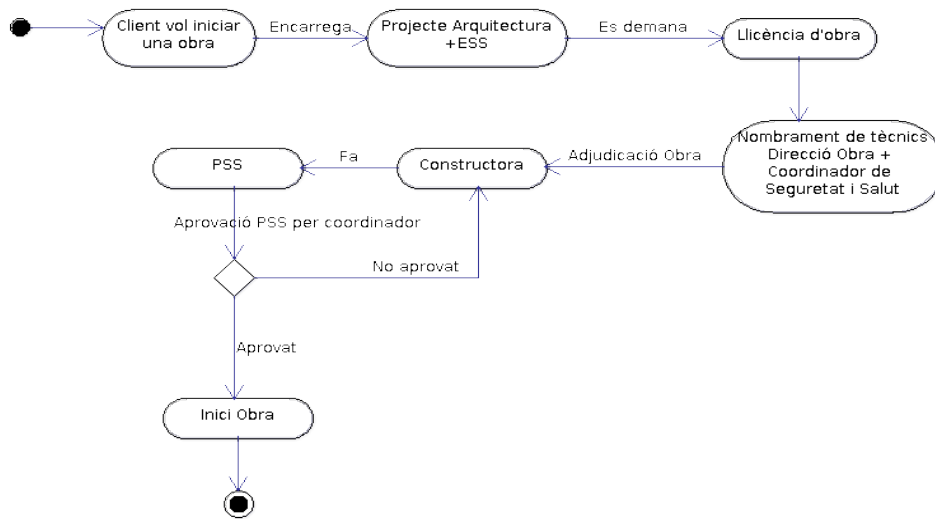


Illustration 3: Procediment per començar una obra de construcció

El següent esquema explica quins són els passos legals per començar una obra de construcció i entendre la rellevància dels PSS.

El RD 1627/97 defineix que:

Obra de construcció: Qualsevol obra pública o privada, on es duguin a terme treballs de construcció o d'enginyeria civil la relació no exhaustiva dels quals figura al següent llistat:

- Excavacions.
- Moviments de terres.
- Construcció.
- Muntatge i desmuntatge d'elements prefabricats
- Condicionaments o instal·lacions.
- Transformacions.
- Rehabilitacions.
- Reparacions.
- Desmantellaments.
- Enderrocs.
- Manteniments.
- Conservació. Feines de pintura i de neteja
- Sanejament.

Contractista: Tota persona física o jurídica que assumeix contractualment amb el promotor amb mitjans humans o materials, propis o aliens, el compromís d'executar la totalitat o part de les obres seguint el projecte i el contracte.

El promotor està obligat a la redacció d'un Estudi de Seguretat i Salut.

Aquest estudi comprendrà:

Memòria descriptiva dels procediments, equips tècnics i medis auxiliars que s'hagin d'utilitzar o que la seva utilització es pugui preveure; identificant els riscos laborals que poden ésser evitats, indicant a tal efecte els riscos laborals que puguin ser evitats, indicant a aquest efecte les mesures tècniques necessàries per a això; relació dels riscos laborals que no es puguin eliminar d'acord amb el que assenyala anteriorment, especificant les mesures preventives i proteccions tècniques tendents a controlar i reduir aquests riscos i valorant la seva eficàcia, en especial quan es proposin mesures alternatives.

Plec de condicions particulars en el que es tindran en compte les normes legals i reglamentàries aplicables a les especificacions tècniques pròpies de l'obra de què es tracti, així com les prescripcions que s'hauran de complir en relació amb les característiques, la utilització i la conservació de les màquines, estris, eines, sistemes i equips preventius.

Plànols en els que es desenvoluparan els gràfics i esquemes necessaris per la millor definició i comprensió de les mesures preventives definides a la Memòria, amb expressió de les especificacions tècniques necessàries.

Mesuraments de totes aquelles unitats o elements de seguretat i salut en el treball que hagin estat definits o projectats.

Pressupost que quantifiqui el conjunt de despeses previstes per a l'aplicació i execució de l'estudi de seguretat i salut.

En aplicació de l'estudi de seguretat i salut o, si escau, de l'estudi bàsic, cada contractista elaborarà un pla de seguretat i salut en el treball en el qual s'analitzin, estudiïn, desenvolupin i complementin les previsions contingudes en l'estudi o estudi bàsic, en funció del seu propi sistema d'execució de l'obra.

En aquest pla s'inclouran, si s'escau, les propostes de mesures alternatives de prevenció que el contractista proposi amb la corresponent justificació tècnica, que no podran implicar disminució dels nivells de protecció previstos en l'estudi o estudi bàsic.

En el cas de plans de seguretat i salut elaborats en aplicació de l'estudi de seguretat i salut les propostes de mesures alternatives de prevenció inclouran la valoració econòmica de les mateixes, que no podrà implicar disminució de l'import total, d'acord amb el segon paràgraf de l'apartat 4 de l'article 5.

El PSS inclourà les mateixes parts que un ESS és a dir:

- Memòria:
 - Dades obra.
 - Dades Tècnics.
 - Dades Treballadors (recurs preventiu, encarregat).

- Dades Constructora.
- Dades Promotor.
- Dades Partides:
 - Descripció.
 - Avaluació de riscos.
 - Maquinaria que intervé.
 - Medis Auxiliars.
 - Mesures preventives.
- Plec de condicions.
- Amidaments.
- Plànols.

2.2 Entrevistes per la recollida de requisits.

La recollida de requisits es fa sobre dues persones rellevants dins l'empresa que té interès en el disseny: El president e impulsor del canvi i obertura a nous mercats i l'arquitecte encarregat de fer els plans de seguretat i salut a l'actualitat.

Però abans de res caldrà fer uns aclariments més planers respecte al que és un pla de seguretat i salut:

Un pla de seguretat i salut en endavant PSS, és un document d'obra amb validesa legal, en el sentit que en cas d'accident laboral és pren com a referència les activitats que s'havien de dur a terme per mantenir la seguretat i evitar els accidents.

Aquest document està compost per una memòria, un pressupost, un pleg de condicions i uns plànols.

El PSS és la resposta que un contractista d'una obra dona a un estudi de seguretat i salut en endavant ESS.

Simplificant el ESS són unes previsions genèriques de la seguretat que s'hauria de dur a terme dins l'obra i el PSS és la concreció de les mesures proposades per el contractista. D'aquesta manera poden ésser més concretes i adaptades al procés productiu de la pròpia empresa així com a la pròpia obra.

En concret, un PSS té les següents parts de la memòria:

- Definició del contractista que farà l'obra.
- Definició del promotor que pagarà l'obra.
- Localització de l'obra.
- Definició de les partides de l'obra.
 - Descripció de la partida.
 - Maquinaria i equips auxiliars usades per l'execució de la partida.
 - Avaluació de riscos de la partida.
 - Mesures preventives a utilitzar tant col·lectives com individuals.

En el món de les obres una partida engloba totes les tasques que és fan per un ofici, en una fase de l'obra anomenada capítol amb un tipus de material determinat. Per exemple s'entenen per capítols:

- Treballs previs a l'inici de les obres.

- Excavacions.
- Fonaments.
- Estructura.
- Picapedrer.
- Guixaire.
- Pintura
- Fusteria.
- Electricitat.
- Llanterneria.
- Cobertes.

Entre d'altres. En canvi una partida s'entén per exemple:

En el cas d'estructures:

- Pilars d'Estructura metàl·lica
- Forjat reticular de caixa perduda.
- Estructura de biguetes semi resistents
- Mur de càrrega de formigó

Com es pot veure és una concreció d'un tipus d'ofici en la utilització d'un determinat tipus de material.

Això fa que la granularitat en la descripció de l'obra sigui molt exhaustiva i molt fina. Tant és així que aquest tipus de descripció en capítols i partides s'utilitza per fer els amidaments i aconseguir obtenir el pressupost de l'obra. En el nostre cas els amidaments només són necessàries per comptar els equips individuals (cascs, botes,...) i les proteccions col·lectives (xarxes, barreres,...) i valorar-ho.

Una vegada descrit el producte esperat els requisits expressats per els responsables de l'empresa agafen un significat més profund.

El president destaca com a punts rellevants de l'aplicació:

- Que generi ingressos, és a dir que el producte es pugui cobrar de manera segura.
- Que estigui a l'abast 24x7
- Que la plataforma que implementi sigui la mateixa de la que l'empresa disposa fins ara i vol seguir desenvolupant.
- Que els clients siguin capaços de fer-ho anar tot sols. Les limitacions en el coneixements informàtics i preventius del tipus d'empreses usuàries és conegut, al final ho faran anar els administratius.
- Que de qualcuna manera el client tingui accés a definir les seves pròpies partides a partir de les comunes que es tenen desenvolupades dins l'empresa i d'aquesta manera concretar per cada empresa i obra la seva manera de construir.
- També la de poder triar les mesures preventives per partida que considerin oportunes o disposin.
- Que respecte a la responsabilitat legal, s'eximeixi a l'empresa proveïdora de qualsevol responsabilitat en l'ús de l'eina, és a dir que en cas d'accident laboral a una obra on s'ha fet el PSS amb l'aplicació no es pugin demanar responsabilitats del seu mal ús.
- Que el cobrament sigui senzill, immediat i previ a la generació del PSS.

El arquitecte explica que l'aplicatiu ha de permetre:

- Que els clients pugin registrar-se
- Conservar els seus plans.
- Editar-los / reutilitzar-los.
- Designar les partides del pla a partir de plantilles predefinides:
 - Habitatge unifamiliar.
 - Habitatge plurifamiliar
 - Piscina
 - Reforma d'interior
- Que disposin d'un magatzem de treballadors, tècnics, promotors, constructors que sigui personal per usuari registrat.
- Que els plans es pugin fer amb facilitat (intuïtivament).
- Que pugin anomenar els recursos preventius específicament.
- Que els pressuposts siguin automàtics amb poques dades donades dins la definició d'obra. Per exemple que si donam el perímetre de l'obra i el nombre de pisos sobre rasant ja podem inferir el número de metres de barana de forjats. O sabent el número de treballadors poder determinar el número de cascs, botes, arnesos de seguretat etc.
- Que existeixi un repositori de plànols de detall que permetin explicar cada mesura preventiva usada.

2.3 Diagrames identificatius dels casos d'ús.

Es trien els casos d'ús més representatius com són el de login, el de pagament, el d'entrada de dades i el de selecció de partides .

2.3.1 Login

Resum de la funcionalitat: Permetre que un usuari ja registrat en el sistema , accedeixi mitjançant usuari i contrasenya

Actors: Usuari i Administrador

Precondició: Usuari registrar al sistema i no logejat al sistema.

Postcondició : L'usuari està logejat al sistema.

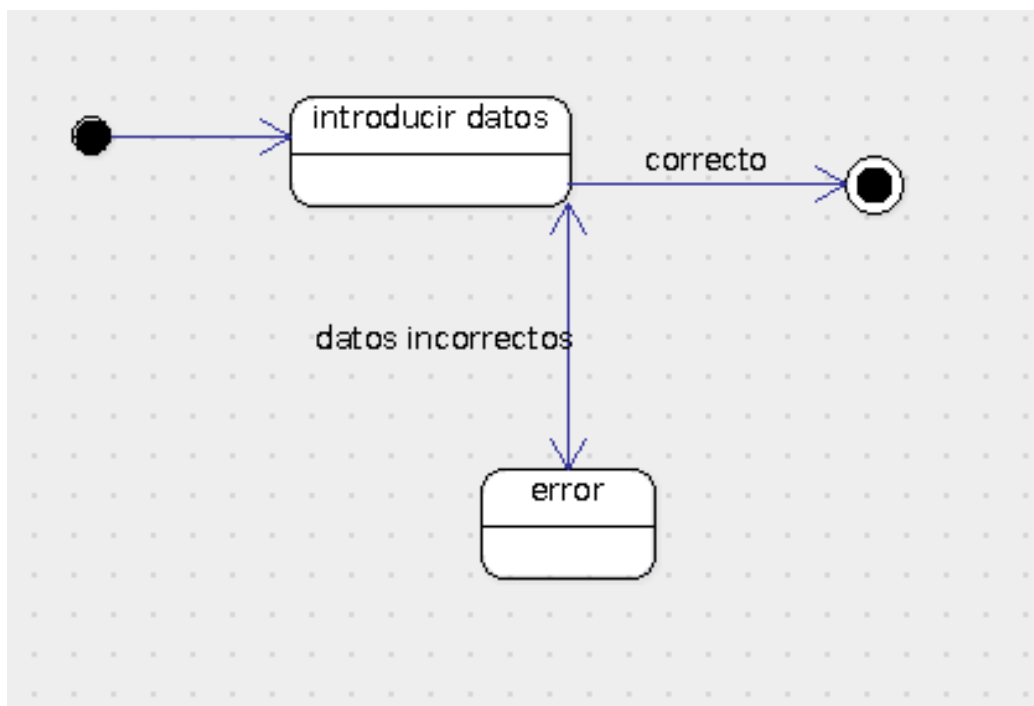


Illustration 4: Diagrama d'estats del cas d'ús de login

L'usuari introduirà el seu usuari i la seva contrasenya i demanarà la conformitat de les dades. Si aquests són correctes serà redirigit a la pàgina principal, si no error.

2.3.2 Pagament

Resum de la funcionalitat: Permetre a un usuari logejat que ha generat correctament un pla de seguretat, i que ha esgotat la promoció, el seu pagament per poder-lo descarregar. El pagament es realitza amb una aplicació externa (en aquest cas Paypal).

Actors: Usuari, Paypal

Precondició: Usuari registrat amb pla generat correctamen que ha acabat la promoció i que per davallar-lo s'ha de pagar.

Postcondició: El pagament s'ha realitzat i se li permet davallar el pla.

2.3.3 Selecció de partides

Resum de la funcionalitat:

Permet a un usuari definir les partides agrupades per capítols i triar els que s'ajusten millor al tipus d'obra.

Si vol pot agafar un tipus predifinit d'obra que engloba un determinat conjunt de partides i després fer un ajust més fi sobre les partides triades.

Actors: User

Precondició:

Usuari d'alta a la base de dades i haver emplenat tots els anteriors camps, és a dir haver predifinit tècnics de l'obra, haver predifinit el contractista, haver predifinit el promotor i l'obra. Si no té accés a la pantalla de partides.

Post condició:

L'obra disposa de les partides associades

Diagrama d'estats

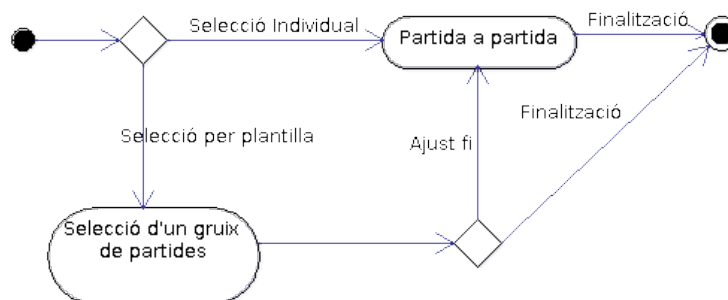


Illustration 5: Diagrama d'estats per la selecció de partides

Diagrama cas d'ús

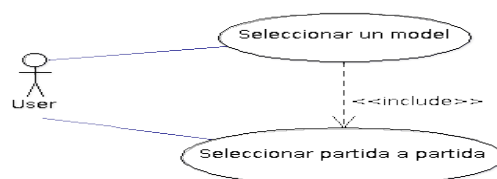


Illustration 6: Diagrama cas d'ús de la selecció de plantilles

2.3.4 Cas d'ús entrades de dades

Resum de la funcionalitat:

Permet als usuaris entrar unes dades noves que poden ésser del tipus (Promotor, tècnic, treballador, obra o contractista) o seleccionar-les d'un que prèviament ells hagin introduït . El nou introduït tan així com el seleccionat queden seleccionats com dada (Promotor, tècnic, treballador, obra o contractista) actual

Actors:

Usuari

Precondició:

Usuari logejat i que s'hagi seguit la seqüència d'entrada de dades corresponent, ja que no és pot introduir determinades dades abans que unes altres.

Postcondició:

Dada seleccionada presa com actual.

Diagrama d'estats

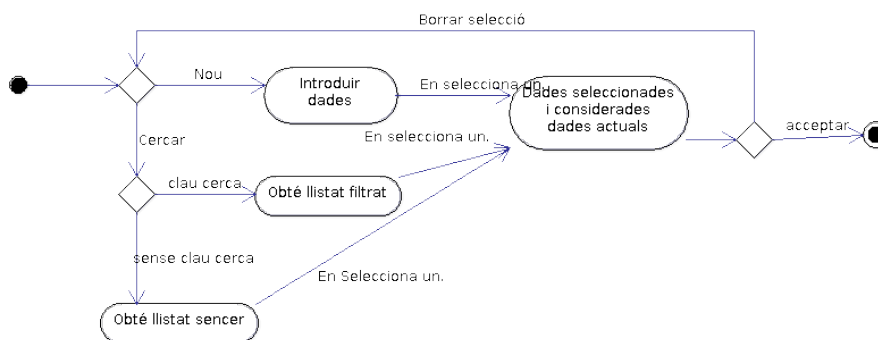


Illustration 7: Diagrama d'estats genèric per l'entrada de dades a l'aplicatiu

Diagrama del cas d'ús

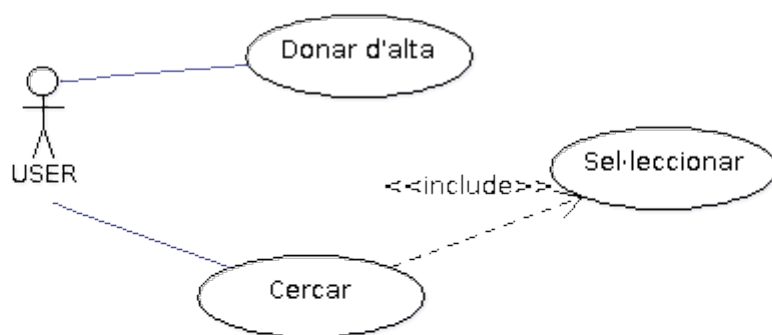


Illustration 8: Diagrama de cas d'ús per l'entrada de dades a l'aplicatiu

3. Anàlisi

3.1 Adaptació de la solució al «framework»

Donat que el «framework» de Django utilitza un model MVC, va determinar utilitzar la següent arquitectura de classes proveïdes per el «framework».

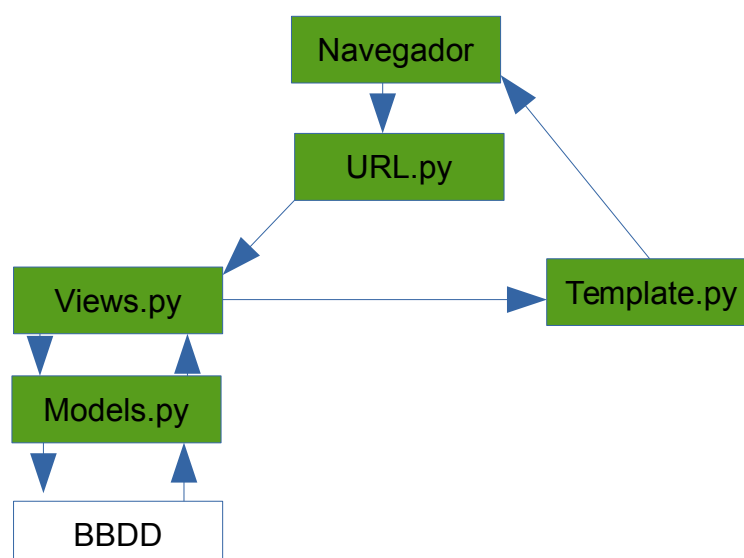
Curiosament per Django el model MVC no és estrictament controlador ja que tot el «framework» fa de controlador, a ells els agrada dir-ho MTV (model-template-view). És a dir model, plantilla i vista, on el model és la representació de la base de dades (persistència) mitjançant classes, la plantilla són els fitxers de Html dinàmic cridats per les vistes i les vistes és on resideix la lògica del negoci de tot l'aplicació.

Concretament la vista d'una funció particular és cridada a través d'una URL particular. A Django una vista descriu quina dada és representada, però normalment la vista delega en una plantilla (template) la seva representació. Els «templates» descriuen com es representa aquesta dada.

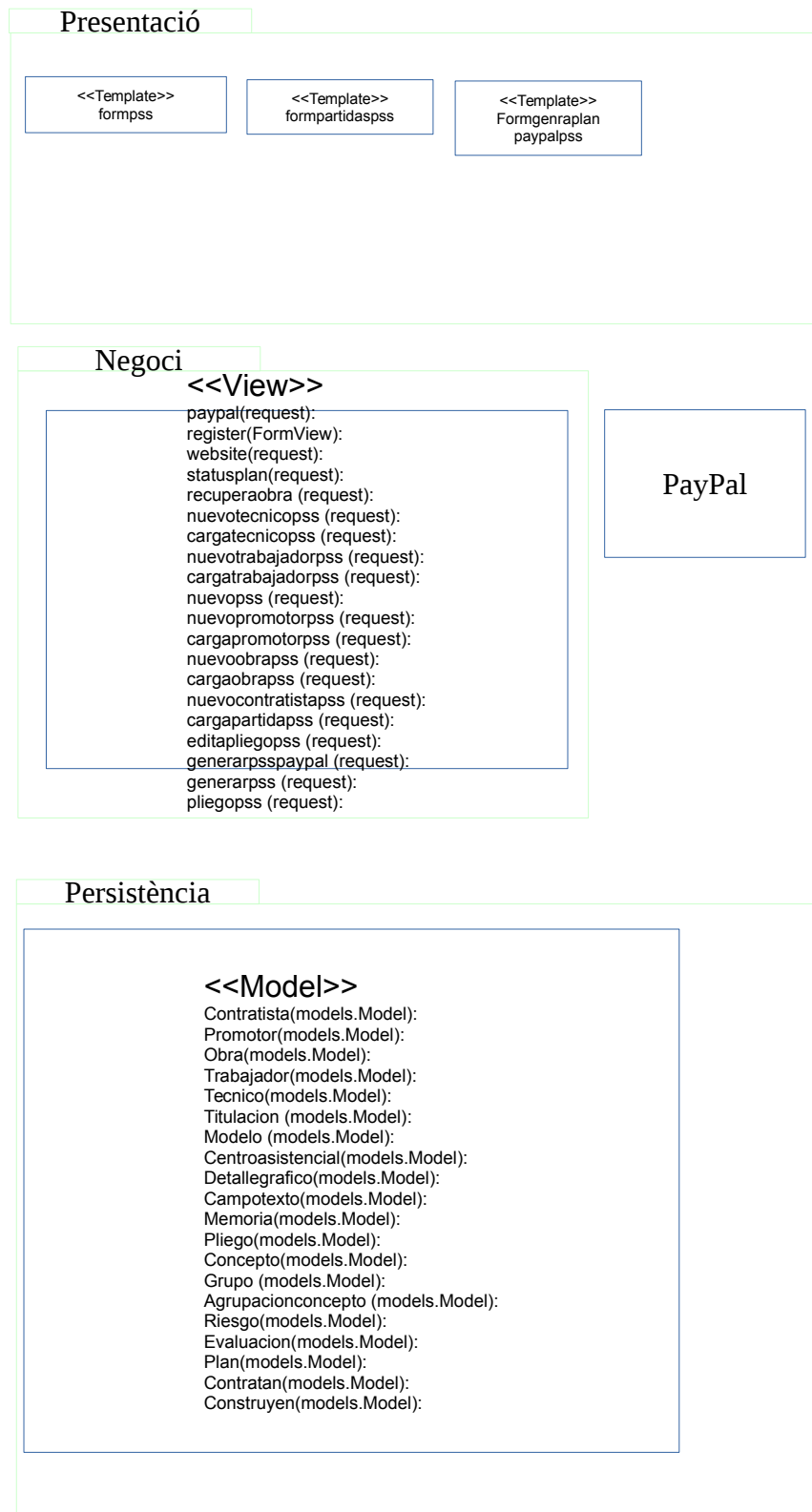
- El gestor de Django per l'accés a la BBDD, realment l'accés en Django és transparent, ja que per aquest framework les taules estan mapejades com a classes a un fitxer anomenant «models.py», els camps de la taula s'accedeixen com atributs de la classe.
- Cada crida URL es responda per una funció (vista) definida dins el fitxer views.py
- Cada vista produeix un resultat sobre una plantilla (template), que genera html dinàmic. Aquest html dinàmic és generat per un llenguatge de plantilles propis. Django no deixa embeure python pur dins de les plantilles per evitar vulneracions del principi de disseny DRY (don't repeat yourself).

D'aquests comentaris s'extrau el següent diagrama explicatiu del funcionament del «framework»

Drawing 1: Diagrama del funcionament del framework



Per tant el diagrama UML en capes, representació del punt de vista de la computació, és a dir, el disseny intern de la funcionalitat de l'aplicació mitjançant diagrames de components dintre d'una arquitectura en capes. Cada capa a nivell lògic es representa en UML com un paquet. En aquest diagrama s'ha tingut en compte el «framework».



3.2 Diagrames de seqüència.

Es trien els casos d'ús més representatius com són el de login, el de pagament, el d'entrada de dades i el de selecció de partides .

3.2.1 Login

L'usuari introduirà el seu usuari i la seva contrasenya i i demanarà la conformitat de les dades. Si aquests són correctes serà redirigit a la pàgina principal, si no error.

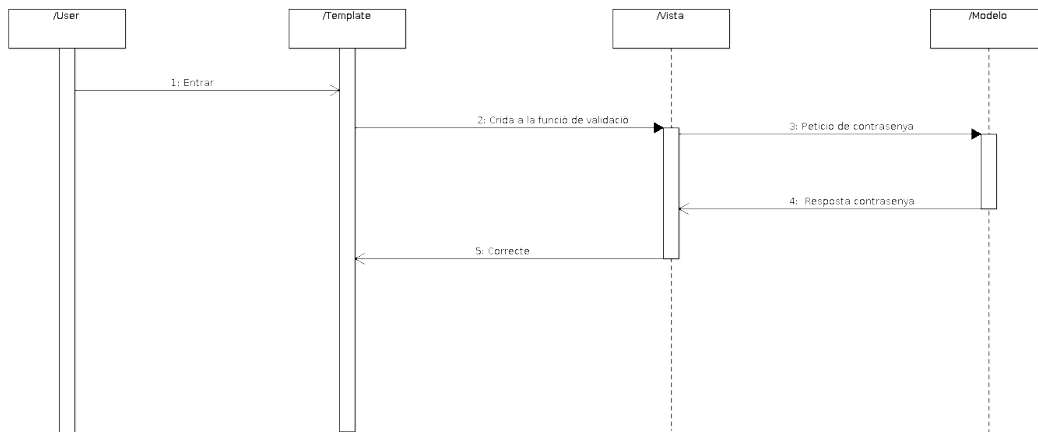


Illustration 9: Diagrama de seqüència de login

La vista de login és contribuïda per el framework, llavors no és accessible des de les vistes d'usuari.

3.2.2 Pagament

Resum de la funcionalitat: Permetre a un usuari logejat que ha generat correctament un pla de seguretat, i que ha esgotat la promoció, el seu pagament per poder-lo descarregar. El pagament es realitza amb una aplicació externa (en aquest cas Paypal).

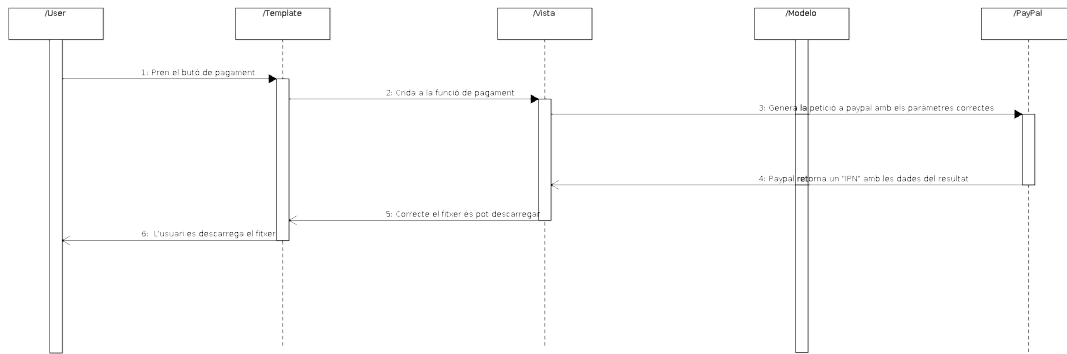


Illustration 10: Diagrama de seqüència de pagament

3.2.3 Selecció de partides

Resum de la funcionalitat:

Permet a un usuari definir les partides agrupades per capítols i triar els que s'ajusten millor al tipus d'obra.

Si vol pot agafar un tipus predifinit d'obra que engloba un determinat conjunt de partides i després fer un ajust més fi sobre les partides triades.

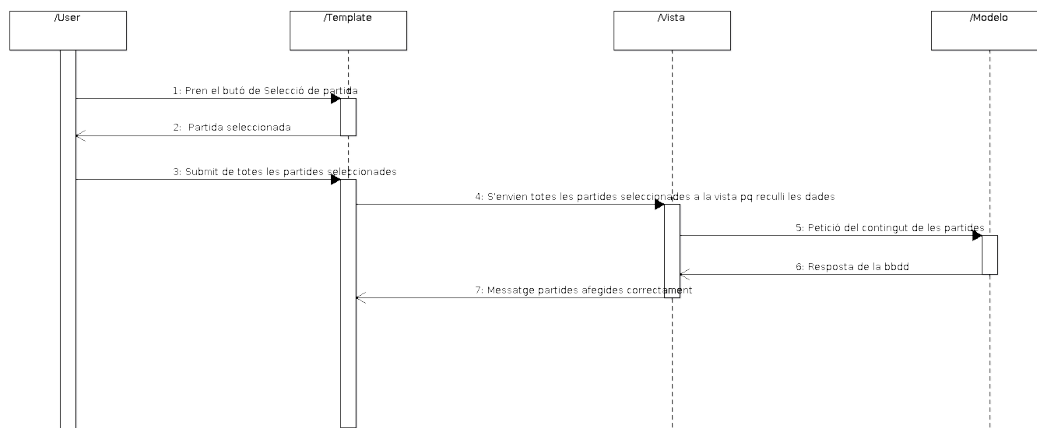


Illustration 11: Diagrama de seqüència de selecció de partides

3.2.4 Cas d'ús entrades de dades

Resum de la funcionalitat:

Permet als usuaris entrar unes dades noves que poden ésser del tipus (Promotor, tècnic, treballador, obra o contractista) o seleccionar-les d'un que prèviament ells hagin introduït . El nou introduït tan així com el seleccionat queden seleccionats com dada (Promotor, tècnic, treballador, obra o contractista) actual

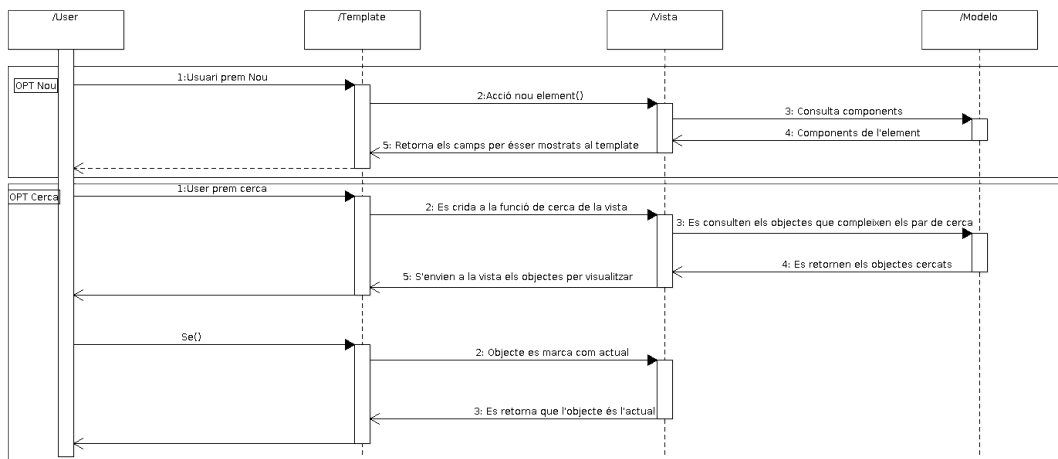


Illustration 12: Diagrama de seqüència d'entrada de dades

Degut a que en funció de l'objecte que es vulgui cercar o crea nou intervenen diferents vistes i models s'ha fet la següent taula. Fixem-nos que el template és el mateix per cadascun d'ells, el que simplifica molt la reutilització d'elements. Tot i així en la generació de les vistes s'hagués pogut reutilitzar més codi fent que la mateixa vista servis per distints models, però s'hauria perdut en claretat del codi. De cap manera el model s'hagués pogut reutilitzar, degut a que cada objecte tenia atributs diferents.

Vista (View) *	Plantilla (template)	Model **
nuevotecnico	formpss.html	tecnico
cargatecnicopss	formpss.html	tecnico
nuevotrabajador	formpss.html	trabajador
cargatrabajador	formpss.html	trabajador
nuevopss	formpss.html	plan
cargapss	formpss.html	plan
nuevopromotor	formpss.html	promotor
cargapromotorpss	formpss.html	promotor
nuevoobrapss	formpss.html	obra
cargaobrapss	formpss.html	obra
nuevocontratista	formpss.html	contratista
cargacontratista	formpss.html	contratista
cargapartidapss	formpartidapss.html	partida

3.3 El model de dades

Com es pot veure s'han creat les classes i les relacions entre elles. Hem de destacar la relació quaternària existent entre plan de seguretat, Promotor, Obra i Contractista. Aquesta existeix degut a que:

Un contractista contrata un obra que és d'un promotor e hi presenta un pla de seguretat.

I després aquest mateix contractista construeix l'obra amb treballadors per això existeix la següent relació ternària.

Respecte als tècnics estan associats a l'obra.

El següent punt clau de l'estructura és el «Concepte». Aquesta classe és capdal ja que engloba al contingut del pla tant pot ésser una partida, un medi auxiliar (bastida,escala,...), un epi(equip de protecció individual, casc, bota, etc), com una maquinaria, ja que dins el pla encara que tingui continguts diferents és tractat igual i la classe disposa dels mateixos conceptes.

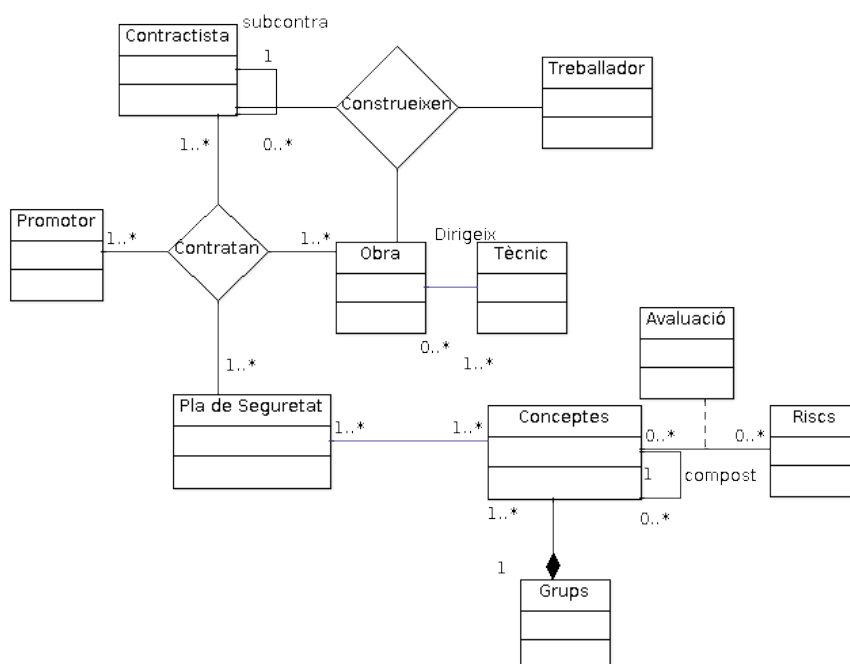
Per exemple existeix la relació reflexiva del concepte amb sí mateix degut a que una partida pot disposar de medis auxiliars, màquines, equips de protecció. A la vegada tots aquests conceptes són avaluats de la mateixa manera és a dir que tener uns riscos que és poden avaluar amb una possibilitat de que succeeixi el fet i la gravetat del mateix.

La relació entre risc i concepte és l'avaluació i cada avaluació pot tenir una o varies mesures preventives.

D'aquesta manera s'explica les relacions entre les classes que intervenen en el pla.

El concepte és un punt cabdal que mereix un poc d'explicació.

Un concepte és la unitat bàsica del pla. A més una vegada analitzada les dades apareix que pot ésser una partida, un medi auxiliar, una maquinaria, ... ja que aquests conceptes estan formats per les mateixes parts e interactuen de la mateixa manera. La reflexivitat del concepte, és a dir que una partida pot tenir medis auxiliars o maquinaries és clau per entendre-ho.



4. Disseny Tècnic

4.1 Introducció: bases de l'arquitectura de desenvolupament.

El disseny tècnic ve limitat o potenciat, en el meu cas potenciat per el framework i el llenguatge sobre el que està implementat.

Tant python com django es converteixen amb unes eines per crear l'aplicatiu amb molt poc temps.

Per la mida de l'aplicació ha estat d'allò més convincent degut a que la potència de Python i l'estructura del framework eren perfectes per el projecte que es volia emprendre.

4.2 Python

Python és un llenguatge de programació d'alt nivell de propòsit general. Guido van Rossum el va crear el 1991. Combina una potència remarcable amb una sintaxi clara i entenedora.

Molt sovint és comparat amb altres llenguatges com Java, Tcl, Perl o Scheme. Utilitza sagnats com a delimitadors de blocs, fet poc freqüent als llenguatges de programació.

Una altra de les característiques d'aquest llenguatge és el tipificat dinàmic i la capacitat per interpretar el codi en temps d'execució, en contradicció d'altres llenguatges com ara C, que ho fan en temps de compilació.

4.3 Framework Django

L'aplicació és desenvoluparà sota el «framework» de python «Django», el qual ens permet partir amb un model Model – Vista – Plantilla, la qual cosa permet d'una manera molt senzilla i molt ràpida un desenvolupament complex d'un model client servidor on el client és web.

Va ser desenvolupat per gestionar diverses pàgines orientades a notícies de la World Company de Lawrence, Kansas, i va ser alliberada al públic sota una llicència BSD 1 el juliol de 2005.

Django és avui un Framework de desenvolupament web de codi obert, escrit en Python , que compleix en certa mesura el paradigma del Model Vista Controlador.

Python és usat en totes les parts del Framework , fins i tot en configuracions, arxius, i en els models de dades.

L'objectiu fonamental de Django és facilitar la creació de llocs web complexos posant èmfasi en la reutilització, la connectivitat i extensibilitat de components, el desenvolupament ràpid i el principi “ No et repeteixis” (DRY , de l'anglès Don't Repeat Yourself).

Django també proporciona una opció administrativa CRUA (crear, llegir, actualitzar i eliminar) de la interfície que es genera de forma dinàmica a través de la introspecció i configurar a través de models d'administració.

Algunes pàgines utilitzen Django des de fa temps, com la web d' Spotify . Aquesta és una pàgina des de la qual es permet escoltar música a través d'Internet sense haver de comprar-la o descarregar-la.

El seu nom li ve donat en honor al guitarrista de jazz Django Reinhardt .

Els orígens de Django com administrador de pàgines de notícies són evidents en el seu disseny, ja que proporciona una sèrie de característiques que faciliten el desenvolupament ràpid de pàgines orientades a continguts.

Per exemple, en lloc de necessitar que els desenvolupadors escriguin controladors i vistes per a les àrees d'administració d'una pàgina, Django incorpora una aplicació per administrar els continguts.

Aquesta aplicació es pot incloure com a part de qualsevol pàgina feta amb Django i pot administrar diverses pàgines fetes amb Django a partir d'una sola instal·lació. Permet la creació, actualització i eliminació d'objectes de contingut, portant un registre de totes les accions realitzades sobre cada un.

Django també proporciona una interfície per administrar els usuaris i els grups d'usuaris (incloent una assignació detallada de permisos).

A més dels arxius que s'han comentat Django també inclou:

- Un sistema de redirecció d'URL.
- Un sistema de comentaris.
- Eines per a syndicar contingut via RSS i/o Atom.
- Un servidor web independent i lleuger per a desenvolupament i proves.
- Un sistema de validació que permet diferenciar entre les etiquetes HTML i els valors susceptibles de ser guardats en la base de dades.
- Un entorn d'emmagatzematge de memòria cau que pot ser utilitzat per qualsevol dels mètodes de memòria cau.
- Suport per les classes de Middleware que poden intervenir en diverses etapes de processament de sol·licituds i dur a terme les funcions habituals.
- Un sistema intern que permet comunicar esdeveniments entre components d'una aplicació a través de senyals preestablertes.
- Un sistema d'internacionalització, incloent les traduccions dels propis components de Django en diferents idiomes.
- Un sistema de serialització que pot produir i llegir representacions XML(extendible Markup Language) i/o JSON (JavaScript Object Notation) de les instàncies de model de Django .
- Un sistema per estendre les capacitats del motor de plantilles.
- Una interfície creada amb Python per a unitàries. El nucli de Django consisteix en una assignació entre objecte i relació que hi ha entre:
- Els models de dades (definit com les classes de Python) i una base de dades relacional (" model.py "),

- un sistema per processar les sol·licituds amb un sistema de plantilles web("views.py") i una expressió regular obtinguda per URL ("Controller").

Tot i això existeixen diferències entre la nomenclatura Django i el patró MVC.

Aquest fet es deu a que el disseny de Django no es va voler vincular a res en particular sinó que es volia desenvolupar una eina que funcionés el millor possible. Si bé és cert que s'assembla molt a la implementació del patró MVC , per Django la view descriu quines dades seran presentades i no com es veuran. En el format de les dades és on entren en joc els templates .

Django aparenta implementar el patró MVC , però el " controlador " de MVC és anomenat " vista " en Django i la " vista " de MVC és " template " en Django .

Es diu que el controller d'un MVC clàssic està representat pel propi Framework .

És a dir, el sistema que envia un request a la vista corresponent, d'acord amb la configuració d'URL de Django (arxiu de configuració). En el cas de voler fer una correspondència clara entre Django i MVC seria un Framework MTV : Model, Template, View . Tenint en compte l'arquitectura, veurem a grans trets com es processa un request en Django .

El flux de dades es pot representar com

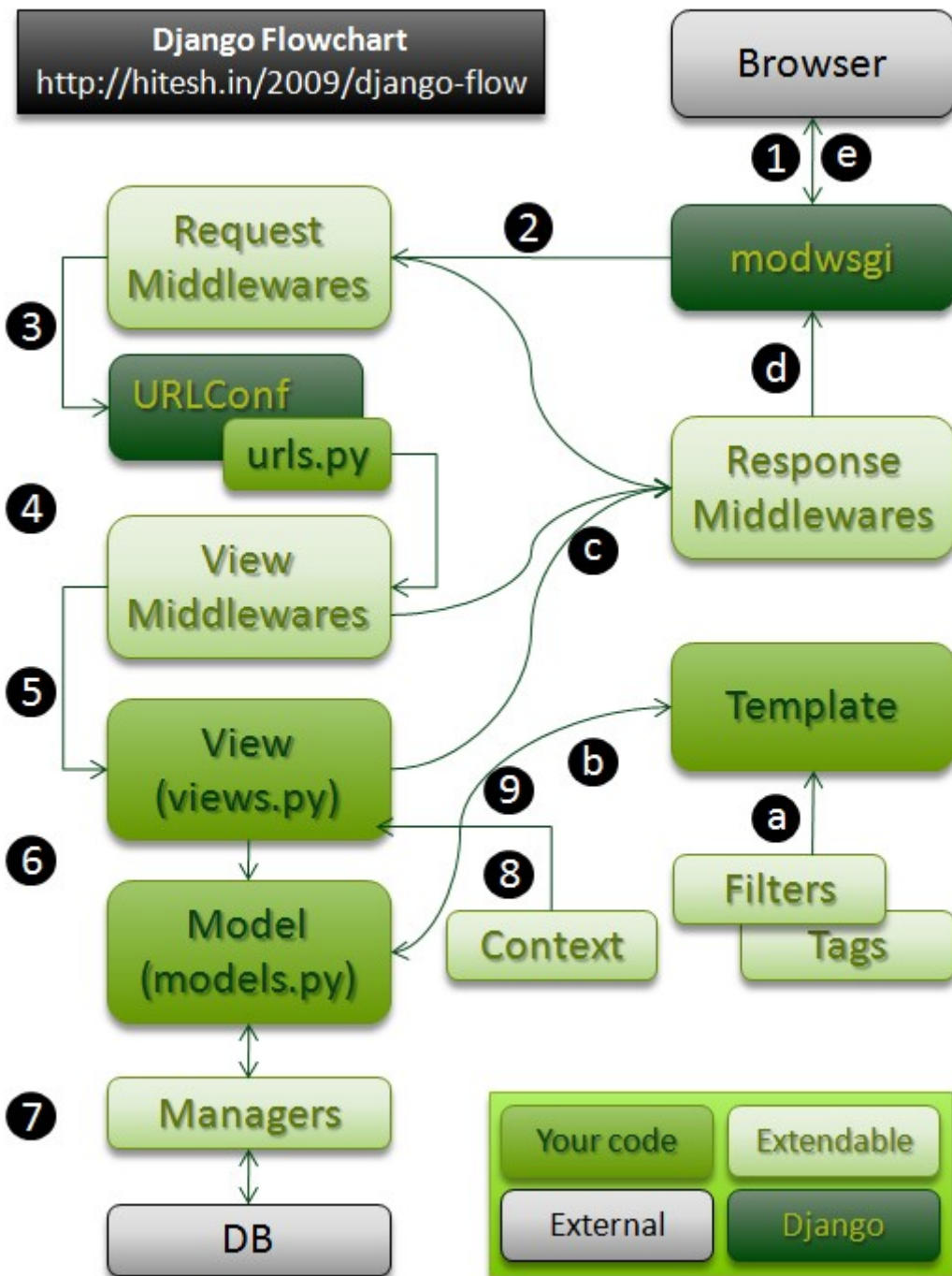


Illustration 13: Diagrama de fluxe del framework

1. Un usuari reclama una pàgina.
2. La sol·licitud de la pàgina arriba al Middleware que en poden fer un pretractament.
3. La URL apunta a la vista usant urls.py
4. Les vistes de Middleware són cridades les quals poden manipular o respondre la petició
5. La funció de la vista és invocada
6. La vista podria opcionalment accedir a la data a través dels models
7. Totes les interaccions del model de la BBDD estan fets via un Manager o controlador.
8. Les visten poden emplear un context especial si és necessari
9. El context es passa a la plantilla per renderitzar

- a. La plantilla usa filtres i marques per renderitzar la sortida.
- b. La sortida és retornada a la vista.
- c. HTTPResponse és enviada com a la resposta de Middelw
- d. Qualsevol resposta del middleware pot enriquir la resposta o retornar-ne una altre completament distinta
- e. La resposta és enviada cap a el navegador del usuari.

4.3.1 El gestor de BBDD de Django

Django disposa d'una capa d'abstracció de la base de dades molt important el que permet no preocupar-se de quina base de dades s'utilitza.

La manera d'aconseguir aquest fet és mitjançant la definició de les taules de la base de dades com a classes del fitxer "models.py" i després el propi framework proveix d'una sintaxi pròpia d'orientació a objectes per interactuar amb la mateixa.

Per tant tampoc és necessari establir consultes sobre la base de dades amb SQL query, si no amb notació orientada a objectes com he comentat abans. Tan mateix per consultes molt complexes el framework permet utilitzar SQL en cru com qualsevol altre llenguatge.

Per il·lustrar aquests fets valgui l'exemple següent:

```
class Condicion(models.Model):
    idcondicion = models.AutoField(primary_key=True)
    grupo_riesgo = models.ForeignKey("Grupo_Riesgo")
    texto = models.TextField()
    fecha_creacion = models.DateTimeField()
    def __unicode__(self):
        return u'%s' % self.idcondicion
class Meta:
    db_table = u'condicion'
    app_label = 'cima20'

class Admin:
    pass
```

Com es pot veure es defineix la classe Condicion amb les variables idcondicion, grupo_riesgo, texto, fecha_creación. Com es pot observar es defineixen el tipus de variables com a objectes de la classe pare "models", aquests són el primary key (AutoField), etc.

A partir de la def __unicode__ es permet definir el nom que retornarà per cada fila de la taula de condició quan se li demani.

La class Meta, permet definir el nom de la taula y l'aplicació que l'utilitzarà, ja que Django permet dins el mateix projecte definir vèries aplicacions.

La class Admin és la més generosa amb diferència ja que el propi frame work proveix d'un entorn web d'administració de la base de dades on es té accés a totes les taules on es defineixi aquest Meta.

Com es pot veure permet unes facilitats per gestionar les bases de dades que estan fora de tot dubte. És més les bases de dades que utilitza cada aplicació poden ésser tan locals com remotes el que permet aplicar el framework sobre aplicacions distribuïdes.

Les bases de dades suportades de manera nativa són:

- Mysql
- Postgresql
- SQLite

Com he comentat abans, la manera de definir la base de dades sobre el framework és a través del fitxer "settings.py", adjunt un exemple:

```
DATABASE_ENGINE = 'sqlite3'
```

```
DATABASE_NAME = 'primary.db'
```

```
DATABASE_USER = ''
```

```
DATABASE_PASSWORD = ''
```

```
DATABASE_HOST = ''
```

```
DATABASE_PORT = ''
```

```
DATABASES = dict(  
    primary = dict(  
        DATABASE_ENGINE=DATABASE_ENGINE,  
        DATABASE_NAME=DATABASE_NAME,  
        DATABASE_USER=DATABASE_USER,  
        DATABASE_PASSWORD=DATABASE_PASSWORD,  
        DATABASE_HOST=DATABASE_HOST,  
        DATABASE_PORT=DATABASE_PORT,  
    ),  
    secondary = dict(  
        DATABASE_ENGINE=DATABASE_ENGINE,  
        DATABASE_NAME='secondary.db',  
        DATABASE_USER=DATABASE_USER,  
        DATABASE_PASSWORD=DATABASE_PASSWORD,  
        DATABASE_HOST=DATABASE_HOST,  
        DATABASE_PORT=DATABASE_PORT,  
    )  
)
```


),
)

On cada paràmetre es força autoexplicatiu.

4.4. Diagrama entitat relació de les dades.

A partir del diagrama de classes del punt 3 podem anar desenvolupant-lo fins obtenir el següent diagrama entitat relació més complet

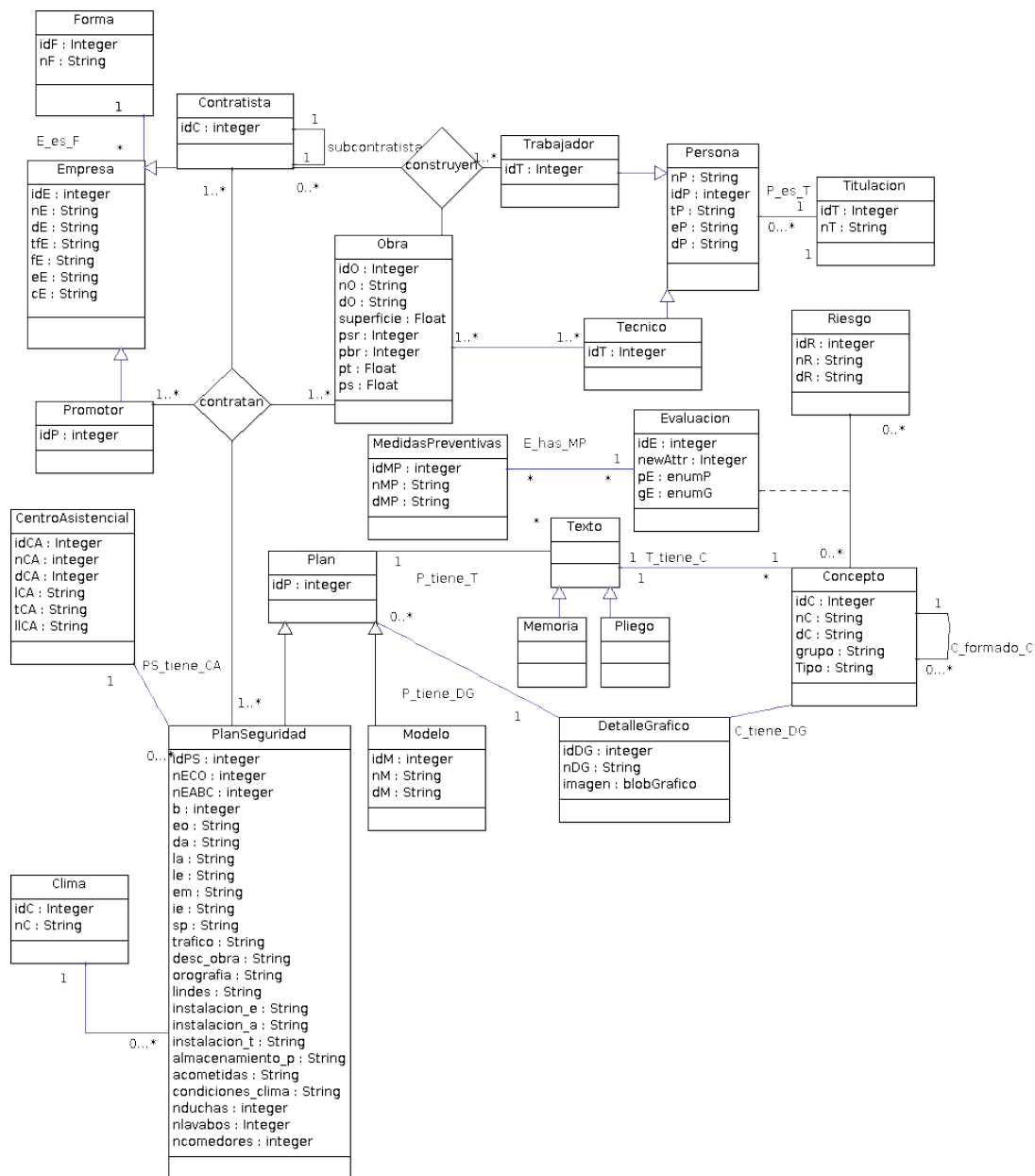


Illustration 14: Diagrama entitat relacio

4.6. Diagrama final de les dades.

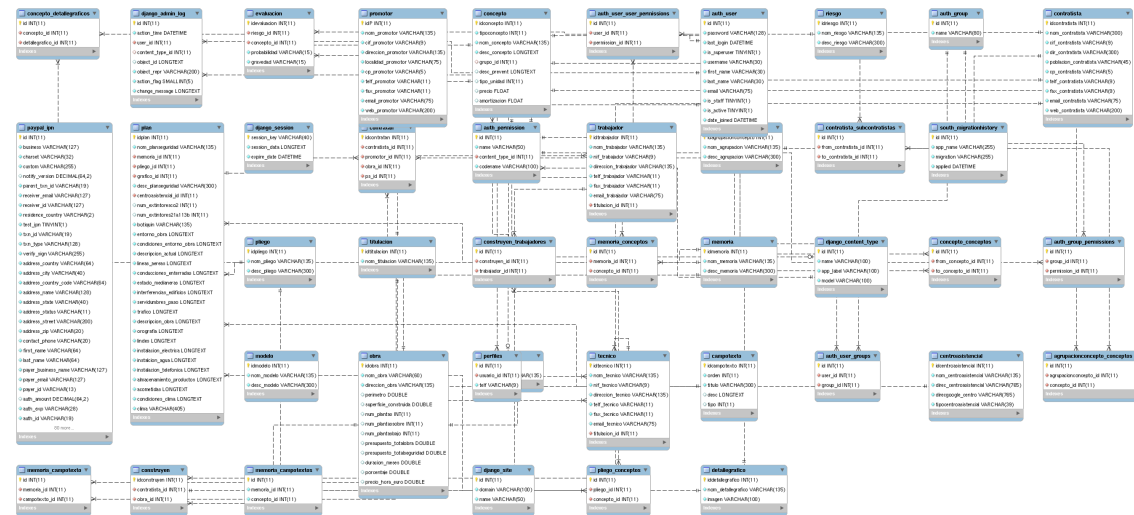


Illustration 15: Diagrama final de Taules

Aquest diagrama final inclou les taules que s'han inclòs per el sistema de logeig del framework, així com les taules afegides per controlar el sistema de pagament i òbviament les taules del diagrama anterior.

4.7 Esbos de les pantalles

Realitzar aquest exercici sobre les pantalles va desvetllar que la gestió de promotors, constructors, tècnics, treballadors i obra eren iguals. Ja que sobre aquestes dades es feien les mateixes operacions (crear, seleccionar, des seleccionar i cercar).

Llavors després d'aquesta conclusió que es podrà observar a partir de les pantalles proposades a partir del següent punt, només es té l'esbos de només un dels conceptes (tècnic o constructor o treballador o promotor o obra) i d'un menú principal aquí representat per fitxes, però que per html es va canviar amb botons que carregen pàgines diferents.

En aquest esbos es veu les pestanyes de les que havia parlat abans i centrats en obra: un primer botó per generar-ne una de nova o fer una cerca.



Illustration 16: Pantalla d'obra

Abaix de la pantalla es representaran els resultats amb un botó per seleccionat com es veu en el següent esbos.

Una vegada cercat i seleccionat l'element es representa com element actual amb la possibilitat de esborrar-lo i de poder repetir la sistemàtica anterior.



Illustration 17: Pantalla d'obra retorna resultats cerca

En el cas de les partides, part fonamental del aplicatiu el que es cercava era una manera de tenir agrupades per capítols totes les partides disponibles i que l'usuari les pogués seleccionar una a una o amb un perfil o predefinit, tal com es diu als requisits.

La proposta de pantalla és la següent:

The screenshot shows a web application interface with a navigation menu at the top and a main content area. The navigation menu includes the following items: Personas PSS>, Obra, Contratistas, Promotor, Partidas, PSS, Reset PSS, Estados PSS, RecuperaPSS, and Logout. The main content area is divided into two columns and three rows, displaying six chapters (Capitol 1 through Capitol 6). Each chapter contains two items (Partida 1 and Partida 2), each with an unchecked checkbox.

Capitol	Partida 1	Partida 2
Capitol 1	<input type="checkbox"/>	<input type="checkbox"/>
Capitol 2	<input type="checkbox"/>	<input type="checkbox"/>
Capitol 3	<input type="checkbox"/>	<input type="checkbox"/>
Capitol 4	<input type="checkbox"/>	<input type="checkbox"/>
Capitol 5	<input type="checkbox"/>	<input type="checkbox"/>
Capitol 6	<input type="checkbox"/>	<input type="checkbox"/>

Illustration 18: Proposta Pantalla Partides

4.8. Pantalles proposades

Com he comentat abans les pantalles proposades a partir dels esbosos anteriors són les següents:
Per cerca Tècnic

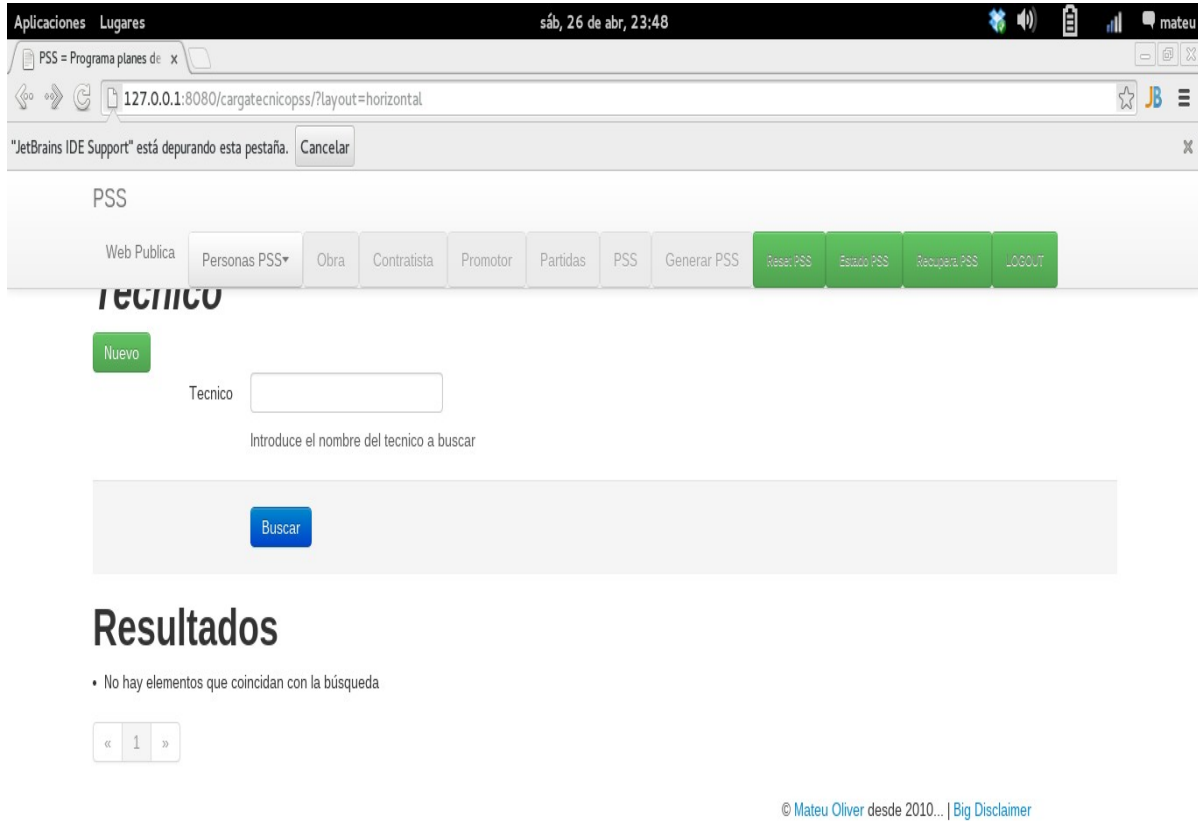


Illustration 19: Pantalla definitiva tècnics

Per resultats de cerca Tècnic:

The screenshot shows a web browser window with the following elements:

- Browser Header:** "Aplicaciones Lugares", "sáb, 26 de abr, 23:50", and user "mateu".
- Address Bar:** "127.0.0.1:8080/cargatecnicopss/?layout=horizontal".
- Notification:** "JetBrains IDE Support" está depurando esta pestaña. [Cancelar]
- Navigation Menu:** Web Publica, Personas PSS, Obra, Contratista, Promotor, Partidas, PSS, Generar PSS, Reser PSS, Estado PSS, Recupera PSS, LOGOUT.
- Section Header:** "TECNICO" in large bold letters.
- Form:** A "Nuevo" button, a text input field labeled "Tecnico", and the instruction "Introduce el nombre del tecnico a buscar".
- Search Button:** A blue "Buscar" button.
- Results Section:** Titled "Resultados", it contains a table with five rows of search results.
- Taskbar:** Shows various open applications like Grooveshar..., planSegurid..., root@Arago..., modelDB.m..., Imágenes, PAC2_EPC..., PACgeneric..., PSS = Prog..., and planss - [*/...].

Resultado	Acción
tecnico 3	Seleccionar
tecnico 3	Seleccionar
tecnico 33	Seleccionar
tecnico4	Seleccionar
tecnico5	Seleccionar

Illustration 20: Pantalla resultat tècnics

Per nou Tèctic:

Aplicaciones Lugares sáb, 26 de abr, 23:50

PSS = Programa planes de x

127.0.0.1:8080/nuevotecnicopss/

"JetBrains IDE Support" está depurando esta pestaña. Cancelar

PSS

Web Publica Personas PSS Obra Contratista Promotor Partidas PSS Generar PSS Reset PSS Estado PSS Recupera PSS LOGOUT

NUEVO TECNICO

Nom tecnico

Nif tecnico

Direccion tecnico

Telf tecnico

Fax tecnico

Email tecnico

Titulacion

Guardar

© Mateu Oliver desde 2010... | Big Disclaimer

Grooveshar... planSegurid... root@Arago... modelDB.m... Imágenes PAC2.EPC... PACgeneric... PSS = Prog... plans - [*/...

Illustration 21: Pantalla entrada de dades per tècnics

Com s'ha comentat abans les pantalles de selecció, cerca i resultat de cerca són iguals per tècnic, treballador, obra, contractista, promotor.

Les que són diferents conceptualment són les de partides:

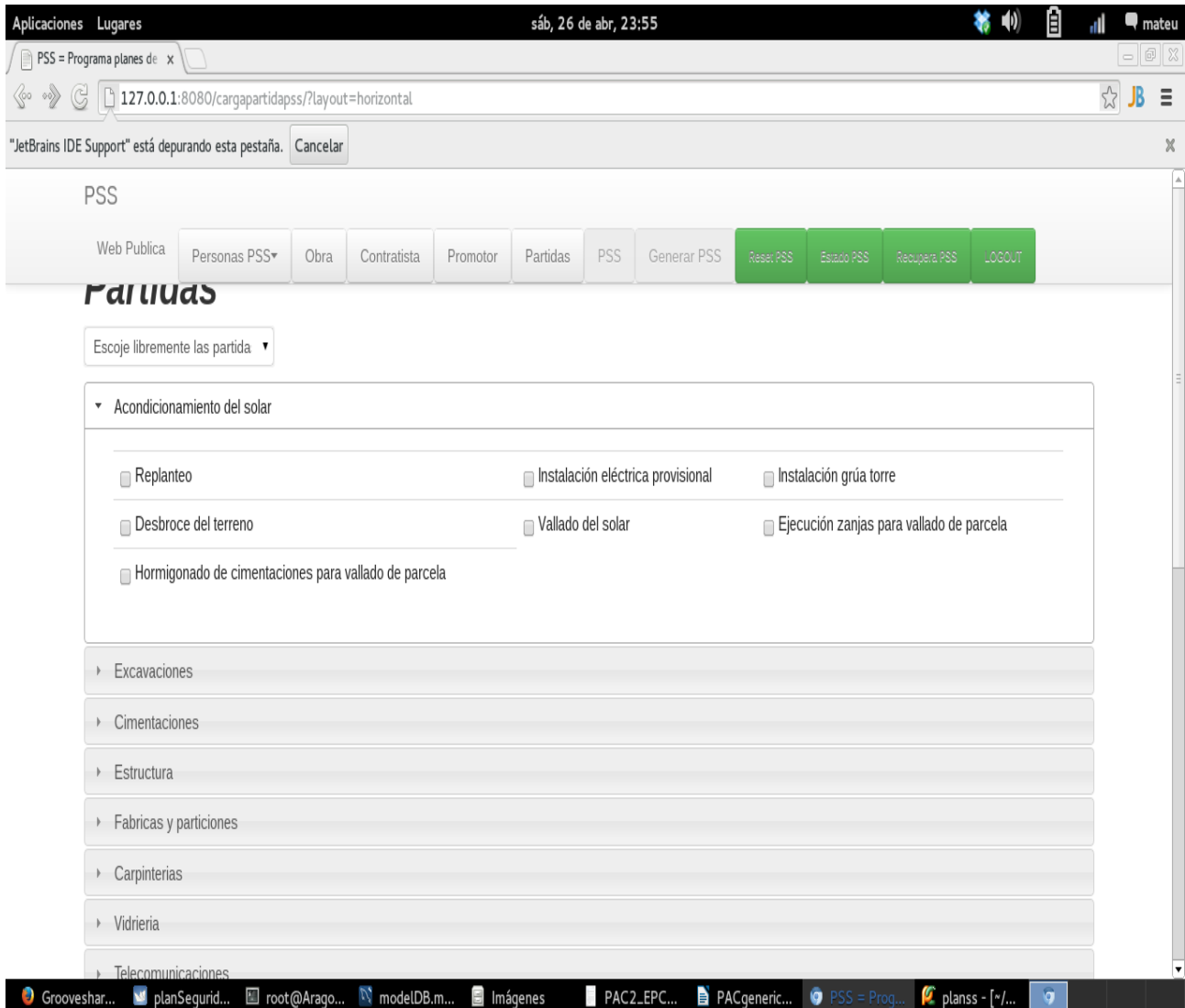


Illustration 22: Pantalla definitiva entrada partides

PSS que és la pantalla que defineix informacions relacionades amb el pla, que no són asumibles per el concepte obra, ni per el concepte promotor.

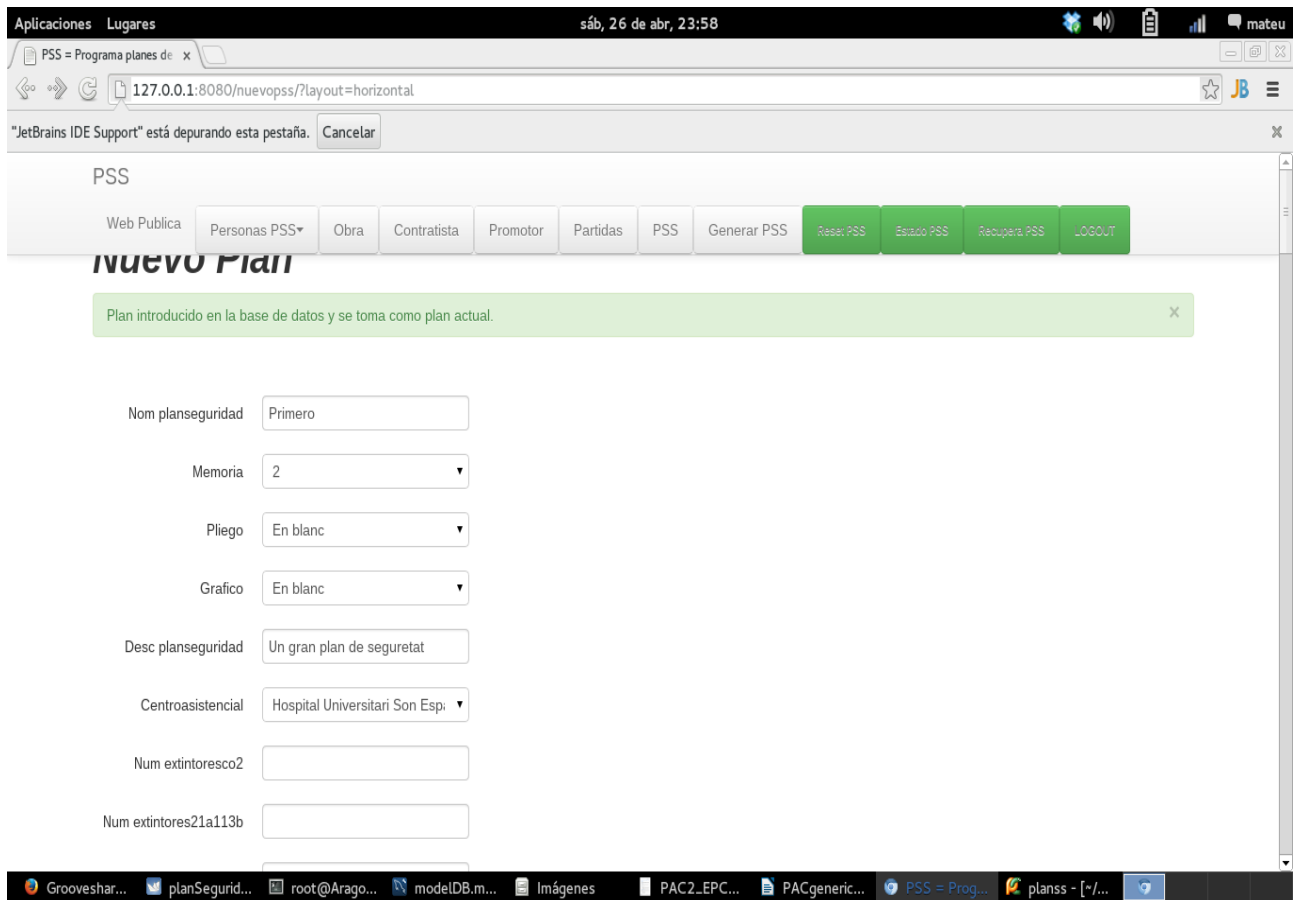


Illustration 23: Pantalla definitiva per nou pla

Existeix una ajuda per saber a quin estat es troba el pla que és estat del pla a uns dels botons verds que pot ésser cridat a qualsevol moment amb el següent resultat:

The screenshot shows a web browser window with the URL `127.0.0.1:8080/cargacontratistapss/?layout=horizontal`. The page title is "PSS" and the navigation menu includes "Web Publica", "Personas PSS", "Obra", "Contratista", "Promotor", "Partidas", "PSS", "Generar PSS", "Reset PSS", "Estado PSS", "Recupera PSS", and "LOGOUT".

The main content area is titled "Contratista Actual" and shows a search form for "contratista1". A green "Nuevo" button is visible. The search input field is labeled "Contratista" and has the placeholder text "Introduce el nombre del contratista a bu...". A blue "Buscar" button is below the input field.

A dropdown menu titled "Estatus del Plan" is open, showing a list of items:

- Tecnicos []
- trabajadores [1L]
- Obra 2
- Contratista 1
- Promotor 1
- PSS 42
- Partidas [77, 80, 81, 83, 205, 89, 95, 102, 103, 106, 117, 122, 134, 137, 139, 141, 142, 143, 152, 153, 158, 169, 170, 238, 127, 218, 194, 195, 219, 221, 239, 200, 226, 227]
- Memoria 57
- Pliego 0
- Detalles gráficos []

Below the search form, the "Resultados" section shows a message: "• No hay elementos que coincidan con la búsqueda". A pagination control shows "« 1 »".

At the bottom right, there is a copyright notice: "© Mateu Oliver desde 2010... | Big Disclaimer".

Illustration 24: Pantalla estat del pla

5. Implementació.

5.1 Eclipse amb plugin pydev.

L'eina utilitzada per el desenvolupament del projecte va ésser l'eclipse amb el plugin de python pydev.

Amb aquesta combinació s'aconseguia tenir un corrector de la sintaxi de python a més de les avantatges d'editor de text del eclipse.

Ara bé no s'arribava a la qualitat del mateix editor en el cas de java. Ja que no disposava d'un repositori de classes i l'autocompleció no funcionava.

El debugger tampoc, llavors tota la depuració s'havia de fer amb print de variables durant l'execució del codi, el que alenteix la tasca.

Es va seguir amb aquest estat fins que vaig decidir a fer un canvi

5.2 Pycharm de jetbrains.

El següent IDE que es va provar va ésser el pycharm, un ide pensat només per python i per Django.

La diferència va ésser notable, ja que la depuració línia a línia funciona, dins el propi ide existeixen totes les eines Django necessàries i l'autocompleció és molt bona.

Permeten deixar provar el IDE durant un mes i després comprar una llicència, afortunadament existeixen tot un repositori de llicències, així doncs finalment es va comprar la llicència d'estudiant per 28 euros.

5.3 Mysql.

Mysql és un sistema de gestió de bases de dades relacional multi-fil i multiusuari, que usa el llenguatge SQL (Structured Query Language).

MySQL ha esdevingut molt popular gràcies a la seva velocitat en executar consultes i el seu suport de forma nativa per part del llenguatge PHP, en l'elaboració d'aplicacions web, en l'entorn del programari lliure.

Es pot fer ús de MySQL en aplicacions de tota mena (web, d'escriptori o d'altres) de forma lliure i gratuïta sota les condicions de la llicència GPL. Si es vol integrar MySQL com a part d'un producte privatiu cal adquirir una llicència d'ús específica per a aquest propòsit.

A més de ser una base de dades bona s'ha utilitzat mysql degut a que es volia integrar amb l'entorn al núvol d'Amazon i aquest entorn en el que es refereix a django només usa aquesta base de dades. En un principi el desenvolupament s'havia fet damunt Postgresql, però gràcies a la manera en que està dissenyat Django el canvi de base de dades va ésser molt senzill. Només va caldre configurar el fitxer «settings.py» a puntant al nou motor de base de dades i migrar les dades, degut a que les dues funcionen amb SQL també va ésser molt immediat.

5.4 Bitbucket.org repositori Git remot.

Un altre programari important ha estat la utilització d'un control de versions i no perquè fos més d'un desenvolupador, si no perquè degut al caràcter del projecte part d'ell és desenvolupava al despatx on treball i altre a casa.

Llavors un sistema de control de versions remot es fa molt important, al principi usava el dropbox i el git, però vaig arribar a la conclusió que era més adient usar el bitbucket.org

El bitbucket és una aplicació web que permet descarregar els fitxers git sobre ell de tal manera es té accés a les versions des de qualsevol lloc i desenvolupant amb varies màquines sempre les tens sincronitzades.

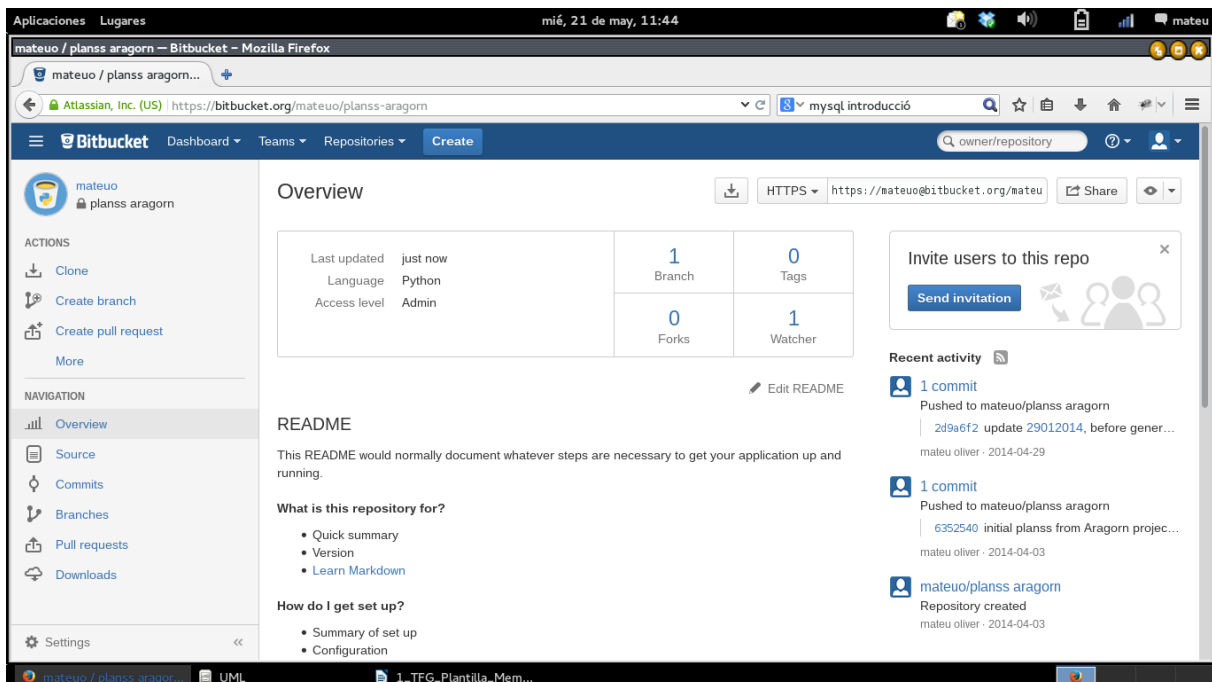


Illustration 25: Pantalla aplicació bitbucket.org

Les instruccions bàsiques de git són:

```
git init
```

Per crear un repositori nou

Per omplir el repositori amb un altre ja existent:

```
git clone /path/to/repository
```

Per omplir amb un repositori remot:

```
git clone username@host:/path/to/repository
```

Per passar els canvis fets al repositori git local:

```
git add *
```

Per validar els canvis:

```
git commit -m "Commit message"
```

Finalment per pujar al repositori remot:

```
git push origin master
```

D'aquesta manera s'aconsegueix sincronitzar els repositoris git locals de les dues màquines que he estat utilitzant.

5.5 EC2 d'Amazon.

Amazon proveïx d'infraestructura al núvol molt potent, independentment del preu que per projectes inicials com aquest és gratuïta durant un any han aconseguit distribuir el servei per productes més o menys complets.

Principalment ells et lloguen una infraestructura màquina (IaaS) el que ells coneixen com «Elastic Compute Cloud» (EC2), un emmagatzemant «Simple Service Storage» (S3) i una base de dades al núvol Relation Database Service (RDS).

Per startups aquests serveis són gratuïts amb les següents condicions:

Per EC2:

750 hores

Linux 750 hores o Windows

Per S3:

5 GB emmagatzament standard

20,000 peticions de sortida

2,000 peticions d'entrada

Per RDS :

750 hores Micro DB Instance

20 GB de DataBase Storage

20 GB per Backups, 10MM I/Os

Aquest ha estat el servei que s'ha estat provant amb resultats molt bons, hi ha que dir que el seu sistema de cobrament és molt bó, en el sentit que és molt eficient. Ja que tot d'una que te surts d'aquestes condicions et cobren , es vera que no és molt, però s'ha de controlar ja que tenen el teu número de targeta de crèdit i és molt immediat.

5.5.1 ElasticBeanstalk

Independentment de la infraestructura (IaaS) la gent d'Amazon com he comentat abans a empaquetat els seus serveis per aconseguir facilitar al desenvolupador les seves tasques, per això es disposa del servei ElasticBeanstalk que senzillament és un contenidor d'aplicacions, llavors ja no lidies amb un maquinari si no que directament amb un contenidor amb la qual cosa el manteniment i gestió de la màquina que va per davall és transparent i per tant et pots concentrar en la programació i no en el manteniment de llibreries i versions.

Elastic Beanstalk gestiona de manera automàtica els detalls d'implementació de l'aprovisionament de capacitat , equilibri de càrrega , autoescalat i gestió de l'estat de l'aplicació. Alhora , amb Elastic Beanstalk , es té el control absolut dels recursos

de AWS que potencien l'aplicació i es pot accedir als recursos subjacents quan es vulgui. Elastic Beanstalk utilitza els serveis de AWS , com Amazon Elastic Cloud Compute (Amazon EC2) , Amazon Simple Storage Service (Amazon S3) , Amazon Simple Notification Service (Amazon SNS) , Elastic Load Balancing i Auto Scaling per oferir la mateixa infraestructura d'alta fiabilitat , escalable i rendible de la qual depenen milers d'empreses avui en dia .

Per pujar l'aplicació es fa partint d'un repositori git, fet que va desencadenar que:

- Conegués una altre aplicació del mateix.
- Em motivàs a entendre-lo i a usar-lo.

A més d'usar el git s'empra un fitxer de text anomenat «requeriments.txt» on s'estableixen les versions de les llibreries que la teva aplicació usa.

Exemple del fitxer:

```
Django==1.4.1
MySQL-python==1.2.3
django-bootstrap-toolkit==2.15.0
wsgiref==0.1.2
```

5.6 Manual d'ús i Joc de proves.

5.6.1 Dinàmica d'ús de l'aplicatiu

L'aplicatiu parteix de l'axioma d'ordre d'entrada de dades, llavors no es poden emplenar els camps d'una manera aleatòria.

Per forçar aquest ordre s'ha imposat que els botons que permeten l'accés als diferents formularis s'activin i es desactivin en funció de la quantitat de informació introduïda.

Llavors l'ordre és el següent:

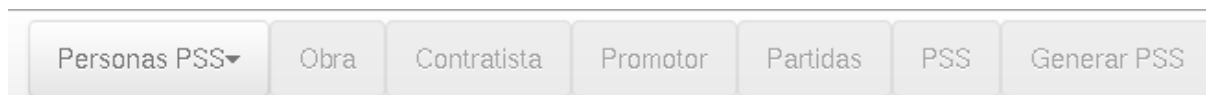


Illustration 26: Detall barra de navegació aplicació

Primer Personas que inclou els treballadors i els tècnics de l'obra, per després anar introduint els següents: Obra, Contractista, Promotor, Partides, PSS i finalment generar el PSS

5.6.2 Introducció dades necessàries per fer un pla.

Cada botó dona accés per introduir la informació de cada taula del diagrama ER, llavors es reclamen les dades necessàries per donar d'alta cada fila de la taula, per donar d'alta un treballador ens farà falta:

el nom, el nif, la direcció, el telèfon, fax, mail i titulació .

Quan a un treballador se li demana la titulació es refereix al càrrec dins l'obra:

- Recurs preventiu.
- Encarregat.

```

class Trabajador(models.Model):
    idtrabajador = models.AutoField(primary_key=True)
    nom_trabajador = models.CharField(max_length=135)
    nif_trabajador = models.CharField(max_length=9, blank=True)
    direccion_trabajador = models.CharField(max_length=135, blank=True)
    telf_trabajador = models.CharField(max_length=11, blank=True)
    fax_trabajador = models.CharField(max_length=11, blank=True)
    email_trabajador = models.EmailField(blank=True)
    titulacion = models.ForeignKey("Titulacion", blank=True)
    def __unicode__(self):
        return self.nom_trabajador
    class Meta:
        db_table = u'trabajador'
        app_label = 'planss'
    class Admin:

```

El renderitzat d'aquesta informació queda de la següent manera:

Nuevo Trabajador

Nom trabajador	<input type="text"/>
Nif trabajador	<input type="text"/>
Direccion trabajador	<input type="text"/>
Telf trabajador	<input type="text"/>
Fax trabajador	<input type="text"/>
Email trabajador	<input type="text"/>
Titulacion	<input type="text"/>

Guardar

Illustration 27: Pantalla entrada dades nou treballador

En el cas d'un tècnic:

```

class Tecnico(models.Model):
    idtecnico = models.AutoField(primary_key=True)
    nom_tecnico = models.CharField(max_length=135)
    nif_tecnico = models.CharField(max_length=9, blank=True)
    direccion_tecnico = models.CharField(max_length=135, blank=True)
    telf_tecnico = models.CharField(max_length=11, blank=True)
    fax_tecnico = models.CharField(max_length=11, blank=True)
    email_tecnico = models.EmailField(blank=True)
    titulacion = models.ForeignKey("Titulacion")

    def __unicode__(self):
        return self.nom_tecnico

class Meta:
    db_table = u'tecnico'
    app_label = 'planss'

```

i el renderitzat queda:

Nuevo Tecnico

Nom tecnico	<input type="text"/>
Nif tecnico	<input type="text"/>
Direccion tecnico	<input type="text"/>
Telf tecnico	<input type="text"/>
Fax tecnico	<input type="text"/>
Email tecnico	<input type="text"/>
Titulacion	<input type="text" value="-----"/> ▾

Guardar

Illustration 28: Pantalla entrada de dades nou tècnic

En el cas de l'obra:

Aquí s'ha de donar la informació de les característiques generals de l'obra com situació, perímetre, número de plantes per damunt rasant o per sota.

El perímetre per exemple és important ja que a l'hora de valorar proteccions col·lectives de perímetre s'utilitzarà aquest valor per fer el càlcul.


```

class Obra(models.Model):
    idobra = models.AutoField(primary_key=True)
    nom_obra = models.CharField(max_length=60)
    direccion_obra = models.CharField(max_length=135)
    tecnicos=models.ManyToManyField("Tecnico")
    perimetro = models.FloatField(null=True, blank=True)
    superficie_construida = models.FloatField(null=True, blank=True)
    num_plantas = models.IntegerField(null=True, blank=True)
    num_plantassobre = models.IntegerField(null=True, blank=True)
    num_plantasbajo = models.IntegerField(null=True, blank=True)
    presupuesto_totalobra = models.FloatField(null=True, blank=True)
    presupuesto_totalseguridad = models.FloatField(null=True, blank=True)
    duracion_meses = models.FloatField()
    porcentaje = models.FloatField(default=35.0)
    precio_hora_euro = models.FloatField(default=18.0)

    def __unicode__(self):
        return self.nom_obra
    class Meta:
        db_table = u'obra'
        app_label = 'planss'
    class Admin:
        pass

```

i el renderitzat queda

Nom obra

Direccion obra

Tecnicos

Mantenga presionado "Control", o "Command" en un Mac, para seleccionar más de una opción

Perimetro

Superficie construida

Num plantas

Num plantassobre

Num plantasbajo

Presupuesto totalobra

Presupuesto totalseguridad

Duracion meses

Porcentaje

Precio hora.euro

Illustration 29: Pantalla nova obra

En el cas del contractista s'han de definir les seves dades bàsiques com nom adreça, adreça, cif, codi postal, població etc.

```

class Contratista(models.Model):
    idcontratista = models.AutoField(primary_key=True)
    nom_contratista = models.CharField(max_length=300)
    cif_contratista = models.CharField(max_length=9, blank=True)
    dir_contratista = models.CharField(max_length=300, blank=True)
    poblacion_contratista = models.CharField(max_length=45, blank=True)
    cp_contratista = models.CharField(max_length=5, blank=True)
    telf_contratista = models.CharField(max_length=9, blank=True)
    fax_contratista = models.CharField(max_length=9, blank=True)
    email_contratista = models.EmailField(blank=True)
    web_contratista = models.URLField(blank=True)
    subcontratistas=models.ManyToManyField('Contratista', blank=True)

    def __unicode__(self):
        return self.nom_contratista

class Meta:
    db_table = u'contratista'
    app_label = 'planss'

class Admin:
    pass

```

i el renderitzat queda de la següent manera:

Nuevo Contratista

Nom contratista	<input type="text"/>
Cif contratista	<input type="text"/>
Dir contratista	<input type="text"/>
Poblacion contratista	<input type="text"/>
Cp contratista	<input type="text"/>
Telf contratista	<input type="text"/>
Fax contratista	<input type="text"/>
Email contratista	<input type="text"/>
Web contratista	<input type="text"/>

Il·lustration 30: Pantalla nou contractista

En el cas del promotor són els mateixos camps.

```

class Promotor(models.Model):
    idP = models.AutoField(primary_key=True)
    nom_promotor = models.CharField(max_length=135)
    cif_promotor = models.CharField(max_length=9, blank=True)
    direccion_promotor = models.CharField(max_length=135, blank=True)
    localidad_promotor = models.CharField(max_length=75, blank=True)
    cp_promotor = models.CharField(max_length=5, blank=True)
    telf_promotor = models.CharField(max_length=11, blank=True)
    fax_promotor = models.CharField(max_length=11, blank=True)
    email_promotor = models.EmailField(blank=True)
    web_promotor = models.URLField(blank=True)
    def __unicode__(self):
        return self.nom_promotor
    class Meta:
        db_table = u'promotor'
        app_label = 'planss'
    class Admin:
        pass

```

I el renderitzat.

Promotor

Nom promotor	<input type="text"/>
Cif promotor	<input type="text"/>
Direccion promotor	<input type="text"/>
Localidad promotor	<input type="text"/>
Cp promotor	<input type="text"/>
Telf promotor	<input type="text"/>
Fax promotor	<input type="text"/>
Email promotor	<input type="text"/>
Web promotor	<input type="text"/>

Illustration 31: Pantalla nou promotor

A l'apartat partides la pantalla no és directa si no que està més elaborada.

Primer filtrant les partides de la taula conceptes i segon agrupant les partides (conceptes) per el seu grup (capítol).

```
class Concepto(models.Model):
    UNIDAD_CHOICES = (
        (1, 'm1'),
        (2, 'm2'),
        (3, 'm3'),
        (0, 'ud')
    )
    TIPOCONCEPTO_CHOICES = (
        (4, 'Partida'),
        (3, 'EPI'),
        (2, 'Maquinaria'),
        (1, 'Medio Auxiliar')
    )
    GRUPO_CHOICES = (
        (1, 'Acondicionamiento del solar'),
        (2, 'Excavaciones'),
        (3, 'Cimentaciones'),
        (4, 'Estructura'),
        (5, 'Fabricas y particiones'),
        (6, 'Carpinterias'),
        (7, 'Vidrieria'),
        (8, 'Telecomunicaciones'),
        (9, 'Climatizacion'),
        (10, 'Electricidad'),
        (11, 'Fontaneria'),
        (20, 'Solados alicatados'),
        (21, 'Senyalizacion'),
        (22, 'Mobiliario'),
        (23, 'Jardineria'),
        (19, 'Urbanizacion')
    )
    idconcepto = models.AutoField(primary_key=True)
    tipoconcepto = models.IntegerField(choices=TIPOCONCEPTO_CHOICES)
    nom_concepto = models.CharField(max_length=135)
    desc_concepto = models.TextField()
    desc_prevent = models.TextField()
    grupo = models.ForeignKey("Grupo", blank=True, null=True)
    #grupo = models.IntegerField(choices=GRUPO_CHOICES, blank=True, null=True)
    detallegraficos = models.ManyToManyField("DetalleGrafico", blank=True)
    conceptos = models.ManyToManyField("Concepto", blank=True)
    tipo_unidad = models.IntegerField(choices=UNIDAD_CHOICES, blank=True, null=True)
    precio = models.FloatField(blank=True, null=True)
    amortizacion = models.FloatField(blank=True, null=True)
    def __unicode__(self):
        return u'%s' % self.nom_concepto
    class Meta:
        db_table = u'concepto'
        app_label = 'planss'
    class Admin:
        pass
```

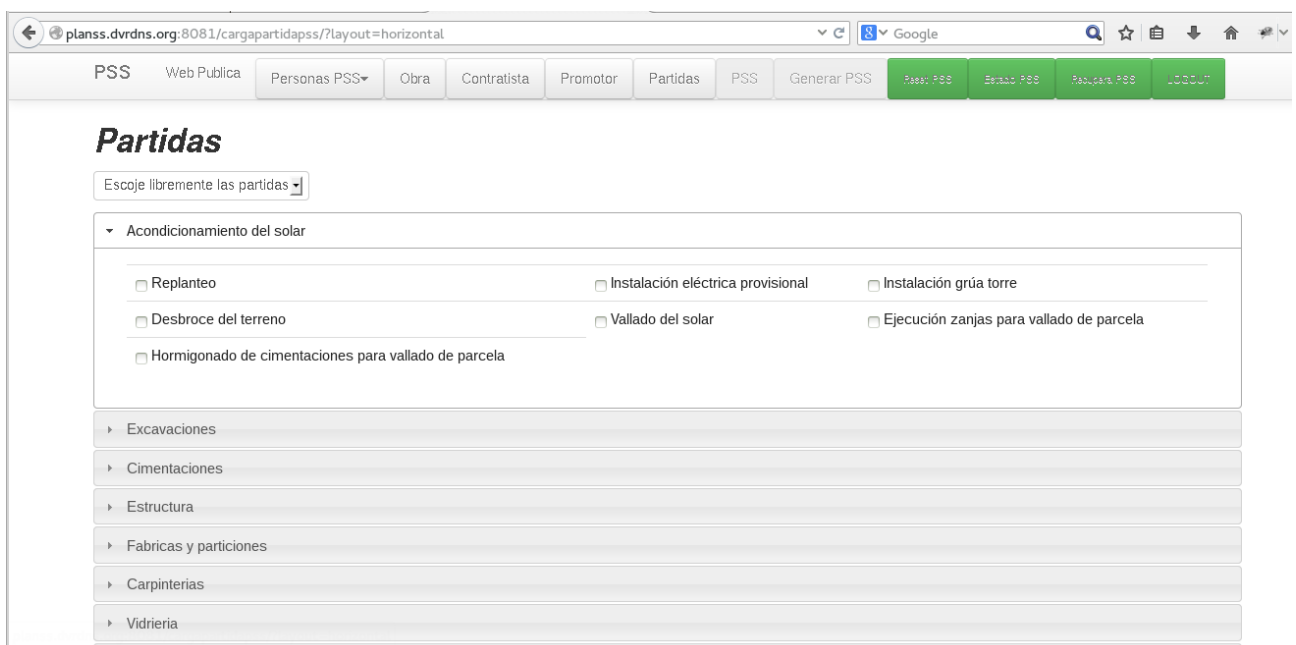


Illustration 32: Pantalla partides

Com es pot veure de cada grup (capítol) es poden seleccionar les partides que es vulguin i també es pot partir d'una preselecció si usam el desplegable i triam un predefinit. Que a la vegada també es podrà modificar a la nostra conveniència.

El apartat següent PSS defineix concretament mesures que aplicarà l'empresa contractista sobre l'obra com extintors, número de vestuaris etc.

```
class Plan(models.Model):
    CLIMA_CHOICES = (
        ('Clima Mediterraneo', 'Clima Mediterraneo'),
        ('Clima Actlantico', 'Clima Atlantico'),
        ('Clima Continental', 'Clima Continental'))
    idplan = models.AutoField(primary_key=True)
    nom_planseguridad = models.CharField(max_length=135)
    memoria = models.ForeignKey("Memoria")
    pliego = models.ForeignKey("Pliego")
    grafico = models.ForeignKey("DetalleGrafico", blank=True)
    desc_planseguridad = models.CharField(max_length=300, blank=True)
    centroasistencial = models.ForeignKey("Centroasistencial")
    num_extintoresco2 = models.IntegerField(null=True, blank=True)
    num_extintores21a113b = models.IntegerField(null=True, blank=True)
    botiquin = models.CharField(max_length=135, blank=True)
    entorno_obra = models.TextField(blank=True)
    condiciones_entorno_obra = models.TextField(blank=True)
    descripcion_actual = models.TextField(blank=True)
    lineas_aereas = models.TextField(blank=True)
    conducciones_enterradas = models.TextField(blank=True)
    estado_medianeras = models.TextField(blank=True)
    interferencias_edificios = models.TextField(blank=True)
    servidumbres_paso = models.TextField(blank=True)
    trafico = models.TextField(blank=True)
    descripcion_obra = models.TextField(blank=True)
    orografia = models.TextField(blank=True)
    lindes = models.TextField(blank=True)
```

Nom planseguridad	<input type="text"/>
Memoria	<input type="text" value="-----"/>
Pliego	<input type="text" value="-----"/>
Grafico	<input type="text" value="-----"/>
Desc planseguridad	<input type="text"/>
Centroasistencial	<input type="text" value="-----"/>
Num extintoresco2	<input type="text"/>
Num extintores21a113b	<input type="text"/>
Botiquin	<input type="text"/>
Entorno obra	<input type="text"/>

5.6.3 Generació del pla i pagament.

Una vegada s'han definit les informacions anteriors ha arribat el moment de generar el pla, fer el pagament i posteriorment descarregar-lo.

La següent pantalla il·lustra aquest fet.

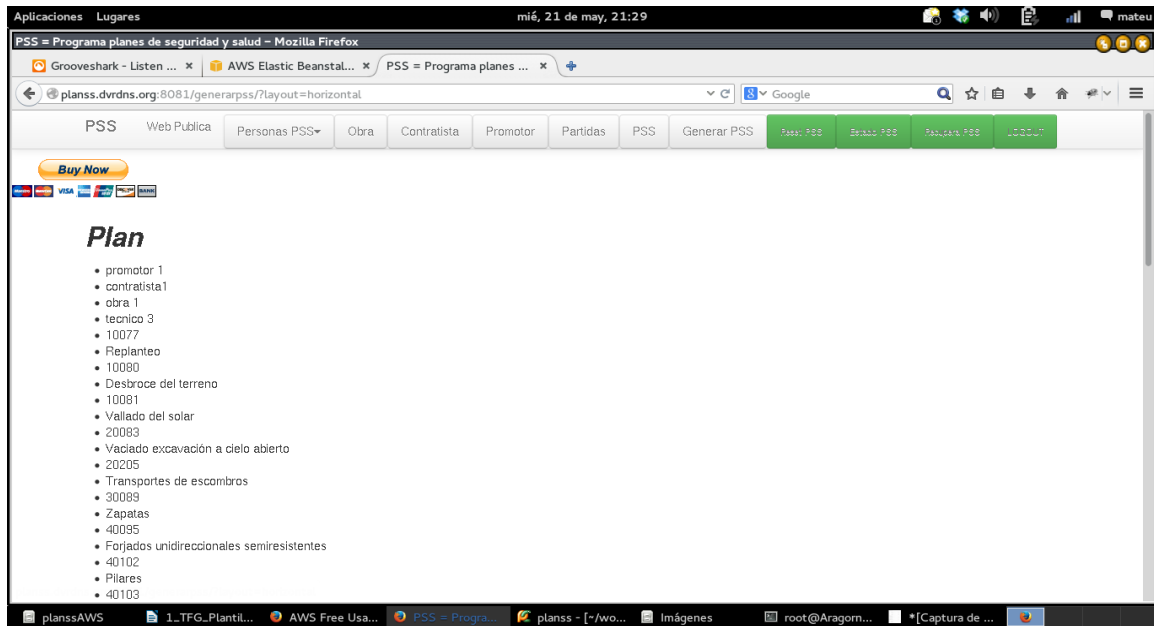


Illustration 33: Pantalla generació de pla i pagament

Com es pot veure té incrustat un botó de pagament lligat amb paypal de tal manera que en pulsar-lo ens redirigeix a la tenda per realitzar el pagament.

5.6.4 Ús de l'entorn d'administració subministrat per el framework.

El framework proveix d'un entorn d'administració d'usuaris i taules definides que és molt pràctic a l'hora de manejar la informació de les taules i actualitzar-les ja que no cal cap coneixement de SQL.

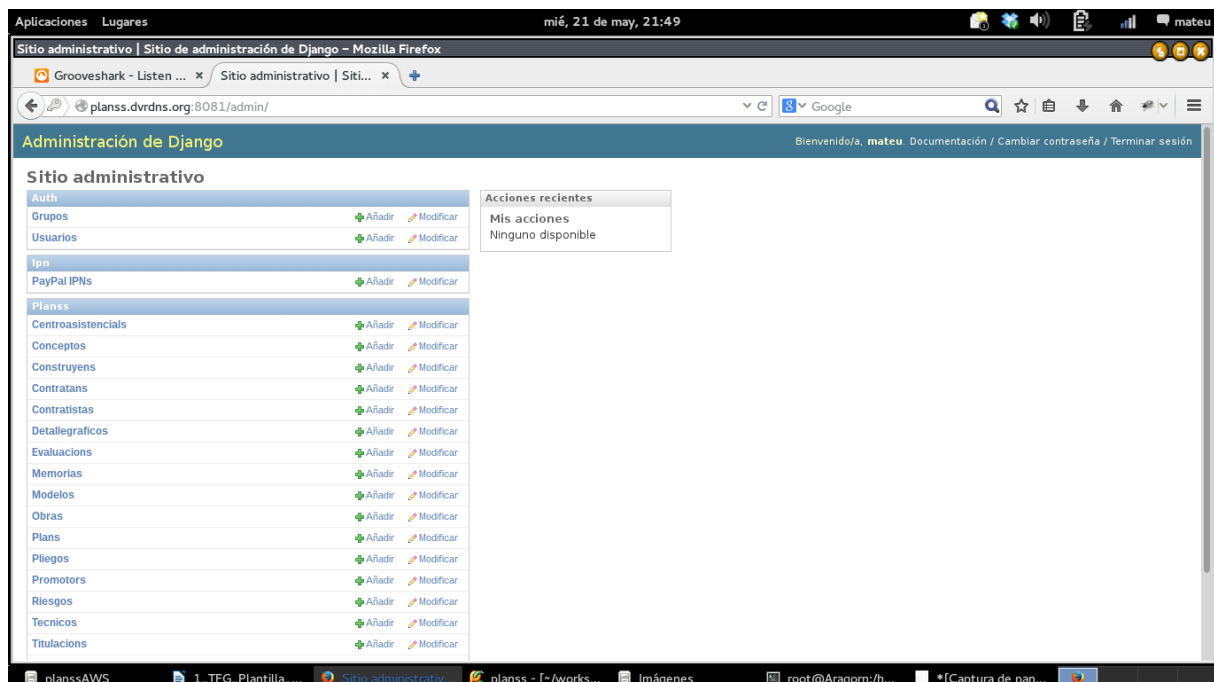


Illustration 34: Pantalles entorn d'administració del framework

Per cada taula es veu un llistat files que la conformen en el cas particular de la taula conceptes:

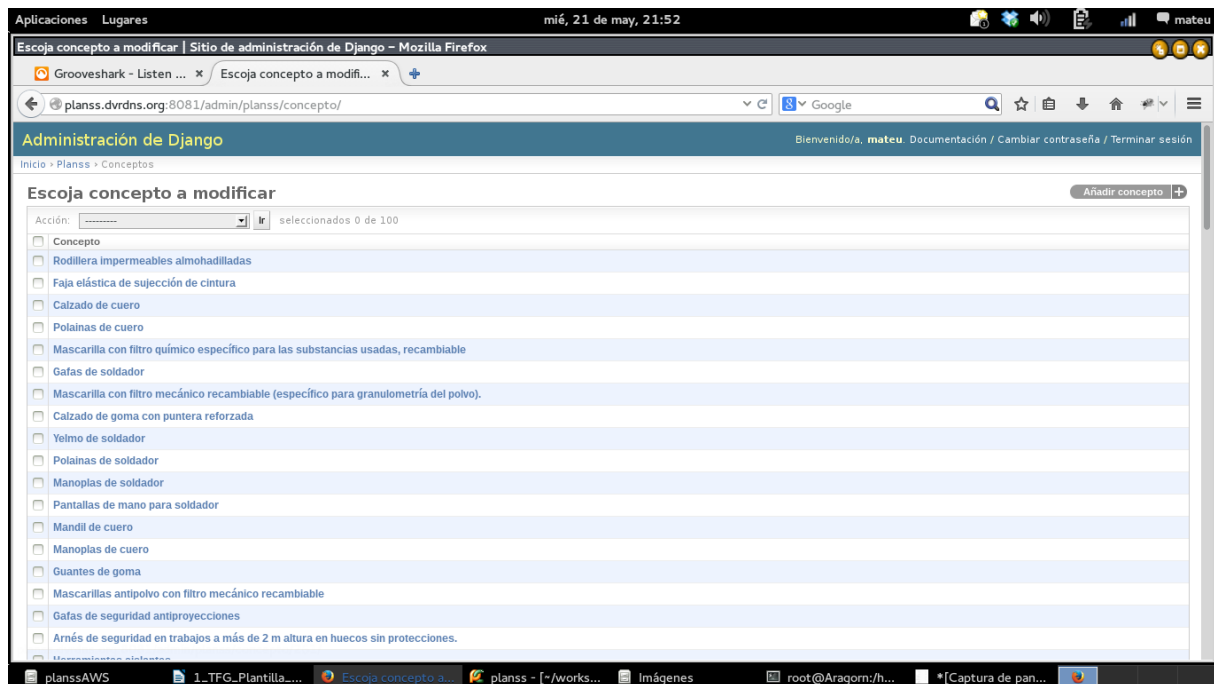


Illustration 35: Pantalla de l'administració de la taula concepte

i per cada concepte en particular:

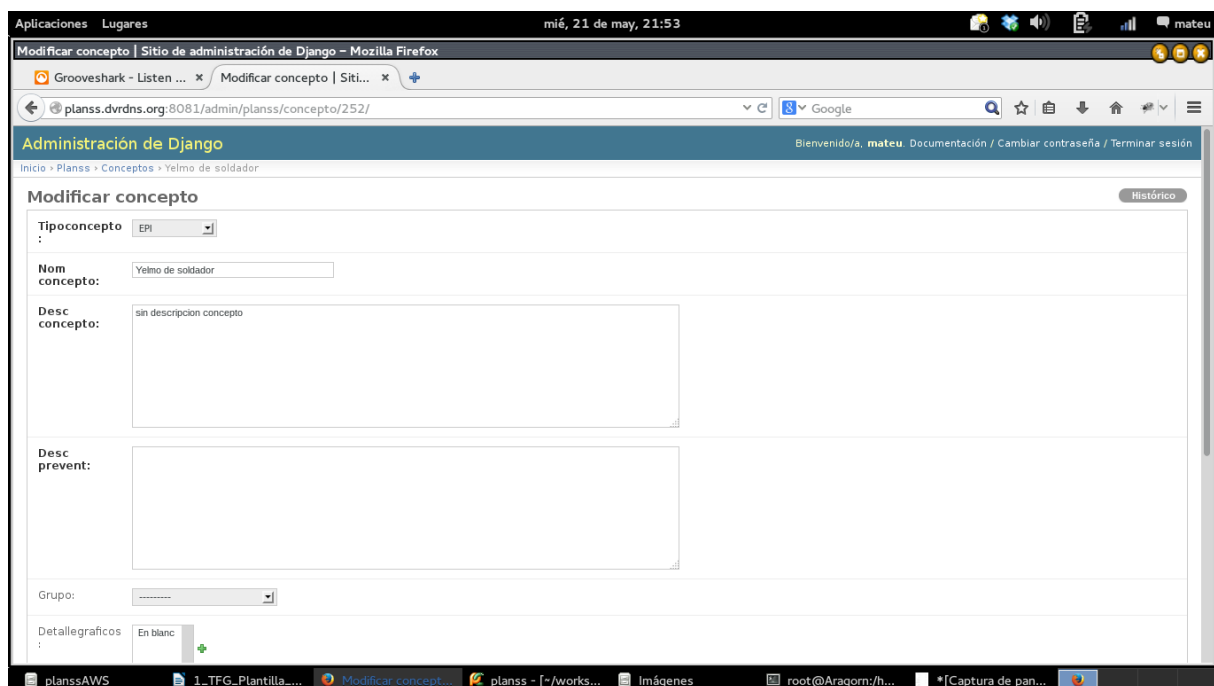


Illustration 36: Exemple de modificació de concepte

Això permet que els manteniments es realitzin d'una manera molt senzilla.

6. Estat actual del projecte.

A dia d'avui el projecte està en ús dins l'empresa, l'arquitecte les contribucions del qual han estat usades com a requisits, l'està usant actualment. És a dir és plenament funcional.

El que s'està fent per diferents tècnics de prevenció és millorar les dades que nodreixen la base de dades per aconseguir cobrir més partides i millorar les actuals.

Per la meua banda segueix fent feina per completar els punts que me queden pendents:

- Cobraments a la xarxa amb paypal.
- Instal·lació a l'estructura d'Amazon.

Els punts anteriors estan molt avançats.

A Amazon hi ha una versió anterior que ja està instal·lada, però que no és funcional, la vaig usar com a prova d'instal·lació i el pagament estic ultimant la darrera pantalla de baixada del document.

Aquest punt duu un poc de feina degut a que la llibreria per generar el document en pdf per poder-se-lo descarregar una vegada s'hagi fet el pagament no permet un mimetisme exacte amb el html.

És a dir hi ha llibreries python que permeten html → pdf, però no s'ajusten i finalment hauré de generar el fitxer del pla directament amb llibreries pdf per generar el pdf per davallar.

Llavors s'ha aconseguit l'objectiu del projecte, segons la planificació hi ha temps per solucionar aquests contratemps degut a que vaig plantejar finalitzar la programació el 16 de juliol i en l'actualitat som a 8 de juny, llavors estimo que podré tenir-ho llest segons la planificació.

7. Pressupost econòmic

El preu hora de programador està ajustat al meu salari, com si estigués contractat de plantilla, que és com realment estic encara que realitzant altres funcions que no són les de programador.

Concepte de programació	Dies	Hores	Preu/hora	Euros
Pressa de requisits	14	112	17,5	1960
Generació de continguts	12	96	17,5	1680
Casos d'ús	14	112	17,5	1960
Diagrames bases de dades	16	128	17,5	2240
Programació	30	240	17,5	4200
Llicència pycharm	-	-	-	86

Total de 12126 euros en despeses de programació

Concepte de manteniment	Euros/any
Redirecció dyndns.org	89
Elastic Beanstalk d'amazon	300

Les despeses de manteniment anual s'estimen en 389 euros

8. Glossari

Obra: Qualsevol obra pública o privada on es desenvolupen feines de construcció.

Promotor: Qualsevol persona física o jurídica per compte de la qual es realitza una obra.

Projectista: Autor o autors, per encàrrec del promotor, de la totalitat o part d'un projecte.

Coordinador de seguretat: Tècnic competent integrat dins la direcció facultativa designat per el promotor per dur a terme les tasques encomanades al article 9 del rd 1627/97.

Direcció facultativa : Tècnic o tècnics encarregats per el promotor de la direcció i control de l'execució de l'obra.

Contractista: Persona física o jurídica que assumeix contractualment davant el promotor i mitjans humans i materials propis o aliens, el compromís d'executar la totalitat o part de les obres amb subjecció al projecte i al contracte d'obra.

Estudi de seguretat: Document tècnic realitzat per tècnic competent, on s'analitzen els riscos inherents a l'obra i es proposen mesures correctores.

Pla de seguretat: Documentació específica, redactada per el contractista per donar contesta i concretar les mesures preventives avaluades al Estudi de seguretat i salut.

Capítol: El capítol d'una obra és l'agrupació de partides que tenen en comú la mateixa fase de l'obra i el mateix gremi que les executa.

Partida: Conjunt de recursos (materials, ma d'obra o maquinaria), necessaris per construir un tot indivisible que queda integrat dins una obra i que constitueix la part més petita en que es considera dividida la mateixa en un pressupost, i que per això és susceptible de ser amidada i certificada com partida d'obra.

9. Bibliografia

- Martelli, Python guía de referencia, Anaya-o'reilly, 2003, Madrid
- <http://www.djangobook.com/en/2.0/index.html> 9 de febrer de 2014
- <https://www.djangoproject.com/> 9 de febrer de 2014
- <https://www.youtube.com/playlist?list=PLxxA5z-8B2xk4szCgFmgonNcCboyNneMD> 9 de febrer de 2014
- <https://developer.paypal.com/> 10 d'abril de 2014
- http://librosweb.es/bootstrap_3/ 10 de març de 2014
- http://www.insht.es/InshtWeb/Contenidos/Normativa/TextosLegales/RD/1997/1627_97/PDFs/realdecreto16271997de24deoctubreporloqueseestablecend.pdf, 15 de febrer de 2014
- http://www.madrid.org/cs/Satellite?c=CM_InfPractica_FA&cid=1109168018558&idTema=1142598752894&language=es&pagename=ComunidadMadrid%2FEstructura&perfil=1273044216448&pid=1273078188154 01 de març de 2014