

# INDEX

<b>Index de figures .....</b>	<b>V</b>
<b>Index de taules .....</b>	<b>VII</b>
<b>1 Introducció.....</b>	<b>9</b>
1.1 CookingEat.....	9
<b>2 Objectius.....</b>	<b>11</b>
2.1 Propòsit.....	11
2.2 Finalitat.....	11
<b>3 Estudi previ.....</b>	<b>13</b>
3.1 Competència .....	13
3.1.1 <i>RECIPESARCHENGINE</i> .....	13
3.1.2 <i>RECIPE OF THE DAY</i> .....	14
3.1.3 <i>¿QUÉ COCINO HOY?</i> .....	14
3.1.4 <i>COCKTAIL FLOW</i> .....	15
3.2 Sistema operatiu Android .....	17
3.3 Drupal .....	19
3.3.1 <i>Què és?</i> .....	19
3.3.2 <i>Com funciona?</i> .....	21
3.3.3 <i>Mòduls instal·lats</i> .....	22
3.4 Eines i materials.....	29
3.4.1 <i>Eclipse</i> .....	29
3.4.2 <i>ADT</i> .....	29

3.4.3	<i>Eines de disseny (Photoshop)</i> .....	29
3.5	Arquitectura client-servidor asíncrona .....	30
3.5.1	<i>Rest (J-SON)</i> .....	30
3.5.2	<i>XML</i> .....	30
<b>4</b>	<b>Definició de l'aplicatiu</b> .....	<b>33</b>
4.1	Mock-up.....	33
4.2	Casos d'ús.....	34
4.2.1	<i>Llista de plats</i> .....	35
4.2.2	<i>Llista de postres</i> .....	35
4.2.3	<i>Lista de salses</i> .....	35
4.2.4	<i>Lista de begudes</i> .....	36
4.3	Llista d'usuaris .....	36
4.4	Model vista-controlador.....	38
4.4.1	<i>Què es què en aquest projecte</i> .....	39
4.5	Diagrama de seqüència.....	39
4.5.1	<i>Llistar receptes</i> .....	39
4.5.2	<i>Llistar usuaris</i> .....	40
4.5.3	<i>Mostrar recepta</i> .....	41
<b>5</b>	<b>Desenvolupament de l'aplicació</b> .....	<b>43</b>
5.1	Disseny de l'aplicació.....	43
5.2	Codificació de l'aplicació.....	45
5.3	Android Manifest.....	45

5.4	Pantalles de l'aplicació .....	49
5.4.1	<i>Main Activity</i> .....	49
5.4.2	<i>Gestió de dades</i> .....	51
5.4.3	<i>Botó Lista de usuarios</i> .....	57
5.4.4	<i>Llista de receptes d'usuari</i> .....	60
5.4.5	<i>Mostrar recepta</i> .....	62
5.5	Altres classes .....	66
5.5.1	<i>SharedValues.java</i> .....	66
<b>6</b>	<b>Creació del arxiu APK .....</b>	<b>67</b>
6.1	Com generar l'arxiu APK i el Keystore .....	67
<b>7</b>	<b>Testing .....</b>	<b>71</b>
<b>8</b>	<b>Problemes trobats.....</b>	<b>73</b>
<b>9</b>	<b>Estudi econòmic.....</b>	<b>75</b>
9.1	Materials i amortitzacions.....	75
9.2	Hores d'enginyeria.....	76
9.3	Hores de desenvolupament .....	77
9.4	Pressupost total .....	77
<b>10</b>	<b>Ampliacions .....</b>	<b>79</b>
<b>11</b>	<b>Conclusions.....</b>	<b>81</b>
<b>12</b>	<b>Bibliografia .....</b>	<b>83</b>



## Index de figures

Il·lustració 1. Aplicació RecipeSearchEngine.....	13
Il·lustració 2. Aplicació Recipe of the day.....	14
Il·lustració 3. Aplicació ¿Qué cocino hoy?.....	14
Il·lustració 4. Aplicació Cocktail Flow .....	15
Il·lustració 5. Framework de Drupal .....	19
Il·lustració 7. Pantalla de configuració de Views.....	24
Il·lustració 8. Afegir una nova vista.....	24
Il·lustració 9. Edició d'una vista en Drupal .....	25
Il·lustració 10. Escollir estil de la vista .....	26
Il·lustració 11. Formatar el JSON .....	27
Il·lustració 12. Escollir els camps de la vista .....	28
Il·lustració 13. Mock-Up de l'aplicació .....	33
Il·lustració 14. Casos d'ús de l'aplicació .....	34
Il·lustració 15. Esquema de funcionament del patró MVC.....	38
Il·lustració 16. Diagrama de seqüència de llistar receptes .....	40
Il·lustració 17. Diagrama de seqüència de llistar usuaris.....	40
Il·lustració 18. Diagrama de seqüència de mostrar recepta.....	41
Il·lustració 19. Pantalla d'inici de creació d'arxiu APK .....	68

Il·lustració 20. Selecció del nom de l'aplicació .....	68
Il·lustració 21. Posició on es guardarà la keynote.....	69
Il·lustració 22. Creació de la key.....	70
Il·lustració 23. Lloc on es guardarà l'arxiu .APK .....	70

## **Index de taules**

Taula 1. Comparativa d'aplicacions de cuina.....	17
Taula 2. Materials i amortitzacions.....	75
Taula 3. Hores d'enginyeria.....	76
Taula 4. Hores de desenvolupament.....	77
Tabla 5. Pressupost total.....	77





# 1 Introducció

L'evolució de la tecnologia en els darrers anys ha crescut de manera exponencial desde l'aparició dels ordinadors de sobretaula, però aquesta evolució encara ha sigut més ràpida amb l'aparició dels primers smartphones ara fa uns anys. Fa deu anys, ningú creia que podríem arribar a tenir els ordinadors al palmell de la nostra mà; hem passat de trucar i poca cosa més a tenir el smartphone en el nostre dia a dia, podent des de navegar per internet, a entrar a les xarxes socials, o jugar online al poker entre milers de persones de tot el món.

Per altra banda, no només la tecnologia és la que avança, sinó que el món d'Internet ha evolucionat des del punt de buscar una informació a una web a estar tots connectats amb la creació de les xarxes social i webs col·laboratives. Donats aquests dos punts, és normal que cada cop estiguem més interconnectats entre nosaltres canviant els sistemes web fins ara.

## 1.1 CookingEat

CookingEat és una xarxa social de cuiners que recull una gran quantitat de receptes de cuina de tot el món i el perfil d'usuari de tots els cuiners que es registren.

La missió del projecte és crear una comunitat de cuiners tant professionals com amateurs que vulguin compartir les seves receptes i crear contactes gastronòmics.

L'objectiu d'aquesta xarxa és aconseguir molts usuaris i, sobretot, el seu feedback per a poder anar millorant la plataforma dia a dia.

Per tant, estarà dirigit tant a cuiners professionals d'Espanya que vulguin donar-se a conèixer i fer nous contactes com a usuaris que tenen com a hobby cuinar i volen innovar en els seus plats o ensenyar a la comunitat què és el que els agrada cuinar i de quina manera ho fan, ja que cada persona pot fer el mateix plat de diverses maneres. Més endavant es vol expandir territorialment el projecte i que també sigui present a països llatinoamericans com per exemple Mèxic o Argentina, ja sigui afegint receptes d'aquestes regions o animant a cuiners d'aquestes nacionalitats a participar-hi.

La competència que hi ha actualment en aquest camp és baixa ja que és un projecte molt innovador i les pàgines web que tracten aquest tema ho fan sense tenir en compte l'experiència d'usuari, el disseny, la usabilitat i sobretot, el fet de fer servir molta publicitat acabant per donant-li més importància que al contingut de les receptes.

Per altra banda, al mercat americà està tenint molta repercussió aquest tipus de xarxa social relacionades amb la cuina i, per tant, és un exemple a seguir per aplicar al nostre cas.

Una de les estratègies d'aquest projecte per a seguir creixent és la creació d'una aplicació mòbil des d'on els usuaris podran consultar les receptes i serà el tema principal d'aquest projecte final de carrera.

## 2 Objectius

- Creació d'una aplicació mòbil funcional que serveixi per la visualització de la pàgina web de [cookingeat.com](http://cookingeat.com) de forma simple i efectiva.
- Facilitar l'ús de l'aplicació en mòbil en les mateixes funcions que tingui la web.
- Utilitzar la tecnologia Android per les funcionalitats que ofereix, aprendre la seva estructura, components i mètodes de programació que té.
- Utilitzar el CMS Drupal, eina amb la qual ha estat creada la web.
- Aprendre les funcionalitats del protocol REST (J-SON) per poder connectar l'aplicació amb la web.

### 2.1 Propòsit

Aquest projecte de final de carrera consisteix en la creació d'una aplicació en Android de la pàgina web [www.cookingeat.com](http://www.cookingeat.com). Amb aquesta aplicació es podrà fer la visualització de les receptes de la web d'una forma fàcil, senzilla i intuïtiva.

Amb la creació d'aquesta aplicació per Android s'espera, a més a més, que la pàgina web aconseguixi més usuaris, i molt feedback per poder anar millorant dia a dia tant la pàgina web com l'aplicació.

La idea de desenvolupar aquesta aplicació per mòbil ve donada pel fet que, ara mateix, el creixement de les connexions a Internet mòbils creix a un ritme exponencial, la gent utilitza molt més les tablets i smartphones pel dia a dia, cada cop hi ha una línia més definida entre oci amb els smartphones i tablets, i treball amb els ordinadors. Molta gent ha trobat una comoditat el fet de poder veure des de qualsevol lloc la seva pàgina web de notícies (des de el sofà, el bar o de vacances) sense haver de dur l'ordinador portàtil a sobre, sense la preocupació de tenir o no Internet a la zona, o sense haver d'anar on l'usuari tingui l'ordinador de sobretaula.

### 2.2 Finalitat

La utilització de les tecnologies mòbils ha crescut en tots els àmbits de la vida quotidiana de persones de tot el món, des de nens fins a adults, això unit als markets i la gran quantitat

d'aplicacions que surten diàriament, ha fet d'aquest mercat dels més rentables actualment. A més, és una bona forma de promoció de webs que estan associades a aplicacions amb èxit.

Els motius de crear aquesta aplicació són els següents:

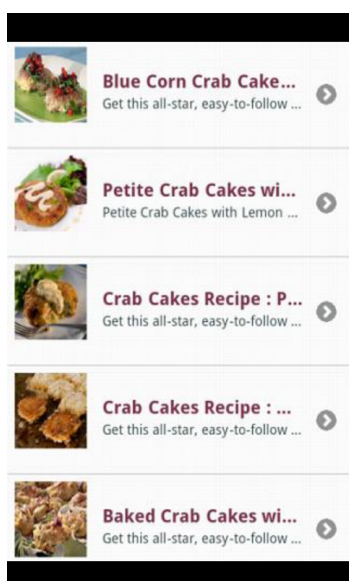
- La possibilitat de veure la web de forma àgil i simple, sense haver d'ampliar i reduir a causa de no poder veure la web sencera, perquè es veu massa petita o per incompatibilitats.
- Donar-li noves funcions que essent només web no podria tenir, aprofitant funcions pròpies dels dispositius, com la geolocalització o el NFC.
- Aprofitar el tràfic de gent que mouen els markets d'aplicacions per aconseguir més tràfic per la pàgina web.

## 3 Estudi previ

### 3.1 Competència

Abans de començar a desenvolupar el programari, es farà un estudi de mercat sobre les aplicacions més interessants en l'àmbit de la cuina que hi ha al market de Google Play actualment.

#### 3.1.1 RECIPESEARCHENGINE



Aquesta aplicació és un buscador que busca les receptes a diferents webs de la seva pròpia base de dades.

L'aplicació simplement té una barra de cerca en la que s'escriu un ingredient de la recepta buscada, i l'app fa un llistat amb les diferents receptes que ha trobat de les diferents pàgines webs. Un cop trobada la recepta interesada, tocant la recepta seleccionada, s'obra una altra pantalla des de la que es pot anar a la web en qüestió o compartir el link.

**Il·lustració 1. Aplicació  
RecipeSearchEngine**

### 3.1.2 RECIPE OF THE DAY



Il·lustració 2. Aplicació Recipe of the day

Aquesta aplicació és molt simple, mostra una foto del plat del dia, els ingredients i la preparació d'aquest, i amb els botons de older and newer es poden veure les receptes anteriors i posteriors al dia.

### 3.1.3 ¿QUÉ COCINO HOY?



Il·lustració 3. Aplicació ¿Qué cocino hoy?

¿Qué cocino hoy? és una aplicació que té una interfície gràfica molt cuidada, amb més de tres mil receptes de tot tipus.

A la cantonada esquerra, es troben les compres que es poden fer de l'aplicació, treure publicitat i comprar veus diferents per escoltar les receptes, i a la dreta un botó que fa la funció de random, fent sortir receptes de forma aleatòria.

A part de la barra de cerca que tenen la majoria d'aplicacions d'aquest estil, té dues formes de visualització, en format llistat o per icones. La part de llistat posa el títol de la recepta i una petita foto del plat. En el cas d'escollir la visualització d'icones, es veu el plat de forma més gran per poder escollir més fàcilment el plat que més interessa.

Quan s'entra a qualsevol plat, es troba a la part de dalt, l'icona de tornar i el botó per afegir a favorits. Després es mostra la fotografia del plat, i seguidament els ingredients on es pot

indicar per quantes persones és el plat i l'elaboració. Tant en els ingredients com en l'elaboració hi ha una icona d'altaveu, que permet sentir la recepta.

Tornant al menú principal, a la part de sota es troben varies seccions:

- **Título:** És el primer que es veu a l'aplicació explicat a la part anterior.
- **Ingredientes:** A partir dels ingredients que més poden interessar, selecciona els plats que els tinguin.
- **Grupos:** Es fa una selecció a partir de diferents grups de receptes.
- **Favoritos:** Es pot veure les receptes que més han agradat.

### 3.1.4 COCKTAIL FLOW

Aplicació que encara que no estigui relacionada amb les receptes culinàries, per la part de menjar, la temàtica és la mateixa que pot tenir una aplicació de cuina.



El model de negoci d'aquesta aplicació està basada en la compra de paquets de còctels que no estan en la versió bàsica, podent comprar paquets per diferents tipus de festes i events per \$0.99.

Les categories trobades en aquesta app són les següents:

**Sorpresa:** Posa un còctel a l'atzar, amb el seu nom i els ingredients necessaris per fer-lo, i es pot mirar la seva preparació i còctels similars.

Il·lustració 4. Aplicació Cocktail Flow

**Categorias:** Aquesta és la secció més extensa de l'app, es poden veure els còctels en:

- Todos los cócteles: Es podran veure tots els còctels per ordre alfabètic, si s'entra es veuran els ingredients i la seva preparació.
- Según la bebida principal: Es poden escollir els còctels en funció d'un ingredient base.

- Según el color: Es poden escollir els còctels basats en el seu color.
- Según el tipo: Es poden escollir el còctels segons el seu tipus. Els tipus són els següents: classical, creamy, long drink, shooter i tropical.

**Paquetes:** Aquesta secció té dos punts importants que són Premium i Free.

- **Premium:** Aquesta part, de pagament, es poden comprar receptes de còctels basats en diferents tipus d'events i festivitats. Cada paquet té un preu de \$0.99.
- **Free:** Aquesta és la secció gratuïta, es poden aconseguir dos paquets a part del paquet gratuït bàsic, Bebidas cortas i Sin Alcohol.
- **Compras:** Secció en la qual, a partir d'uns ingredients específics donats per la pròpia aplicació, es poden veure els preus aproximats dels ingredients dels còctels que vulgui preparar amb les seves recomanacions.
- **Inventario:** A partir dels ingredients que s'introdueixen, l'app indica si es pot fer algun còctel.
- **Favoritos:** Aquí l'app guarda les receptes favorites de l'usuari.



La següent taula és una comparativa on es veurà quines són les principals funcions de cadascuna de les aplicacions escollides i quin serà el punt diferencial de CookingEat vers la competència.

	RecipeSearchEngine	Recipe of the day	¿Qué cocino hoy?	Cocktail Flow	CookingEat
Web	X			X	X
Registre d'usuaris					X
Afegir recepta	X	X	X	X	X
Vidreceptes					X
Buscar receptes	X	X	X	X	X
Interface			X		X
Lectura per veu de receptes			X		X
Xarxa social					X
Preu assequible		X			X

Taula 1. Comparativa d'aplicacions de cuina.

A la comparativa es veu clarament quins són els punts diferencials de l'app de CookingEat, encara que en aquesta primera versió només es desenvoluparà la visualització de receptes, té un gran ventall d'oportunitats que s'aniran explotant en diferents etapes del desenvolupament.

## 3.2 Sistema operatiu Android

Android és un sistema operatiu basat en Linux dissenyat principalment per a dispositius mòbils amb pantalla tàctil, com telèfons intel·ligents i Tablet PC.

Desenvolupat inicialment per Android Inc, que Google va comprar el suport financer i més tard, el 2005, Android es va donar a conèixer el 2007 amb la fundació de l'Open Handset Alliance: un consorci de maquinari, programari, telecomunicacions i empreses dedicades a

la promoció oberta estàndard per a dispositius mòbils. El primer telèfon amb Android va ser venut a l'octubre de 2008.

Android és de codi obert i Google té el codi alliberat sota la llicència Apache. Aquest codi font obert i de llicència permissiva permet que el programari pugui ser modificat i distribuït lliurement pels fabricants de dispositius, operadors mòbils i desenvolupadors entusiastes. A més, Android té una gran comunitat de desenvolupadors que escriuen aplicacions ("apps") que amplien la funcionalitat dels dispositius, escrit principalment en una versió personalitzada del llenguatge de programació Java.

A l'octubre de 2012, hi havia aproximadament 700.000 aplicacions disponibles per a Android, i el nombre estimat de les aplicacions descarregades de Google Play, botiga d'aplicacions d'Android primària, va ser de 25 milions de dòlars.

Aquests factors han permès a Android convertir-se en la plataforma més utilitzada pels telèfon intel·ligents i el programari d'elecció per les empreses de tecnologia que requereixen d'un sistema operatiu lleuger per a dispositius d'alta tecnologia sense desenvolupar-ne un de nou.

Com a resultat, tot i estar dissenyat principalment per a telèfons i tablets, s'han vist aplicacions addicionals en els televisors, consòles de jocs i altres aparells electrònics. La naturalesa oberta d'Android ha encoratjat a una gran comunitat de desenvolupadors i entusiastes d'utilitzar el codi com una base per impulsar projectes comunitaris, que afegeixen noves característiques per a usuaris avançats o portar Android a altre dispositius que no tenen actualitzacions oficials.

Android tenia una quota de mercat mundial de telèfons intel·ligents del 75% durant el tercer trimestre del 2012, amb 500 milions de dispositius activats en activacions totals i 1,3 milions per dia. No obstant això, l'èxit del sistema operatiu s'ha convertit en un objectiu per litigis sobre patents i guerres entre empreses tecnològiques.

Google Play és el principal mercat de venda i distribució d'aplicacions d'Android. En un mercat obert, Google Play et permet tenir el control de com es venen els seus productes. Es

pot publicar sempre que es vulgui, tantes vegades com es vulgui. Es pot distribuir àmpliament a tots els mercats i dispositius o en segments específics.

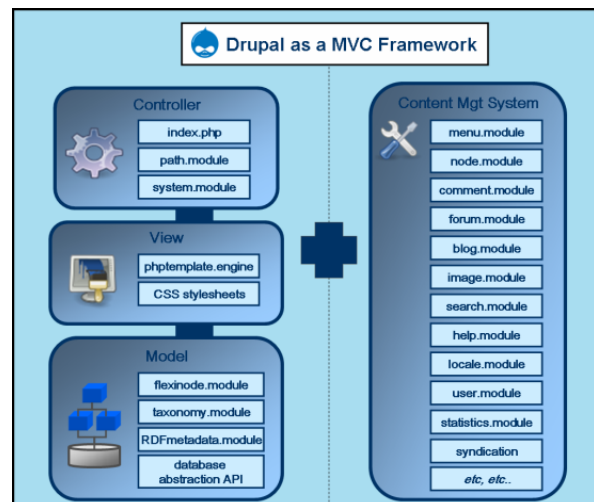
Es pot obtenir beneficis econòmics de la manera que millor s'adapti al negoci. També es té el control total de la fixació de preus per les seves aplicacions i productes integrats en l'aplicació i es pot establir o canviar els preus en qualsevol moneda suportada en qualsevol moment.

### 3.3 Drupal

#### 3.3.1 Què és?

Drupal és un CMS (Content Management System) lliure escrit en PHP i es distribueix sota llicència GNU.

És el back-end del 2.1% de les webs que hi ha actualment a internet, des de blogs personals a webs corporatives, polítiques, o governamentals i ha augmentat el seu ús un 125% en els últims anys.



Il·lustració 5. Framework de Drupal

El core del drupal conté les funcions bàsiques d'un gestor de continguts: registre i manteniment de comptes d'usuari, administració de menús, feeds de RSS, layouts i back-end.

Per altra banda, hi ha una infinitat de mòduls, o extensions, que poden transformar el drupal en qualsevol tipus de web, ja sigui un blog, un e-commerce, un fòrum, una xarxa social, etc.

El contingut a Drupal s'organitza en nodes individuals. Als nodes del tipus "història" els usuaris hi poden afegir comentaris (els comentaris no estan considerats nodes). Depenent de la configuració de la pàgina web afegir nous nodes i/o crear nous comentaris pot estar

permès o no als usuaris. A més, tant els nodes com els comentaris poden necessitar l'aprovació per part dels moderadors abans que el node o el comentari es mostrin a tothom. Les entrades d'un bloc són un altre tipus de node.

L'esquema per defecte de Drupal consisteix en tres columnes. La columna central s'anomena també la columna del contingut. Normalment mostra una llista, amb títol i un resum, dels darrers missatges enviats ordenats per data, mostrant-se els més nous a la part de dalt. Si es fa clic al títol d'un dels nodes es pot llegir el contingut sencer del node a la columna central en una nova pàgina.

Les columnes de la dreta i l'esquerra són les barres laterals ('sidebars'). Les barres laterals poden mostrar blocs (de l'anglès 'blocks', no confondre amb els 'blogs' que en català també se'n poden dir blocs). Aquests blocs solen mostrar enllaços de navegació per accedir a d'altre contingut de la web, per exemple els darrers comentaris enviats o els articles més populars (més llegits). En una instal·lació nova els blocs que es mostren el bloc d'entrada (anomenat 'usuaris registrats') i un bloc de navegació amb un menú amb les accions disponibles. En aquest bloc de navegació es poden mostrar diferents opcions depenent de les accions que estiguen fent en cada moment i dels permisos que puguen tenir.

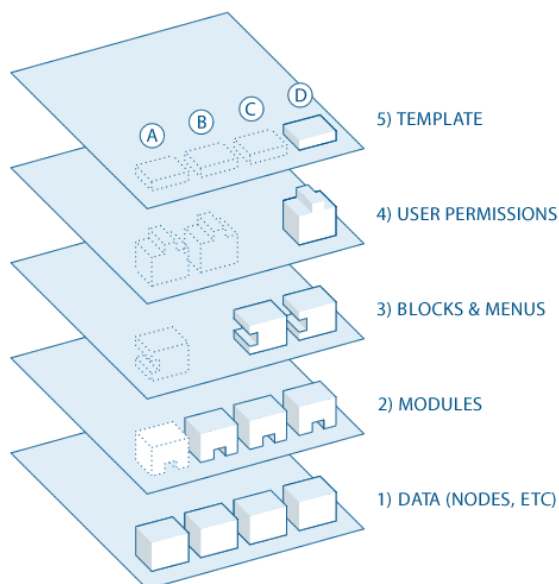
Els blocs es mostren o no en funció del que s'estigui fent en cada moment i dels permisos que es tingui. Per exemple, el bloc per entrar (login block) no es mostrarà si ja s'ha entrat o el bloc d'enviaments recents pot no mostrar-se si no hi ha nodes disponibles. L'administrador pot habilitar o deshabilitar els blocs al menú Administra – Blocs.

Els nodes es poden organitzar en categories, també anomenades taxonomies. Els fòrums són un exemple de nodes organitzats per categories. Les categories es poden organitzar en una jerarquia múltiple, amb categories 'pare' i categories 'fill'.

Els mòduls afegeixen noves funcionalitats al web. Un cop afegit l'arxiu mòdul al subdirectori 'modules' apareix una nova opció per activar-lo al menú Administra - Mòduls. Un cop activat, les noves funcionalitats que aporta estan disponibles de manera automàtica. Els mòduls poden definir nous tipus de node, nous elements de menú i nous blocs per mostrar als costats.

### 3.3.2 Com funciona?

El drupal està compost per cinc capes diferents i totalment independents, fent així que sigui totalment personalitzable i adaptable per a cada necessitat.



Il·lustració 6. Composició de capes del Drupal

Per altra banda, els menús són els enllaços principals i secundaris de la web.

**Permisos:** Definició dels diferents rols de permisos per cada grup d'usuari.

**Template:** L'estil gràfic del drupal.

Aquestes cinc capes poden ser personalitzades des del panell d'administració del drupal, on es decidirà quina serà la finalitat de la web, com serà, com funcionarà i quina experiència d'usuari oferirà:

**Estructura:** L'estructura de drupal es divideix en diferents 'peces' que encaixen unes amb altres, com per exemple, els blocs, els menús, els perfils d'usuari i la navegació.

**Site building:** En aquesta part és on es decideix quin tipus de web serà (blog, e-commerce, xarxa social, workflow, landing page...). A més, tot això es construeix gràcies a vistes, taxonomies, informació geogràfica, workflow, formularis, etc.

**Multilingual:** Es pot realitzar qualsevol tipus de web en multiidioma sense cap limitació.

**Informació:** Contingut que es publicarà a la web, ja siguin articles, entrades de blog, pàgines, enllaços, etc...

**Mòduls:** Funcionalitats bàsiques o d'extensió que es puguin afegir al CMS.

**Blocs i menús:** Els blocs són els diferents apartats de la web que romanen al voltant de la web. Normalment és informació estàtica i sempre visible sense dependre de la secció de la web que s'estigui visitant.

**Theme:** Customització de l'interface mitjançant CSS, HTML, etc. Pot estar afectat per algun mòdul.

**Mobile:** Fàcil adaptació de la web a dispositius mòbils.

### 3.3.3 Mòduls instal·lats

Per a poder accedir a la informació necessària de la pàgina web només s'ha necessitat de la instal·lació del mòdul Views DataSource i el Views com a pre-requisit per a poder-lo fer funcionar.

#### 3.3.3.1 Views datasource

##### Que fa?

View Datasource és un paquet de plugins per el Drupal Views que transforma continguts en formats basats en XML, JSON i XHTML. Aquest formats permeten afegir continguts a la web Drupal per ser fàcilment usats com a fonts d'informació per la web semàntica. Els plugins del Views Datasource permeten veure els continguts que t'interessen desde la llista de nodes creada amb la interfície del Drupal Views en una gran varietat de formats; documents XML fent servir esquemes com OPML, documents RDF fent servir diferents vocabularis com, per exemple, el FOAF, contingut JSON en formats com el Exhibit JSON i documents XHTML fent servir microformats com el hCard.

Views Datasource està pensat com un complement per altres mòduls com el Services i RDF per si en algun moment interessa mostrar la informació en diferents formats.

Aquest mòdul te 4 vistes diferents:

1. Views\_xml. Permet vistes en XML, OPML i Atom.
2. Views\_json. Permet vistes en simile/Exhibit JSON, Canonical JSON, JSONP/JSON
3. Views\_rdf. Permet vistes en FOAF, SIOC i DOAP
4. Views\_xhtml. Permet vistes en hCard, hCalendar i Geo.

D'aquestes 4 vistes la que ens interessa és el views\_json.

### 3.3.3.2 Instal·lació del mòdul

La instal·lació és molt senzilla. Per fer la instal·lació seguirem els següents passos:

1. Descarregar de la pagina web [www.drupalmodules.org](http://www.drupalmodules.org). El mòdul ha de ser compatible amb la versió de Drupal que es faci servir. En aquest cas, es la versió Drupal 6.
2. Descomprimir l'arxiu.
3. Pujar l'arxiu. Utilitzant algun programa de FTP, es puja el mòdul a la carpeta on es tinguin instal·lats els mòduls. Per defecte els mòduls van a la carpeta sites/all/modules.
4. Activar el mòdul. Per activar el mòdul, s'ha d'entrar com a administrador i anar a Administer / Site building / Modules. Des d'aquí es busca el mòdul instal·lat (es troba en el mòdul de views) i s'activa la vista que es necessita. En aquest cas s'ha d'activar el Views JSON i el Views.

**Nota:** Per poder fer funcionar el Views Datasource també es necessari tenir el mòdul Views.

Amb això ja es té el mòdul per poder canviar la vista d'una pàgina web en un arxiu JSON.

### 3.3.3.3 Com funciona

Per explicar el funcionament de la creació del JSON es posarà un exemple pràctic en un servidor de proves.

A la pàgina web <http://sergio.enandroid.com> s'ha creat un node amb una recepta de prova. El que es busca es aconseguir extreure aquesta informació en format JSON per així poder fer que es pugui visualitzar a la l'aplicació de CookingEat per mòvil. Per fer-ho, des del mode administrador, es va a la secció Administer/Site Building/Views i es veurà alguna cosa com aquesta.



sergio.enandroid.com

Home » Administer » Site building

**Views**

List Add Import Tools

Not sure what to do? Try the "Getting started" page.

Storage: Type: Tag: Displays:  
 - All - - All - - All - - All -

Sort by: Order:  
 Name Up Apply

Normal Node view: recetas	Edit   Export   Clone   Delete
Path: api/recetas Page	
Normal Node view: viewJson (vj)	Edit   Export   Clone   Delete
Path: jv Page	
Normal Recipe ingredient view: vista_json (v_json)	Edit   Export   Clone   Delete
Path: api_s/recetas Page	
Default Node view: archive (default)	Enable
Title: Monthly archive Path: archive Block, Page	Display a list of months that link to content for that month.

Il·lustració 7. Pantalla de configuració de Views

Aquí es troba tota la llista de vistes creades a la pàgina de drupal.

El següent pas es afegir la vista que mostrarà el JSON, es clica a la pestanya Add i apareixerà la següent pantalla.

**Views**

List Add Import Tools

**View name: \***

This is the unique name of the view. It must contain only alphanumeric characters and underscores; it is used to identify the view internally and to generate unique theming template names for this view. If overriding a module provided view, the name must not be changed or instead a new view will be created.

**View description:**

This description will appear on the Views administrative UI to tell you what the view is about.

**View tag:**

Enter an optional tag for this view; it is used only to help sort views on the administrative page.

**View type:**

Node  
Nodes are a Drupal site's primary content.

Recipe Unit  
The recipe unit list.

Recipe ingredient  
The recipe ingredients list.

Comment

Il·lustració 8. Afegir una nova vista



Aquí apareixerà:

- **View name:** per posar el nom de la vista a crear.
- **View description:** casella per si es vol posar una descripció de la vista.
- **View tag:** casella per posar un tag si interessa.
- **View type:** el tipus de vista que es vol fer, si d'un node, un tipus recepta, tipus ingredient, ...

Un cop omplertes les dades que interessin, es clica al boto next.

La següent pantalla és potser la més important, és el menu principal de configuració de les vistes en Drupal.

**Edit view vistaJson**

List Add Edit Import Tools

View vistaJson, displaying items of type Node. Export Clone New view

Defaults Page Display the view as a page, with a URL and menu links. Clone display Remove display

**Page**

Page Add display Analyze

**Basic settings**

Name: Page  
 Title: None  
 Style: Unformatted  
 Row style: Fields  
 Use AJAX: No  
 Use pager: No  
 Items to display: 10  
 Distinct: No  
 Access: Unrestricted  
 Caching: None  
 Exposed form in block: No  
 Header: None  
 Footer: None  
 Empty text: None  
 CSS class: None  
 Theme: Information

**Page settings**

Path: None  
 Menu: No menu

**Relationships** + ↑ ↓  
 None defined

**Arguments** + ↑ ↓  
 None defined

**Fields** + ↑ ↓  
 None defined

**Sort criteria** + ↑ ↓  
 None defined

**Filters** + ↑ ↓  
 None defined

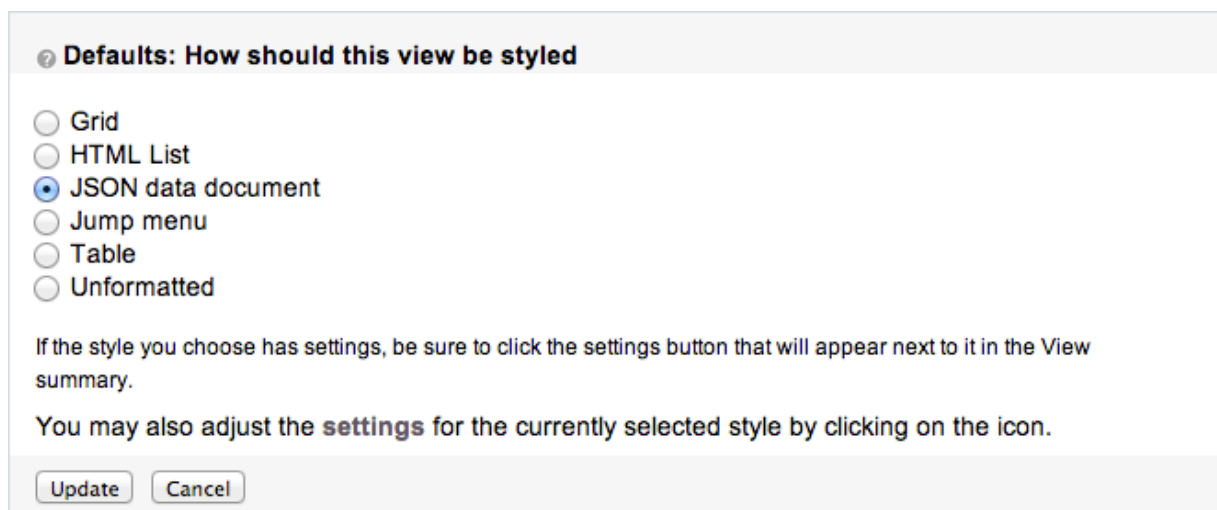
Click on an item to edit that item's details.

Save Cancel

Il·lustració 9. Edició d'una vista en Drupal

En aquesta pantalla s'han de veure els tres punts més importants per poder crear la vista, a la columna la opció Style, a la segona columna Path i a la cinquena columna Fields.

Si es clica a Style sortirà els següent menú:



Il·lustració 10. Escollir estil de la vista

Aquí s'ha d'indicar quin estil es vol definir per a la llista. Es buscarà un JSON data document, o sigui que es selecciona i es clica a "Update default display" sortint un altre menú com el que es veurà a la següent imatge.

**Page: Style options**

Status: using default values. Override

**Root object name:**  
  
The name of the root object in the JSON document. e.g nodes or users or forum\_posts

**Top-level child object:**  
  
The name of each top-level child object in the JSON document. e.g node or user or forum\_post

**Field output:**

Normal  
 Raw

For each row in the view, fields can be output as either the field rendered by Views, or by the raw content of the field.

Plaintext output  
For each row in the view, strip all markup from the field output.

**JSON data format:**

Simple  
 MIT Simile/Exhibit

What object format will be used for JSON output.

**JSONP prefix:**  
  
If used the JSON output will be enclosed with parentheses and prefixed by this label, as in the JSONP format.

**Content-Type:**

Default: application/json  
 text/json

The Content-Type header that will be sent with the JSON output.

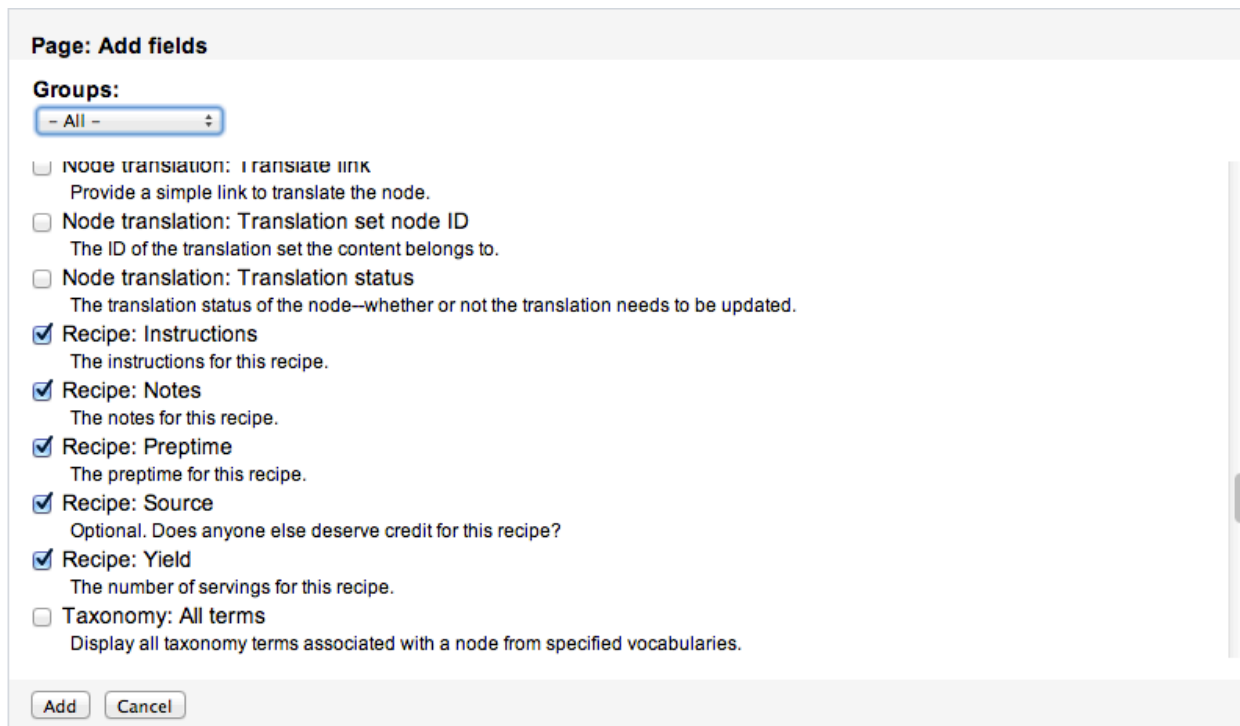
Views API mode  
Not using View API mode means the JSON gets output directly and the server ceases normal page processing. Using it means the server does not cease processing after outputting the JSON. This allows the Views API to be used with the view without having to prematurely terminate page processing.

Il·lustració 11. Formatar el JSON

D'aquesta pantalla com a molt es modificarà els dos primers camps. Root object name serveix per canviar el nom a l'arrel del JSON i Top-level child object per canviar el nom del seu primer fill. Les altres dades es deixem com estan per defecte i es clica a Update default display.

Ara es canvia a la segona columna i es clica a "Path". Aquest camp és la ruta amb la que volem que ens surti el nostre JSON. Guardem i anem al següent punt.

L'últim camp que ens queda es Fields. Aquí se li diu quins camps són els que es volen veure en aquesta vista en format JSON, o sigui que es clicarà sobre el signe + i sortirà el següent menú d'opcions.



Il·lustració 12. Escollir els camps de la vista

En aquest menú s'escollirà quines opcions es veuran en el JSON, com que el que es vol veure són receptes, s'escolliran totes les categories del tipus "Recipe" i la opció de "Node: Title" per tenir el títol de la recepta. Es clica a Add i ja està preparat.

Ara es graba aquesta vista i si ns'escriu la URL que s'havia posat al path de la vista obtindrem el següent resultat:

```
"nodes" : [
  {
    "node" : {
      "Title" : "tortilla de patatas",
      "Instructions" : "Batir huevos\ncortar cebolla y patatas en juliana\nhacer patatas y cebolla a fuego lento",
      "Preptime" : "20",
      "Source" : "Para mi",
      "Yield" : "1"
    }
  }
]
```

## 3.4 Eines i materials

### 3.4.1 Eclipse

Eclipse és un entorn de programari multi-llenguatge de desenvolupament que comprèn un espai de treball i un sistema de plug-ins. Està escrit principalment en Java.

Es pot utilitzar per desenvolupar aplicacions en Java i, per mitjà de diversos plug-ins, altres llenguatges de programació com Ada, C, C ++, COBOL, Fortran, Haskell, Perl, PHP, Python, R, Ruby (incloent Ruby on Rails ), Scala, Clojure, Groovy, i Scheme.

També es pot utilitzar per a desenvolupar per al programari Mathematica. Els entorns de desenvolupament inclouen el desenvolupament d'eines d'Eclipse Java (Jdt) per Java, Eclipse CDT per a C / C ++ i Eclipse PDT per PHP, entre altres.

### 3.4.2 ADT

Android Development Tools (ADT) és un plugin per l'IDE d'Eclipse que està dissenyat per donar-li un ambient potent, integrat en el qual la construcció d'aplicacions d'Android.

ADT amplia les capacitats d'Eclipse perquè pugui configurar ràpidament nous projectes d'Android, crear una interfície d'usuari de l'aplicació, afegir paquets basats en l'API d'Android Framework, depurar les seves aplicacions utilitzant les eines del SDK d'Android, i fins i tot exportar signat, arxius APK amb la finalitat de distribuir l'aplicació.

El desenvolupament en Eclipse amb ADT és molt recomanable i és la forma més ràpida de començar. Amb la configuració del projecte guiada que ofereix, així com la integració d'eines, editors de XML personalitzat, i el panell de resultats de depuració, ADT li dóna un impuls increïble en el desenvolupament d'aplicacions d'Android.

### 3.4.3 Eines de disseny (Photoshop)

Photoshop és l'aplicació líder en l'edició digital d'imatge per l'Interne i altres disciplines dels nous mitjans. És adoptat per milions d'artistes gràfics, dissenyadors o comunicadors d'impressió visual.

## 3.5 Arquitectura client-servidor asíncrona

### 3.5.1 Rest (J-SON)

Representational State Transfer (REST) és un estil d'arquitectura de programari per a sistemes distribuïts com la World Wide Web. REST s'ha convertit en un model de disseny de serveis web.

Els principals objectius de REST són:

- La escalabilitat de les interaccions entre components
- La generalitat de les interfícies
- Desplegament independent dels components
- Components intermedis per reduir la latència, reforçar la seguretat i encapsular els sistemes heretats

REST s'ha aplicat per descriure l'arquitectura web desitjada, per ajudar a identificar els problemes existents, per comparar alternatives de solució i per garantir que les extensions de protocol no violaria les restriccions bàsiques que fan que la web sigui un èxit.

### 3.5.2 XML

Extensible Markup Language (XML) és un llenguatge de marques que defineix un conjunt de regles per a documents de codificació en un format que és alhora llegible i llegible per la màquina. Es defineix en l'especificació XML 1.0 produït pel W3C, i diverses altres especificacions relacionades, tots els estàndards oberts gratuïtament.

Els objectius de disseny de XML donen èmfasi en la simplicitat, generalitat i facilitat d'ús a través d'Internet. Es tracta d'un format de dades textual amb un fort suport a través d'Unicode per als idiomes del món. Encara que el disseny de XML es centra en els documents, que és àmpliament utilitzat per a la representació d'estructures de dades arbitràries, per exemple en els serveis web.

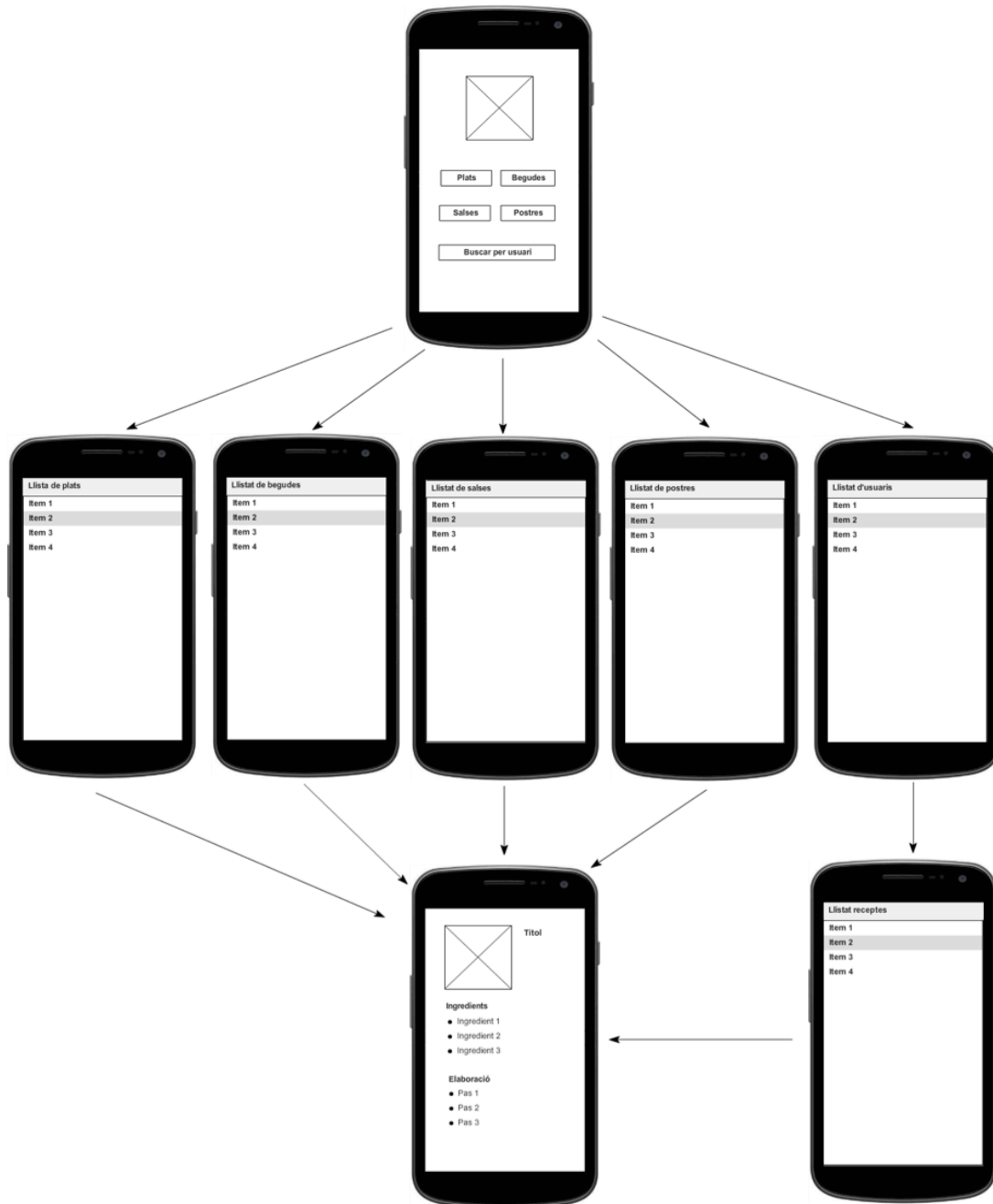
Moltes interfícies de programació d'aplicacions (API) s'han desenvolupat per ajudar els desenvolupadors de programari amb el processament de dades XML, i diversos sistemes d'esquema existeixen per ajudar en la definició de llenguatges basats en XML.





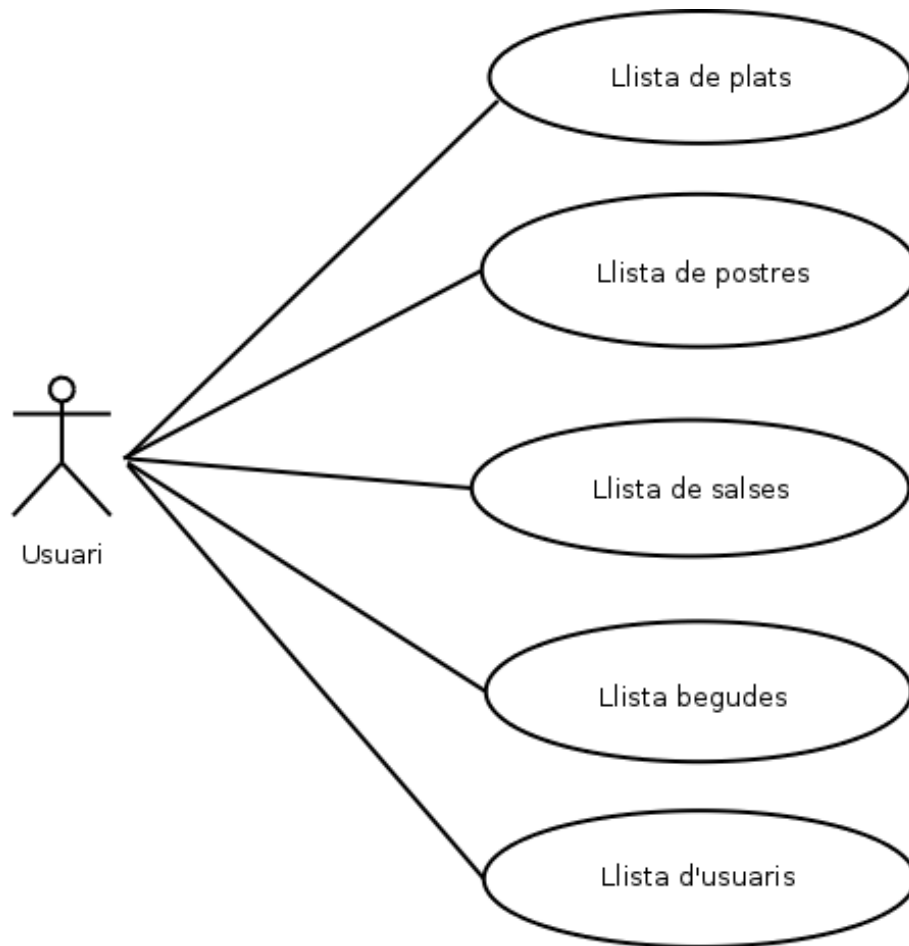
## 4 Definició de l'aplicatiu

### 4.1 Mock-up



Il·lustració 13. Mock-Up de l'aplicació

## 4.2 Casos d'ús



Il·lustració 14. Casos d'ús de l'aplicació

### **4.2.1 Llista de plats**

#### **FLUX INICIAL**

- L'usuari entra a l'aplicació.
- L'usuari clica al botó “Platos”.
- L'aplicació mostra la llista total de receptes de la web.
- L'usuari clica sobre la recepta que vol que li mostri.
- L'aplicació mostra la recepta.

### **4.2.2 Llista de postres**

#### **FLUX INICIAL**

- L'usuari entra a l'aplicació.
- L'usuari clica al botó “Postres”.
- L'aplicació mostra la llista total de receptes que són del tipus “Postres”.
- L'usuari clica sobre el postre que vol que li mostri.
- L'aplicació mostra la recepta.

### **4.2.3 Llista de salses**

#### **FLUX INICIAL**

- L'usuari entra a l'aplicació.
- L'usuari clica al botó “Salsas”.

-L'aplicació mostra la llista total de receptes que són del tipus "Salsas".

-L'usuari clica sobre la salsa que vol que li mostri.

-L'aplicació mostra la recepta.

#### **4.2.4 Lista de begudes**

##### **FLUX INICIAL**

-L'usuari entra a l'aplicació.

-L'usuari clica al botó "Bebidas".

-L'aplicació mostra la llista total de receptes que són del tipus "Bebidas".

-L'usuari clica sobre la salsa que vol que li mostri.

-L'aplicació mostra la recepta.

#### **4.3 Llista d'usuaris**

##### **FLUX INICIAL**

-L'usuari entra a l'aplicació.

-L'usuari clica al botó "Buscar por usuario".

-L'aplicació mostra la llista total d'usuaris.

-L'usuari clica sobre l'usuari que li interessa per veure les seves receptes.

-L'aplicació mostra la llista de receptes del usuari cercat.

-L'usuari clica sobre la recepta que vol que li mostri.

-L'aplicació mostra la recepta.

### **FLUX ALTERNATIU**

-L'usuari entra a l'aplicació.

-L'usuari clica al botó “Buscar por usuario”.

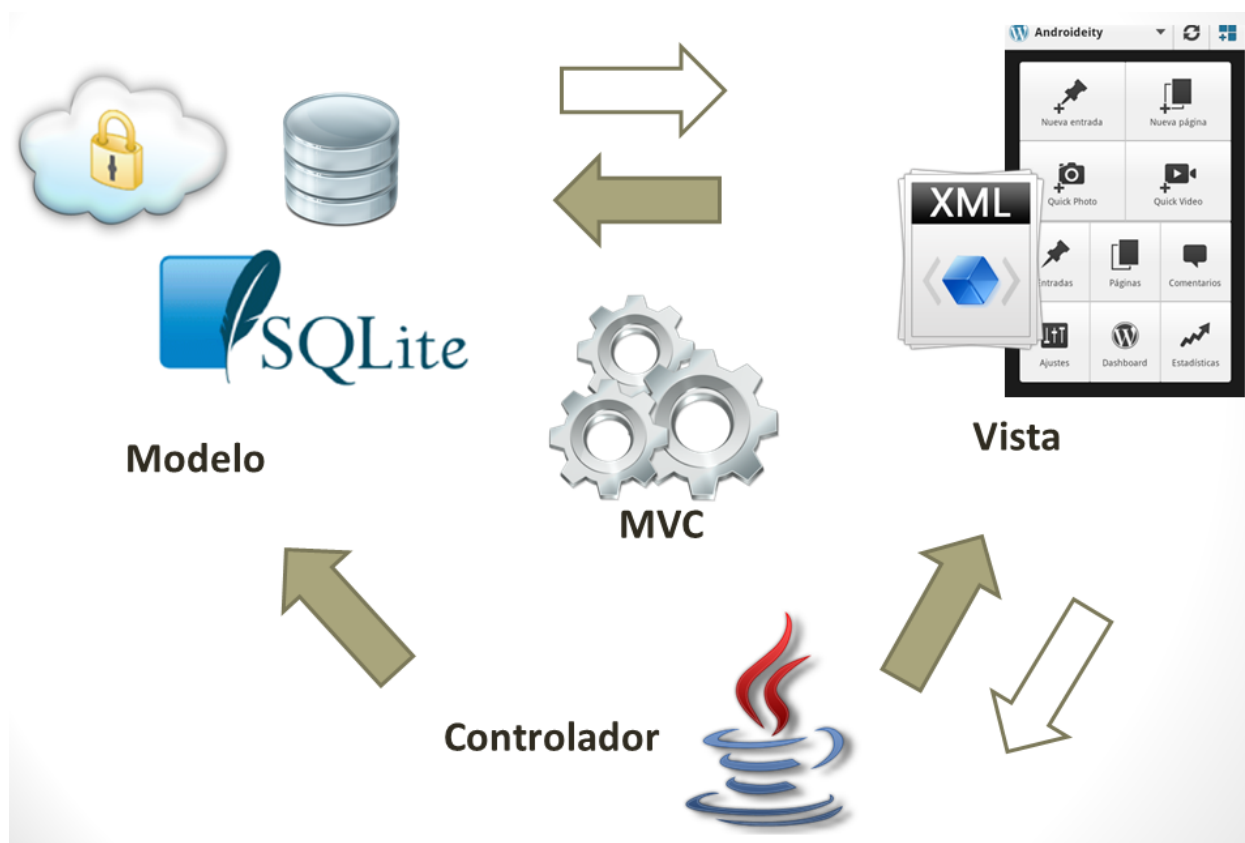
-L'aplicació mostra la llista total d'usuaris.

-L'usuari clica sobre l'usuari que li interessa per veure les seves receptes.

-Si l'usuari no ha ficat cap recepta, l'aplicació mostrarà el missatge de que aquest usuari no te cap recepta inserida.

## 4.4 Model vista-controlador

El patró d'arquitectura principal de les aplicacions Android és el patró model-vista-controlador, MVC a partir d'ara, que ens serveix per separar les dades de l'aplicació, l'interfície d'usuari i la lògica de negoci en tres components ben diferenciades, i que es relacionaran al final per obtenir el resultat buscat.



Il·lustració 15. Esquema de funcionament del patró MVC

D'aquesta manera és molt més senzill seccionar la feina que s'ha de fer, podent crear les diferents seccions específiques d'una forma ben diferenciada, inclòs podent reaprofitar les seccions en altres projectes.

### 4.4.1 Què es què en aquest projecte

**Model.** El model és la part d'on s'extrau la informació que es mostrarà a l'usuari. Normalment aquesta informació s'extreu d'una base de dades en MySQL o d'una Web Service. En aquest cas la informació s'extreu de la segona opció amb la classe "CkeJson" que es l'encarregada de buscar i transformar les dades.

**Vista.** La vista es la interfície amb la que interactuarà l'usuari. En Android aquestes es construeixen en XML. Tant el que seria l'esquelet de l'aplicació, com el format de com es veurà, canviar el color, la lletra, la forma, es fa amb el XML.

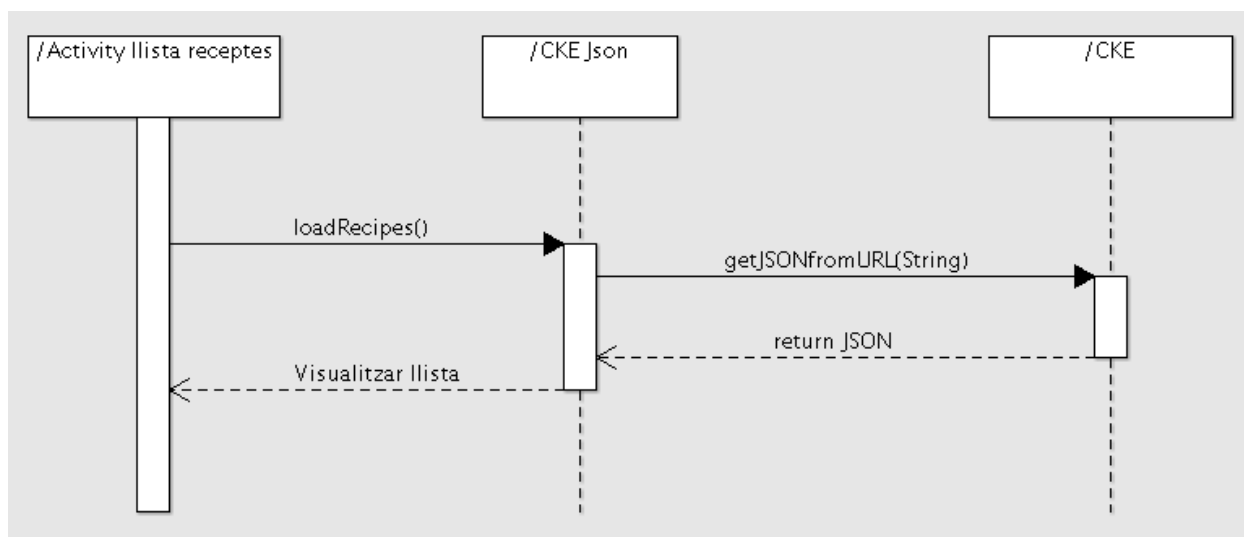
**Controlador.** El controlador és la classes que ajudarà a donar-li vida a la nostra interfície que ja s'ha construït i permetrà mostrar i consumir la informació per l'usuari. Aquests controladors es programen en Java i sense ells no es mostraria cap informació. En el cas d'aquesta aplicació són totes les classes que fan mostrar la llista o recepta seleccionada per pantalla.

## 4.5 Diagrama de seqüència

Aquesta aplicació tres parts diferenciades, llistar receptes, llistar usuaris i mostrar la recepta escollida.

### 4.5.1 Llistar receptes

En aquest diagrama, un cop l'usuari ha escollit la secció que vol, mostrarà la llista dels plats seleccionats.

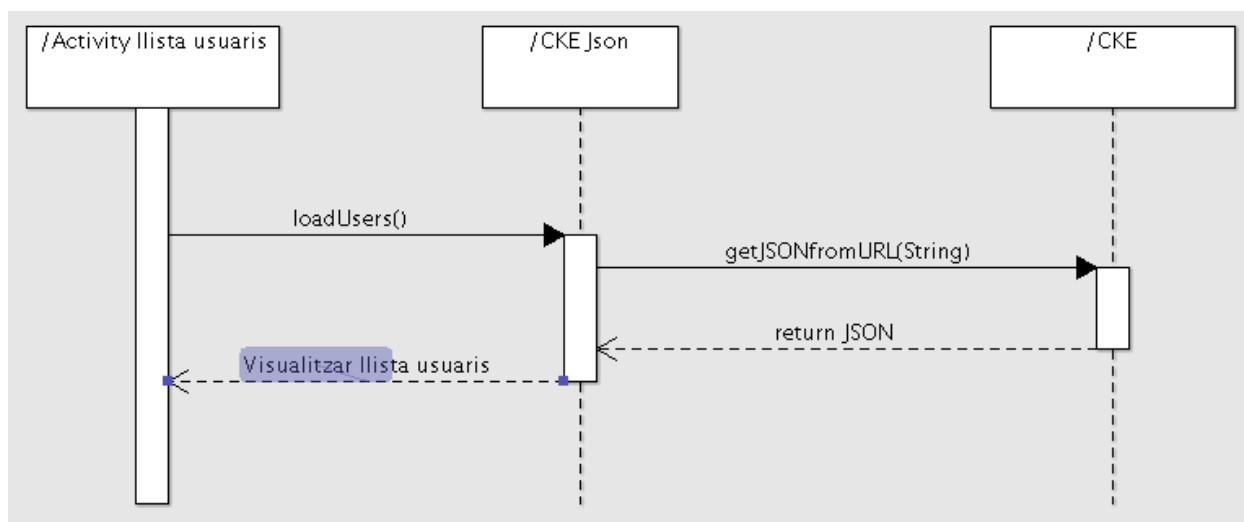


**Il·lustració 16. Diagrama de seqüència de llistar receptes**

Primerament, l'activitat crida al mètode (`loadRecipes()`). Aquest mètode és el que s'encarrega de fer la petició a la web de CookingEat a través de la classe `CkeJson`. Quan ha acabat de fer tot el tractament de la informació, la retorna en forma de JSON, ho emmagatzema tot en dos `ArrayList`, un que serà la que tindrà els títols de la recepta que es mostraran per pantalla i un altra que contindrà la recepta sencera en el cas de que l'usuari vulgui veure una recepta o un altre.

## 4.5.2 Llistar usuaris

En aquesta activitat mostrarà la llista d'usuaris de la pàgina web.



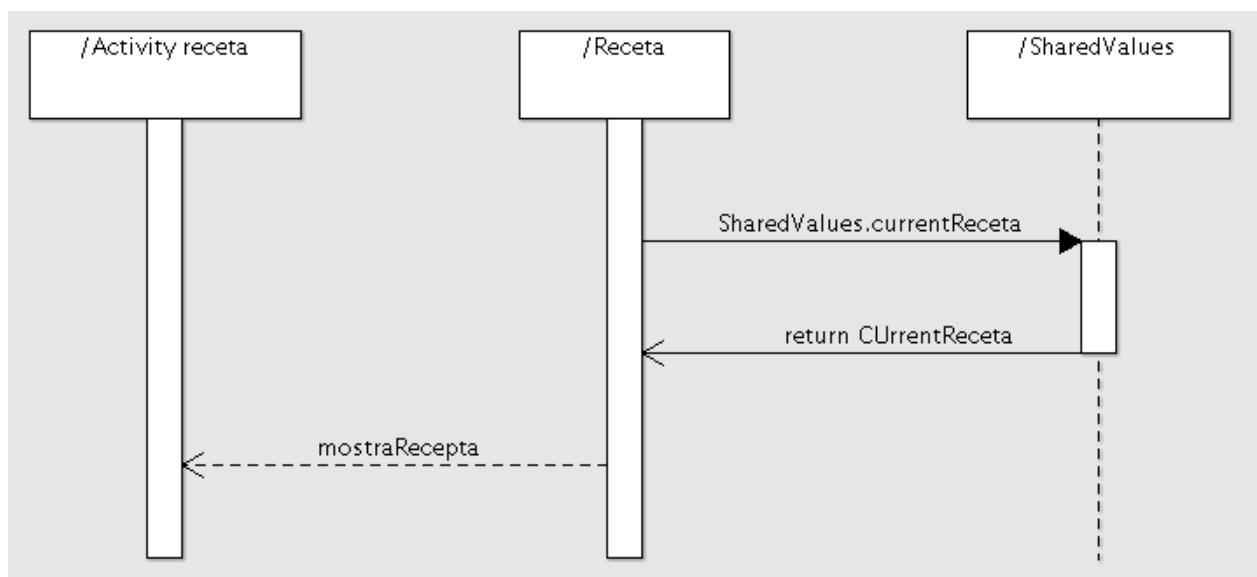
**Il·lustració 17. Diagrama de seqüència de llistar usuaris**



Quan s'ha clicat a la opció d'usuaris, carrega l'activitat "Usuarios". Aquesta crida al mètode (loadUsers()), agafa la web que conté el JSON i a través de la classe CkeJson, com en l'anterior diagrama, tracta el JSON per extreure'n tota la informació que es busca. Un cop acabat, retorna la informació cercada la qual guardarem en un ArrayList i la mostrem per pantalla en forma de ListView.

### 4.5.3 Mostrar recepta

Aquesta activity mostrarà la recepta escollida en qualsevol de les activitats de l'aplicació.



Il·lustració 18. Diagrama de seqüència de mostrar recepta

Un cop creada la llista de receptes en les activitats, quan es clica sobre alguna de les receptes, aquesta es guarda en un objecte SharedValues, el qual quan carrega l'activitat de mostrar recepta, passa la informació en un objecte Receta i retornem la informació a mostrar per aquesta activitat.



## 5 Desenvolupament de l'aplicació

### 5.1 Disseny de l'aplicació

Com s'ha vist al mock-up mostrat anteriorment, l'aplicació consta de quatre tipus de pantalles diferents: el login, el menú principal (amb els seus corresponents botons), els llistats (plats, begudes, salses, postres i buscar per usuari) i recepta.

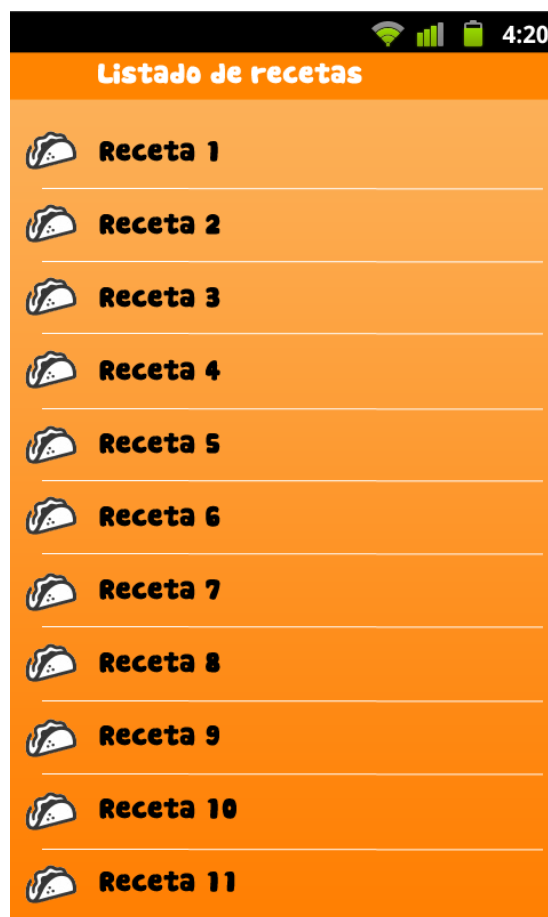
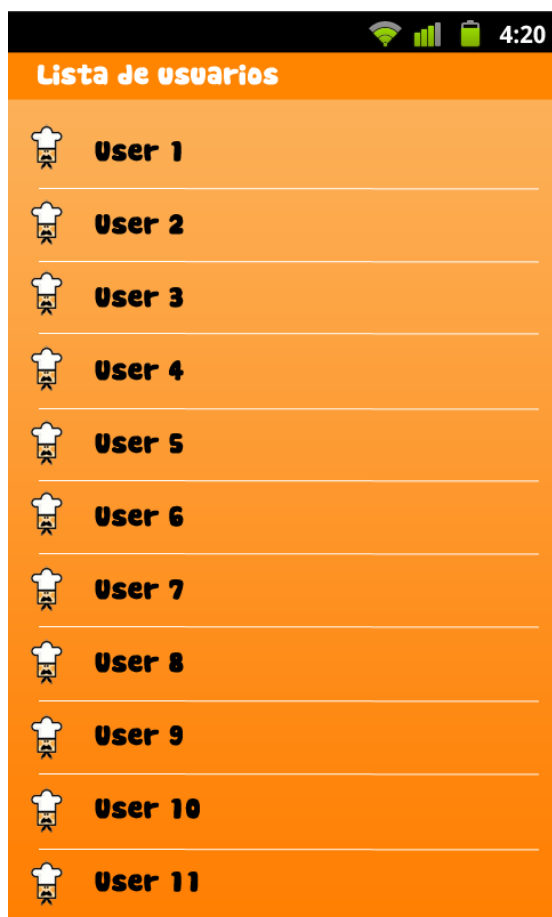
El login demanarà un usuari i contrasenya per a accedir a l'aplicació. En un futur es podria fer directament un login amb Facebook per així estalviar temps a l'usuari.

Per altra banda, al menú se li dona la benvinguda a l'usuari, a més de l'opció de desloguejar, i se li presenta les diferents opcions per a accedir a la recepta desitjada.



Un cop decidit on entrar, si es un llistat d'usuaris es veurà el llistat de tots els usuaris registrats a la web (o a través de l'aplicació) i si s'accedeix a un d'ells, el llistat de receptes que ha escrit.

Per altra banda, els altres llistats seràn de receptes específiques depenent del tipus d'àpat que s'hagi escollit (plats, begudes, postres o salses).





Per acabar, la pantalla de recepta mostrarà tota la informació necessària per a realitzar el plat escollit.

En aquest cas, es veurà la fotografia de la recepta (si en té), el títol del plat, quins són els ingredients que es necessiten i quina quantitat, i quina és l'elaboració necessària.

Totes les receptes per defecte es mostraran per a dues persones, si es vol cuinar per a més s'haurà de fer el càlcul manualment fins a la pròxima actualització de l'aplicació, on s'implementaran noves funcionalitats i informació més àmplia de cada recepta.

## 5.2 Codificació de l'aplicació

Aquest projecte ha estat realitzat en dues parts; primer, l'estudi de l'eina Drupal per a la creació dels JSON necessaris per poder mostrar la informació a l'aplicació, com ja s'ha explicat en la secció de l'estudi previ del Drupal, i la segona, el desenvolupament de l'aplicació Android. En aquesta secció es parlarà dels elements que s'han utilitzat i creat per a l'aplicació de CookingEat.

## 5.3 Android Manifest

Totes les aplicacions en Android tenen un AndroidManifest.xml en el seu directori arrel. Aquest arxiu presenta la informació principal sobre l'aplicació al sistema Android, desde la

versió Android que es fa servir per córrer l'aplicació, fins els accessos necessaris per a que funcionin bé, com els permissos d'accés a internet en aquest cas. Sense aquest arxiu l'aplicació no pot funcionar.

A continuació es veuran les diferents parts del manifest d'aquesta aplicació en concret.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
  package="com.tecnocampus.cookingeat"  
  android:versionCode="1"  
  android:versionName="1.0" >
```

Definició del manifest. Això ho crea l'Eclipse per defecte. Indica la ruta del manifest, el package on està i la versió i nom del projecte.

```
<uses-sdk  
  android:minSdkVersion="8"  
  android:targetSdkVersion="8" />
```

SDK que fa servir l'aplicació. Indica la versió mínima del sdk per funcionar i la màxima.

```
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Permisos que necessita l'aplicació per funcionar correctament.

```
<application  
  android:allowBackup="true"  
  android:icon="@drawable/icon"  
  android:label="@string/app_name"  
  android:theme="@style/AppTheme" >
```

Declaració de l'aplicació. Aquí es pot canviar els elements que es veuran per defecte a l'aplicació, com per exemple, el icone que es veurà per defecte si no es modifica a cap altre lloc.

```
<activity
    android:name="com.tecnocampus.cookingeat.MainActivity"
    android:label="@string/app_name"
    android:screenOrientation="portrait" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

Activity principal. Aquesta serà l'activitat d'inici de l'aplicació.

```
<activity
    android:name="com.tecnocampus.cookingeat.RecetasUsuario"
    android:label="@string/title_activity_recetas_usuario"
    android:screenOrientation="portrait" >
</activity>
```

Activity RecetasUsuario. Mostra les receptes del usuari.

```
<activity
    android:name="com.tecnocampus.cookingeat.UsuariosActivity"
    android:label="@string/title_activity_usuarios"
    android:screenOrientation="portrait" >
</activity>
```

Activity Usuarios. Mostra la llista d'usuaris.

```
<activity
    android:name="com.tecnocampus.cookingeat.Bebidas"
    android:label="@string/title_activity_recetas"
    android:screenOrientation="portrait" >
</activity>
```

Activity bebidas. Mostra les receptes de begudes.

```
<activity
    android:name="com.tecnocampus.cookingeat.RecetaActivity"
```

Activity RecetasUsuario. Mostra la recepta que s'ha clicat.

```
<activity
    android:name="com.tecnocampus.cookingeat.SalsasActivity"
    android:label="@string/title_activity_salsas"
    android:screenOrientation="portrait" >
</activity>
```

Activity SalsasActivity. Mostra les receptes salsa.

```
<activity
    android:name="com.tecnocampus.cookingeat.PlatosActivity"
    android:label="@string/title_activity_platos"
</activity>
```

Activity PlatosActivity. Mostra les receptes.

```
<activity
    android:name="com.tecnocampus.cookingeat.PostresActivity"
    android:label="@string/title_activity_postres" >
</activity>
```



Activity PostresActivity. Mostra les receptes postres.

## 5.4 Pantalles de l'aplicació

Com totes les funcions de l'aplicació són molt semblants la una de l'altra l'explicació se centrarà només en els activitys de buscar per usuari, per a que es tingui una idea de com ha estat programada l'aplicació.

### 5.4.1 Main Activity

Aquesta es la pantalla principal del projecte, consisteix en cinc botons que direccionaran a una secció o un altra de l'aplicació depenen de quin tipus de receptes es volen veure, si receptes en general, postres, salses, begudes o buscar receptes per usuari..

A la figura adjunta es mostra el codi XML de la pantalla principal

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/fondo_inicio"
    tools:context=".MainActivity" >

    <ImageButton
        android:id="@+id/imageBebidas"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignRight="@+id/imageUsuarios"
        android:layout_alignTop="@+id/imageSalsas"
        android:background="@drawable/button_aceptar_bebidas"
    />

    <ImageButton
        android:id="@+id/imagePlatos"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/imageSalsas"
        android:layout_alignRight="@+id/imageSalsas"
        android:layout_marginBottom="18dp"
        android:background="@drawable/button_aceptar_platos"
    />

    <ImageButton
        android:id="@+id/imagePostre"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignRight="@+id/imageBebidas"
        android:layout_alignTop="@+id/imagePlatos"
        android:background="@drawable/button_aceptar_postres"
    />

    <ImageButton
        android:id="@+id/imageSalsas"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/imageUsuarios"
        android:layout_alignLeft="@+id/imageUsuarios"
```

```

android:layout_marginBottom="22dp"
    android:background="@drawable/button_aceptar_salsas"
/>

<ImageButton
    android:id="@+id/imageUsuarios"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="58dp"
    android:background="@drawable/button_aceptar_usuarios"
/>

```

En el background dels botons, en comptes d'apuntar a una imatge, s'apuntarà a un altre XML situat a la carpeta drawable per a que faci l'efecte de botò. La codificació es la següent.

Desde aquest XML es controla que quan es clica sobre un botò, en el codi seria el “`android:state_pressed`” aparegui una imatge quan es true i quan es deixa anar passa a false, aparegui l'altre imatge, fent l'efecte de botò. Es podria haver controlat desde el codi Java directament però fent-ho així separem disseny de model fent el codi més net.

```

<selector xmlns:android="http://schemas.android.com/apk/res/android" >
    <item android:state_pressed="true"
        android:drawable="@drawable/boton_usuarios_up" />
    <item android:state_pressed="false" android:drawable="@drawable/boton_usuarios" />
</selector>

```

### 5.4.2 Gestió de dades

Per gestionar la informació, l'aplicació fa servir la classe “CkeJson” per obtenir les dades de la pàgina web.

Aquesta classe és la que gestiona les peticions Http que es fan de les diferents classes i retorna un objecte JSONObject amb la informació cercada.

El string passat per paràmetre es passa per un `HttpPost` i es converteix en un `InputStream`. Si la URL es erronia retorna una excepció.

Després, com que es necessita que el retorn del servidor estigui en la codificació correcta, amb el `BufferedReader` es posa la codificació que interessa. En el cas de que algun dels passos anteriors fos incorrecte, llençaria una excepció, i si tot va bé retorna el `JSONObject`.

```
package com.tecnocampus.cookingeat.persistencia;

import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.json.JSONException;
import org.json.JSONObject;

import android.util.Log;

public class CkeJson {
    public static JSONObject getJSONfromURL(String req) {
        InputStream is = null;
        String result = "";
        JSONObject json = null;
        try {
            HttpClient httpclient = new DefaultHttpClient();
            HttpPost httppost = new HttpPost(req);
            HttpResponse response = httpclient.execute(httppost);
            HttpEntity entity = response.getEntity();
        }
    }
}
```

```
        is = entity.getContent();
    } catch (Exception e) {
        Log.e(req, e.toString());
    }

    try {
        BufferedReader reader = new BufferedReader(new
InputStreamReader(is, "ISO-8859-1"), 8); //iso-8859-1
        StringBuilder sb = new StringBuilder();
        String line = null;
        while ((line = reader.readLine()) != null) {
            sb.append(line + "\n");
        }
        is.close();
        result = sb.toString();
    } catch (Exception e) {}

    try {
        json = new JSONObject(result);
    } catch (JSONException e) {
        Log.e(req, e.toString());
    }

    return json;
}
}
```

Aquesta és la codificació que fa que els botons cobrin vida. El primer que es fa quan crea la activitat, es declara el “ImageButton” per saber a quin item apunta, i a partir d’aquí amb el “setOnClickListener” es programa el “onClick” de cadascun dels botons, activant el “ProgressDialog” per posar una pantalla de càrrega mentre es carrega la llista a buscar depenent del botó al que s’ha clicat i es crea el “Intent” que serà l’encarregat de fer passar al següent activity.

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    requestWindowFeature(Window.FEATURE_NO_TITLE);
    setContentView(R.layout.activity_main);

    ImageButton goUsuarios = (ImageButton)
findViewById(R.id.imageUsuarios);

    goUsuarios.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View arg0) {
            progressDialog = ProgressDialog.show(MainActivity.this,
""
,
                "Cargando...", true, true);
            Intent usuarios = new Intent(getApplicationContext(),
                UsuariosActivity.class);
            startActivity(usuarios);
            new Thread(new Runnable() {
                public void run() {
                    progressDialog.dismiss();
                }
            }).start();
        }
    });

    ImageButton goBebidas = (ImageButton)
findViewById(R.id.imageBebidas);

    goBebidas.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View arg0) {
            progressDialog = ProgressDialog.show(MainActivity.this,
""
,
                "Cargando...", true, true);
            Intent bebidas = new Intent(getApplicationContext(),

```

```
public void run() {
    new Thread(new Runnable() {
        public void run() {
            progressDialog.dismiss();
        }
    }).start();
});

ImageButton goSalsas = (ImageButton) findViewById(R.id.imageSalsas);
goSalsas.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View arg0) {
        progressDialog = ProgressDialog.show(MainActivity.this,
        "",
            "Cargando...", true, true);
        Intent salsas = new Intent(getApplicationContext(),
            SalsasActivity.class);
        startActivity(salsas);
        new Thread(new Runnable() {
            public void run() {
                progressDialog.dismiss();
            }
        }).start();
    }
});

ImageButton goPlatos = (ImageButton) findViewById(R.id.imagePlatos);
goPlatos.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View arg0) {
        progressDialog = ProgressDialog.show(MainActivity.this,
        "",
            "Cargando...", true, true);
```

```

Intent platos = new Intent(getApplicationContext(),
                                PlatosActivity.class);
startActivity(platos);
new Thread(new Runnable() {
    public void run() {
        progressDialog.dismiss();
    }
}).start();

});

ImageButton goPostres = (ImageButton) findViewById(R.id.imagePostre);
goPostres.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View arg0) {
        progressDialog = ProgressDialog.show(MainActivity.this,
            "Cargando...", true, true);
        Intent postres = new Intent(getApplicationContext(),
            PostresActivity.class);
        startActivity(postres);
        new Thread(new Runnable() {
            public void run() {
                progressDialog.dismiss();
            }
        }).start();
    }
});
}

```



### 5.4.3 Botó Lista de usuarios

#### 5.4.3.1 *Activity\_usuarios.xml*

En el xml d'aquest activity es destacarà l'etiqueta "ListView".

El ListView és una etiqueta que crea una llista de valors de les dades que es vol buscar, i l'etiqueta "android:id" és l'identificador amb el que es reconeixerà en la codificació Java.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/fondo_lista_usuarios"
    android:orientation="vertical" >

    <ListView
        android:id="@+id/mylist"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="25dp"
        android:cacheColorHint="#00000000" >
    </ListView>

</LinearLayout>
```

### 5.4.3.2 *UsuariosActivity.java*

```
public class UsuariosActivity extends Activity {  
  
    private List<String> listUsuariosNombres;  
    ProgressDialog progressDialog;  
    private ListView listView;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        requestWindowFeature(Window.FEATURE_NO_TITLE);  
        setContentView(R.layout.activity_usuarios2);  
        listView = (ListView) findViewById(R.id.mylist);  
        loadUsers();  
    }  
}
```

Aquesta classe es la que fa que es mostri la llista d'usuaris en el xml. Per començar s'inicialitza un **List** amb el nom de `listUsuariosNombres`. Aquesta llista serà la que es farà servir, després d'obtenir les dades del JSON d'usuaris, per emmagatzemar-la i visualitzar-la posteriorment amb la variable `listView`, la qual s'inicia i es relaciona amb el xml a visualitzar amb el “`findViewById(R.id.mylist)`”. Un cop inicialitzades les variables es carrega el mètode “`loadUsers()`”.

Aquest mètode és el que escriu totes les dades buscades a la web a la aplicació. El seu funcionament és el següent:

Primerament es declararà una variable `String` que ha estat creada com hem explicat anteriorment es crea una nova vista del Drupal basada en els autors. Seguidament es crea la variable “`JsonUsuarios`” a la que se li passarà per valor el resultat del mètode `getJSONfromURL` explicat anteriorment. Amb aquest informació emmagatzemada, ja només queda passar la informació del JSON amb la funció “`getJSONObject`” i se li diu la ruta exacta que es vol per treure la dada que es busca, en aquest cas el nom del autor de la recepta. Un cop es té tota la llista ja només queda posar les dades a la llista creada en xml. Per fer-ho es farà servir un `ArrayAdapter` i se li passarà la llista.

```

public void loadUsers() {
    String req;
    req = ("http://cookingeat.com/data/autores");

    JSONObject JsonUsuarios = new JSONObject();
    JsonUsuarios = CkeJson.getJSONfromURL(req);

    int numUsuarios;

    try {
        numUsuarios = JsonUsuarios.getJSONArray("authors").length();
        listUsuariosNombres = new ArrayList<String>();
        for (int i = 0; i < numUsuarios; i++) {

listUsuariosNombres.add(JsonUsuarios.getJSONArray("authors")
                        .getJSONObject(i).getJSONObject("author")
                        .getString("Autor"));
        }

        ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
            android.R.layout.simple_list_item_1,
            android.R.id.text1, listUsuariosNombres);

        listView.setAdapter(adapter);
    }
}

```

Per últim se li dona funcionalitat de botó a la llista per poder escollir les receptes de l'usuari escollit anteriorment. Quan l'usuari de la app clica sobre l'usuari de la llista, activarà el “progressDialog” per posar la pantalla de càrrega i amb la classe “SharedValues” es guarda l'usuari de la posició en la que està a la llista i s'executa la següent activity que serà

```

listView.setOnItemClickListener(new OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view,
        int position, long id) {
        progressDialog = ProgressDialog.show(UsuariosActivity.this,
            "", "Cargando...", true, true);
        SharedValues.getInstance().setCurrentUser(
            listUsuariosNombres.get(position));
        Intent recetasUsuario = new Intent(getApplicationContext(),
            RecetasUsuario.class);
        startActivity(recetasUsuario);
        new Thread(new Runnable() {
            public void run() {
                progressDialog.dismiss();
            }
        })
    }
}

```

“RecetasUsuario”.

```

        }
    }).start();
}
});
} catch (JSONException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
}
}

```

## 5.4.4 Llista de receptes d'usuari

### 5.4.4.1 *activity\_recetas\_usuario.xml*

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    <ListView
        android:id="@+id/recetasUsuario"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="25dp"
        android:cacheColorHint="#00000000" >
    </ListView>
</LinearLayout>

```

### 5.4.4.2 *RecetasUsuarios.java*

Aquesta classe funciona de manera similar a l'anterior variant una mica el mètode

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    requestWindowFeature(Window.FEATURE_NO_TITLE);
    setContentView(R.layout.activity_recetas_usuario);

    listView = (ListView) findViewById(R.id.recetasUsuario);
    loadRecipes();
}

```

loadRecipes().

Aquest mètode, com s'ha dit abans, funciona quasi bé igual que el mètode loadUsers(), amb la diferència de que el String amb el JSON variarà depenent del usuari al que s'ha clicat, on

la ruta "req = ("http://cookingeat.com/data/recetas-de/" + SharedValues.getInstance().getCurrentUser());" on getCurrentUser() és l'usuari buscat.

La segona variació està en que ara es necessitarà, a part del array de la llista de receptes, un segon array del tipus "Receta" on es guardarà la informació de cadascuna de les receptes buscades, així, si l'usuari clica en alguna de les receptes, a través del "onClick" i la posició de la recepta cercada, s'obtidran les dades necessaries per la següent activity que mostra les receptes.

```

        r.setDescripcion(JsonRecetasUsuario.getJSONArray("Items")
                        .getJSONObject(i).getJSONObject("receta")
                        .getString("instructions"));
        listaRecetasUsuario.add(r);
    }

    ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
        android.R.layout.simple_list_item_1,
        android.R.id.text1,
        listaTitulos);

    listView.setAdapter(adapter);

    listView.setOnItemClickListener(new OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view,
        int position, long id) {
        progressDialog = ProgressDialog.show(RecetasUsuario.this,
            "", "Cargando...", true, true);
        SharedValues.getInstance().setCurrentReceta(
            listaRecetasUsuario.get(position));
        Intent recetas = new Intent(getApplicationContext(),
            RecetaActivity.class);
        startActivity(recetas);
        new Thread(new Runnable() {
            public void run() {
                progressDialog.dismiss();
            }
        }).start();
    });
});

```

## 5.4.5 Mostrar recepta

### 5.4.5.1 *Activity\_receta.xml*

Aquesta activity està creada de la següent manera:

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
```

Un “LinearLayout” principal que conté la informació per veure.

Un “ScrollView” que fa que la pantalla tingui scroll vertical per si la recepta és més llarga del que abarca la pantalla i poder baixar-la.

```
<ScrollView
    android:id="@+id/scrollView1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >
```

Un segon “LinearLayout” que conté un “TextView” que serà el títol de la recepta.

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="211dp"
    android:orientation="vertical" >

<TextView
    android:id="@+id/recetaActivity"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="15dp"
    android:layout_marginRight="15dp"
    android:layout_marginTop="25dp"
```

```
android:text="titulo"  
    android:textSize="22sp" />
```

Un “ImageView” si la recepta te imatge.

```
<ImageView  
    android:id="@+id/imageReceta"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="25dp"  
    android:layout_gravity="center_horizontal"  
    android:src="@drawable/ic_launcher" />
```

Un tercer “TextView” que contindrà la preparació de la recepta.

```
<TextView  
    android:id="@+id/preparacion"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="44dp"  
    android:layout_marginLeft="15dp"  
    android:layout_marginRight="15dp"  
    android:text="preparacion" />  
  
    </LinearLayout>  
</ScrollView>  
</LinearLayout>
```

#### 5.4.5.2 RecetaActivity.java

Aquesta classe mostra la recepta escollida en qualsevol de les llistes seleccionades desde el menú principal. Per començar només es necessita declarar un atribut “Receta” per poder

```
public class RecetaActivity extends Activity {
    private Receta cr; //current_receta

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.activity_receta);
    }
}
```

visualitzar les dades que es volen. Després simplement es passa el valor obtingut amb la classe “SharedValues” a l’atribut “cr” i es passen les dades al “textView” del títol, la imatge amb el mètode privat “mostrarImagen” i al TextView que mostrarà la preparació de la recepta.

Aquest “Typeface” serveix per poder canviar la font per defecte per la que es vulgui, afegint-la a la carpeta “assets”.

```
Typeface tf = Typeface.createFromAsset(getAssets(), "fonts/Cookies.ttf");

this.cr = SharedValues.getInstance().getCurrentReceta();
TextView textView=(TextView) findViewById(R.id.recetaActivity);
TextView preparacion = (TextView)findViewById(R.id.preparacion);
textView.setTypeface(tf);
mostrarImagen(cr.getImagen());
textView.setText(cr.getTitle());
```

En el moment de posar la descripció de la recepta per no tenir problemes amb la codificació del text i poder-ho veure correctament s'utilitza la classe “Spanned” per, amb el “setText”, poder-ho passar a un format llegible.



```
Spanned s = Html.fromHtml(cr.getDescripcion
    preparacion.setText(Html.fromHtml(s.toString()));
}
```

Mètode privat que mostra la imatge de la recepta si la té

```
private void mostrarImagen(String s){
    ImageView imageView =(ImageView) findViewById(R.id.imageReceta);

    URL url;
    try {
        url = new URL(s);

        Bitmap imagen;

        imagen
        BitmapFactory.decodeStream(url.openConnection() .getInputStream());
        imageView.setImageBitmap(imagen);
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

## 5.5 Altres classes

### 5.5.1 SharedValues.java

Aquesta classe singletona és un contenidor de les dades que necessita guardar l'aplicació, el usuari per poder mostrar la llista de receptes d'aquest usuari i la recepta per mostrar.

```
package com.tecnocampus.cookingeat.modelo;
public class SharedValues {
    private static SharedValues INSTANCE = null;

    // Current User
    private String currentUser = "";
    private Receta currentReceta = null;

    private SharedValues() {
    }

    public static SharedValues getInstance() {
        if (INSTANCE == null)
            INSTANCE = new SharedValues();
        return INSTANCE;
    }

    public String getCurrentUser() {
        return currentUser;
    }

    public void setCurrentUser(String currentUser) {
        this.currentUser = currentUser;
    }

    public void setCurrentReceta(Receta receta) {
        this.currentReceta = receta;
    }

    public Receta getCurrentReceta() {
        return currentReceta;
    }
}
```

## 6 Creació del arxiu APK

Si es vol que l'aplicació creada surti al Play Store, s'ha de crear l'arxiu apk, instal·lador en Android, i crear una keystore.

El sistema Android necessita que totes les aplicacions instal·lades han de ser signades digitalment amb un certificat en el que la clau privada només la tingui el desenvolupador de l'aplicació. El sistema Android fa servir aquest certificat com una manera d'identificar el autor d'una aplicació i establir relacions entre les aplicacions. El certificat no es fa servir per controlar quines aplicacions pot instal·lar l'usuari i no necessita ser signat per una autorització certificada, està totalment permès per les aplicacions Android utilitzar aplicacions signades per un mateix.

Punts importants sobre la signa digital en Android:

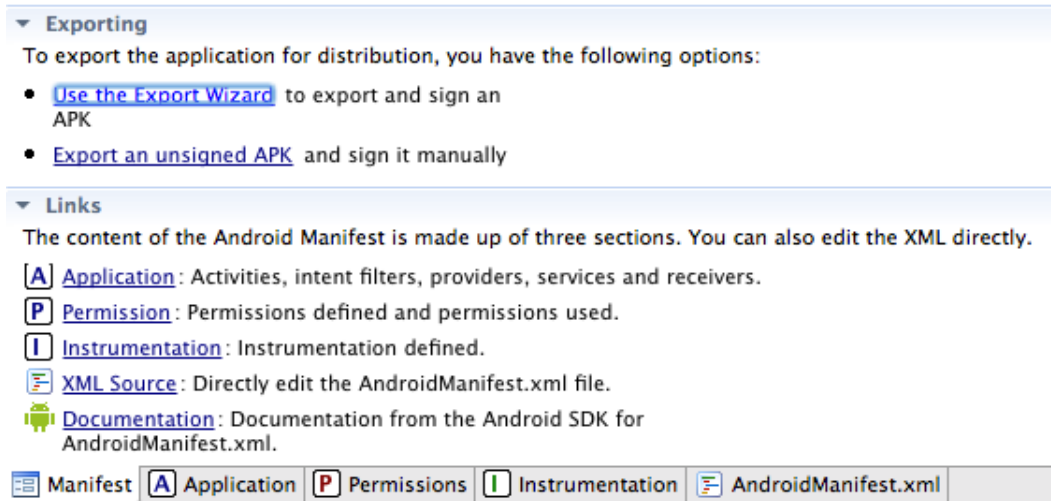
- Totes les aplicacions han de ser signades. El sistema no et deixarà instal·lar una aplicació que no hagi estat signada. Es una mesura de seguretat i un requisit de garantia.
- A l'hora de debugar, com s'ha fet servir l'Eclipse/ADT, aquest mateix programa ja crearà per defecte un debug certificate, amb el que es podrà instal·lar l'aplicació en un dispositiu o fer servir l'emulador.
- Quan es llença l'aplicació per l'usuari final, s'ha de general una keystore privada, la que es crea per fer proves no serveix per llençar-la al Play Store.
- D'aquesta forma amb la nostra keystore pròpia només el creador podrà actualitzar l'aplicació.

És molt important signar l'aplicació perquè sinó no es podrà fer servir amb cap dispositiu Android ni emulador.

### 6.1 Com generar l'arxiu APK i el Keystore

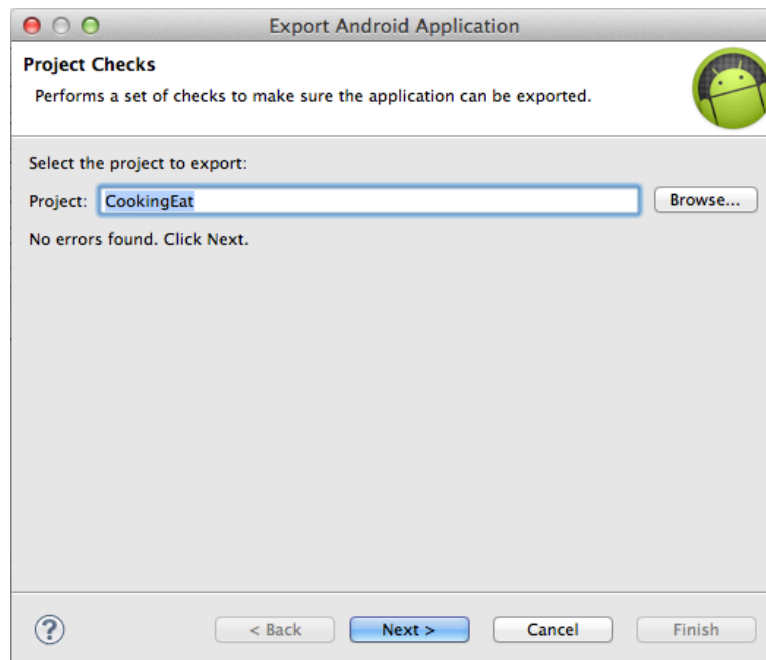
Per generar els dos arxius seguirem els següents passos fent servir l'Eclipse:

1. Des de l'abre de directori s'obre l'arxiu AndroidManifest.xml i a la primera pestanya a la part de exporting es fa click a "Use the export wizard" com es veu a la imatge següent.



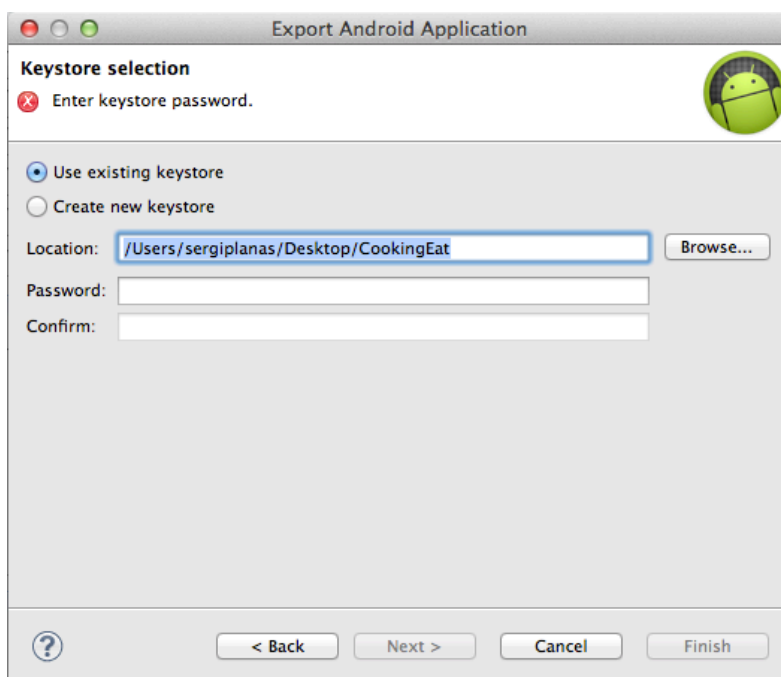
Il·lustració 19. Pantalla d'inici de creació d'arxiu APK

2. En aquesta pantalla s'escull el projecte a exportar, en aquest cas CookingEat.



Il·lustració 20. Selecció del nom de l'aplicació

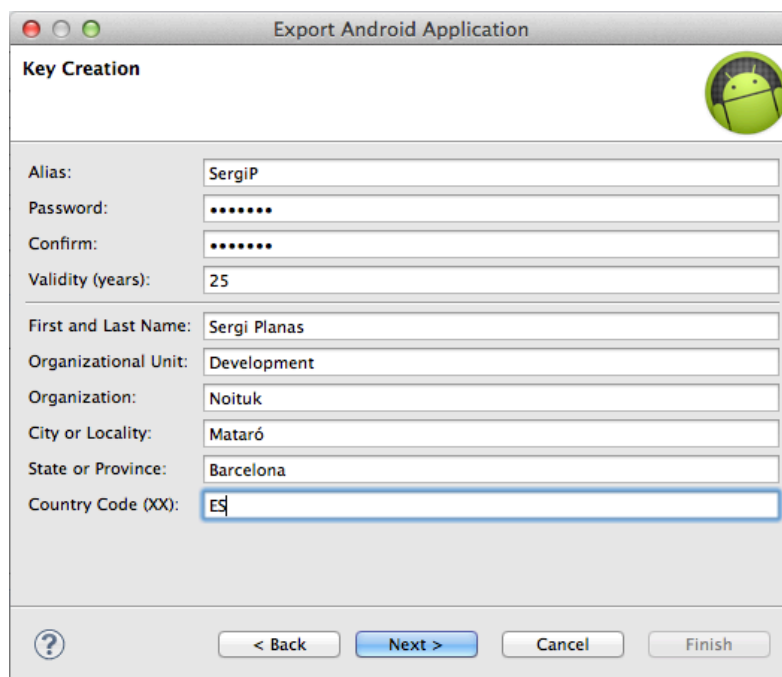
3. En aquesta pantalla es clicarà sobre “Create new keystore” i a Location es posarà la ruta on es vol crear la keynote i el password.



Il·lustració 21. Posició on es guardarà la keynote.

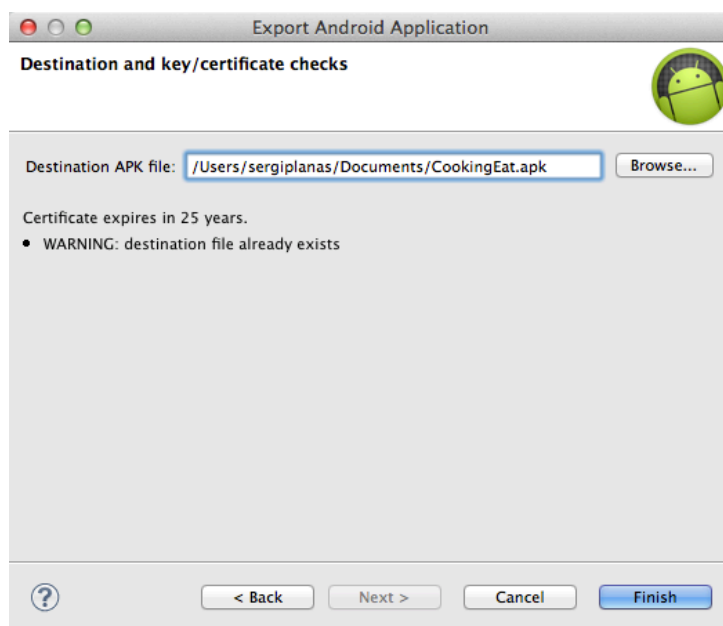
4. Aquí es posarà les dades personals a la keystore.
  - **Alias:** El alias que es farà servir per la keynote.
  - **Password:** Es torna a posar la contrasenya.
  - **Validity (years):** En aquesta part es defineix la duració de la validació de la keystore en anys. Per estar segurs que no expirarà aviat es posarà un número alt, l'Eclipse aconsella mínim 25 anys.

Els següents camps són informació personal i de la organització o empresa per la que es treballa.



Il·lustració 22. Creació de la key.

5. En aquesta pantalla, s'escull el directori on es guardarà l'arxiu .apk signat i també indicarà quan expirarà la keynote. Si es clica al botó finalitzar es tindrà l' aplicació creada i signada.



Il·lustració 23. Lloc on es guardarà l'arxiu .APK

## 7 Testing

Per testejar aquesta aplicació, a causa de la gran quantitat de dispositius mòbils Android que hi ha al mercat, no es pot assegurar que amb un sol testeig en un mòbil l'aplicació funcioni correctament. Per tant, per comprovar que l'aplicatiu funcioni com toca, procedirem al testeig amb diferents dispositius, Nexus S, Galaxy Nexus, Galaxy S3, Samsung Infuse 4g y Galaxy Mini.

Després de fer el testeig amb tots els dispositius, s'han trobat les diferents errades al sistema.

- **Error al escollir certs usuaris (tots els dispositius).** Aquesta errada ve donada a que hi ha certs usuaris que no tenen recepta, provocant un valor null a la variable i fent que l'aplicació deixi de funcionar. Per arreglar-ho simplement s'ha de comprovar que l'usuari tingui receptes disponibles.
- **El programa dóna error només entrar (Galaxy Mini).** Aquest és un problema de resolució, l'aplicació està pensada per visualitzar amb una resolució mínima de 800x480 ja que sinó es veuria massa petit.
- **Problemes de visualització (Galaxy Nexus, Galaxy S3).** Igual que l'error anterior, aquest és un problema de resolució, però en aquest cas a causa de tenir massa resolució fa que els botons surtin en llocs incorrectes, però la seva funcionalitat és correcte. Per arreglar l'error s'han d'adaptar les imatges per a una resolució més alta.
- **Certes receptes triguen a carregar i no es veuen ve les receptes (tots els dispositius).** Aquest error ve provocat a causa d'unes imatges massa grans. Per arreglar l'error les imatges han de ser redimensionades a un tamany adaptat al mòbil.





## 8 Problemes trobats

El problemes trobats durant la realització del projecte es van produir en dos punts, primer, buscant la forma d'enllaçar la pàgina web amb Android, i segon, en la programació en Android.

Drupal al ser un CMS lliure té un gran abast de mòduls i llibreries totalment gratuïts de tot tipus de coses que es vulguin fer per una pàgina web, però també és un inconvenient a l'hora de buscar alguna cosa en concreta. També s'ha de dir que de vegades la descripció dels mòduls està incompleta o que quan es troba alguna opció mínimament usable es veu que el projecte l'han deixat a mitges. Tot i això, hi ha una gran comunitat al darrera que si pot ajuda en tot el possible per a que es puguin eliminar els dubtes.

Un altra dels problemes trobats amb Drupal és la seva corba de dificultat. Normalment quan es comença amb una eina nova, la seva corba d'aprenentatge intenta ser baixa al principi i anar ampliant conceptes nous que dificulten el procés d'aprenentatge. En el cas de Drupal la corba d'aprenentatge es bastant més alta al inici que amb altres eines CMS d'aquest estil com podria ser Joomla. Això provoca que s'hagi de mirar molta informació sobre com funciona.

Seguidament amb Android, el problema trobat és la utilització de les activitats com a pantalles i els seus xml. S'ha d'estar molt atent a l'hora d'escriure el xml i escriure be els seus atributs, ja que depenent de com s'escriuin les etiquetes l'Eclipse no ho detecta com a error i al principi costa de veure el perquè ha deixat de funcionar l'aplicació. I sobretot, el `AndroidManifest.xml`, donat que aquest arxiu es el que diu si una aplicació funciona o no el dispositiu mòvil, s'ha de configurar correctament, i no només això, s'ha de vigilar de declarar cadascuna de les activitats, sinó el projecte no funcionarà.



## 9 Estudi econòmic

### 9.1 Materials i amortitzacions

Material necessari per a la creació del projecte i la seva amortització. Com a materials entra:

- Equip informàtic necessari per a la creació de l'aplicació
- Dispositiu mòbil Samsung Nexus S per testeig
- Material gràfic per la realització del disseny.
- En el cas de les llicències de software, al estar treballant amb aplicacions que tenen llicència OpenSource no es necessari pagar res.

Material	Cost
Llicències de software	0€
Equip informàtic	264.67€
Dispositiu mòbil per testeig	75€
Material gràfic	50€
<b>Total</b>	<b>389.67 €</b>

Taula 2. Materials i amortitzacions

- Equip informàtic:  $1588\text{€}/24 \text{ mesos} = 66.17\text{€/mes} * 4 \text{ mesos} = 264.67 \text{ €}$
- Smartphone Samsung Nexus S =  $450 \text{ €} / 24 \text{ mesos} = 18.75 \text{ €/mes} * 4 \text{ mesos} = 75 \text{ €}$
- Material gràfic:  $10\text{€ per il·lustració} * 5 = 50 \text{ €}$

## 9.2 Hores d'enginyeria

Les hores d'enginyeria són aquelles hores realitzades en la definició del projecte: Estudi de la competència, del sistema operatiu Android i Drupal, les eines necessàries per a aquest desenvolupament i la definició de l'aplicatiu: Mock-ups, casos d'ús, definició del MVC i els diagrames de seqüència.

Es considerarà que el preu per hora per els estudis es de 20 €/h i la definició serà de 50 €/h

<b>Activitat</b>	<b>Hores dedicades</b>	<b>Cost</b>
<b>Estudi de la competència</b>	6	120€
<b>Estudi d'Android i Drupal</b>	45	900€
<b>Estudi d'eines</b>	15	300€
<b>Definició</b>	52	2650€
<b>Total</b>		3970 €

Taula 3. Hores d'enginyeria

### 9.3 Hores de desenvolupament

Per les hores de desenvolupament, es considerarà que el sou mitg d'un programador Java/Android s'estima sobre els 20€/hora.

Activitat	Hores dedicades	Cost
Preparació del mòdul de 20 Drupal View Datasource		400€
Creació de l'estructura de 15 pantalles en Android		300€
Creació de l'API	45	900€
<b>Total</b>		<b>1600 €</b>

Taula 4. Hores de desenvolupament

### 9.4 Pressupost total

Suma total de totes les hores necessitades pel projecte més les amortitzacions.

Grup	Cost
Materials i amortitzacions	389.67€
Hores d'enginyeria	3970€
Hores de desenvolupament	1600€
<b>Total del projecte</b>	<b>5959.67 €</b>

Tabla 5. Pressupost total



## 10 Ampliacions

Aquesta aplicació pot ser ampliada amb moltes funcionalitats que poden funcionar perfectament amb l'àmbit de la cuina, però el que es farà es anar actualitzant l'aplicació varies vegades i en un període curt de temps, fent que cada cop surti una sola funcionalitat nova per a que l'usuari no es senti cohibit amb tantes funcionalitats i poder rebre el feedback de si realment l'usuari fa servir aquella opció o no.

Per tant, per a les pròximes versions de CookingEat, les funcionalitats que es desenvoluparan seran:

### *Adaptació a resolucions més grans i a tablets*

Com aquesta aplicació està de moment, adaptada només a una resolució de 480x800, no es pot aprofitar el tamany de les pantalles de mes pulçades i més resolució. També s'adaptarà l'aplicació per a que l'aplicació es pugui veure en format horitzontal.

### *Vidoreceptes*

Serà un nou concepte de creació de receptes a través del mòbil o la tablet. Es podran realitzar gravacions en real time del que la persona estigui cuinant i, un cop acabada la gravació, es pujarà a la web. Un cop pujada estarà disponible per a qualsevol usuari tant de web com de mòbil.

### *Buscador de receptes*

Buscador de receptes intel·ligent capaç de buscar tant per productes, tipus de menjar, horari de menjar etc... com fer recomanacions a partir del que s'ha cuinat anteriorment i els gustos de la persona.

### *Lectura per veu*

Lectura per veu de les receptes que es consultin per a que, quan s'està cuinant, no s'hagi d'estar pendent del dispositiu mòbil i es perdi la concentració del que s'està preparant a la cuina.



## 11 Conclusions

Un cop finalitzat el projecte, la conclusió principal és que s'ha aconseguit el principal propòsit d'aquest projecte, la connectivitat entre una pàgina web feta en Drupal i una aplicació en Android.

Tot i que es relativament sencilla, l'usuari al menys pot fer la visita de les diferents receptes que te la pàgina web per poder llegir les receptes de forma ràpida i simple desde el mòbil sense necessitat d'anar a la pròpia pàgina web, fent el procés més feixuc per la persona que visita la web.

Una característica que te és l'escalabilitat d'aquest projecte. L'inici d'aquest projecte es potser la part més complicada, ja que a partir d'aquí se li poden anar afegint noves funcionalitats de forma ràpida i simple.

Per últim la realització del projecte ha valgut la pena per els coneixements obtinguts d'Android, una tecnologia molt actual i Drupal un eina molt potent en tot el que estigui relacionat amb la web.



## 12 Bibliografia

- [1] Zigurd Mednieks, Laird Dornin, G.Blake Meike & Masumi Nakamura - *Programming Android* - O'REILLY **Septembre 2012**
- [2] Marko Gargenta - **Learning Android** - O'REILLY **Septembre 2012**
- [3] <http://market.android.com> - *Android Market* **Septembre 2012**
- [4] <https://github.com/aoltra/calculadora/blob/master/src/edu/uhuru/moviles/android/Calculadora/CalculadoraActivity.java> **Octubre 2012**
- [5] <http://drupalmodules.com/module/views-datasource> **Septembre 2012**
- [6] <http://www.jefflinwood.com/2010/08/using-drupals-views-as-a-json-web-service-with-the-rest-server/> **Septembre 2012**
- [7] <https://www.youtube.com/watch?v=tpdVqhAV0UU&feature=relmfu> **Octubre 2012**
- [8] <http://drupal.org/node/425944> **Septembre 2012**
- [9] <http://drupal.org/node/736522> **Septembre 2012**
- [10] <http://blog.hsl.virginia.edu/drupalpress/json-views-drupal-6-uva-health-sciences-web-services/> **Octubre 2012**
- [11] <http://www.beigerecords.com/joe/content/connect-android-drupal%E2%80%99s-json-server-module> **Novembre 2012**
- [12] <http://www.vogella.com/articles/AndroidListView/article.html> **Novembre 2012**
- [13] <http://androcode.es/2012/05/parseando-json-en-android/> **Novembre 2012**