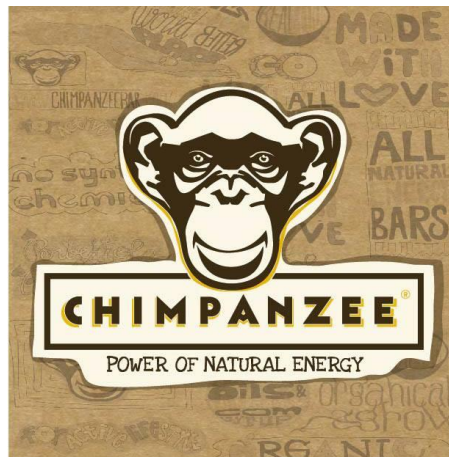




Universitat Oberta de Catalunya

Projecte final de carrera

Programació aplicacions mòbils usant HTML5



Projecte aplicació

CHIMPANZEE BAR

Autor: Daniel Solé Torrelles

Consultor: Carlos Sánchez Rosa

A la Mònica, ja que el seu suport i el seu exemple m'han ajudat constantment.

Als meus fills, en Quim i l'Àlicia. La vostra mirada i admiració m'han donat forces per a que estiguéssiu orgullosos de mi.

A tota la meva família que ha cregut en mi per poder assolir aquest objectiu.

Gràcies a en Carles Sánchez (tutor del projecte) per la seva orientació i guia en la elaboració d'aquesta memòria

INDEX

1. Introducció	5
2. Descripció de projecte.....	5
3. Objectiu	6
4. Funcionalitats	6
Catàleg de productes.....	6
Cercador GPS	6
Info botigues online.....	6
5. Tecnologies a usar	7
jQuery i jQuerymobile	7
WebSQL	7
Google Maps API	7
Phonegap(Cordova).....	7
6. Maquinaria	7
Ordinadors.....	7
Dispositius mòbils.....	8
7. Programari.....	8
MS Office	8
ProjectLibre.....	8
Notepad++	8
Paquet CorelDraw7x.....	8
Chrome	8
8. Planificació.....	8
9. Histories d'Usuari	10
10. Pantalles	10
Menu Inici	10

Menu persistent	11
Configuració.....	11
Catàleg	11
Producte	12
Botigues on-line.....	12
Botigues.....	12
11. Estructura de dades.....	13
Idioma	13
Producte	13
Ingredients.....	13
Valors Nutricionals.....	13
Botigues on-line.....	13
Botigues.....	13
Configuració.....	13
Diagrama Entitat-Relació	14
12. Arquitectura	14
13. Distribució dels arxius	16
14. Control de BBDD.....	17
Websqlinit.js	17
Websqlquery.js.....	18
15. Futures Millores	19
16. Bibliografia.....	20

1. Introducció

Aquest projecte de final de carrera es centra en el desenvolupament d'aplicacions per a dispositius mòbils. Ens basarem en el sistema multiplataforma HTML5 que usarà també CSS i javascript. En concret, en aquest projecte també usarem una BBDD basada en Sql i que serà local, webSql.

L'aplicació que desenvoluparem s'optimitzarà per a un sistema Android, tot i que canviant poques coses podria adaptar-se a iOS.

S'ha decidit desenvolupar aquest projecte amb la idea d'aprofitar la feina feta per a la empresa en la qual treballa. Aquesta eina ens ajudarà en termes de màrqueting. D'aquesta manera també aconseguim que el recorregut de l'aplicació no acabi un cop s'entregui el projecte. Addicionalment aconseguim que aquesta primera experiència arribi al públic final fent que així el feedback sigui molt més ampli.

2. Descripció del projecte

L'objectiu del projecte és efectuar l'anàlisi, disseny i implementació d'una aplicació multi-plataforma per a dispositius mòbils que ofereix diferents funcionalitats útils per als usuaris d'una marca de nutrició esportiva.

Per tant el primer pas és fer un pla de projecte que ens permeti seguir un calendari que ens faci arribar correctament a les dates claus del semestre tenint en compte totes les fases que implica desenvolupar tota la aplicació. Aquest primer pas formarà part de la PAC1

Seguirà l'anàlisi i el disseny de l'aplicació, on haurèm de deixar ben clar les funcionalitats i els passos a seguir al haver de desenvolupar l'aplicació usant les eines i els llenguatges de programació seleccionats. Tot això conformarà la PAC2

La PAC3 estarà formada per el desenvolupament efectuat basant-se en tot el preparat en la part de disseny. L'aplicació haurà d'estar acabada o pràcticament per a que en el darrer pas, la PAC4, es puguin fer les proves pertinents, els retocs necessaris i es puguin preparar tota la documentació requerida.

3. Objectiu

L'objectiu és crear una app de suport d'una marca de nutrició esportiva. Oferirà opcions que l'usuari pot trobar útils. Aquestes opcions en la seva majoria es poden veure a www.chimpanzeebar.com. Aquesta web és ben completa, l'app s'inspirarà en ella, però la seva distribució i el seu funcionament serà diferent.

La marca cerca amb aquesta app augmentar el valor de la mateixa i treballar més en la fidelització de l'usuari final. Per tant, les característiques ofertes han de ser útils per a l'usuari.

Per altre banda, l'objectiu de la marca és el mercat espanyol, però no descarta que la app es pugui usar en altres països per altres clients i distribuïdors. Així doncs, caldrà tenir en compte que en un futur es puguin anar afegint nous idiomes en properes actualitzacions.

4. Funcionalitats

Les principals funcionalitats del que es desenvoluparan en aquest projecte son:

Catàleg de productes: Un catàleg de productes àgil, on es puguin veure tots els productes de la marca amb les seves característiques. Dintre d'aquestes podem distingir els valor nutricionals, si son productes vegans, lliures de gluten, lliures de lactosa, etc. Addicionalment s'haurien de poder veure els ingredients de cada un dels productes per tal que de l'usuari pugui valorar si n'hi ha algun que pugui provocar al·lèrgies.

Cercador GPS: Es disposarà d'un posicionament GPS i es mostrarà la situació en el mapa de punts de venda que disposen de productes de la marca. Es valorarà la opció de mostrar un itinerari per arribar al punt de venda més proper.

Info botigues online: Les botigues online no estaran disponibles al cercador GPS, pel que es donarà la opció de mostrar un llistat de totes les botigues online que ofereixen el producte dintre del seu catàleg.

Al desenvolupar totes aquestes funcionalitats, caldrà tenir en compte que no només els desenvoluparà per a un sol idioma, si no que amb el temps els idiomes poden anar ampliant-se. Tan mateix, també caldrà tenir en compte que poden anar sorgint noves funcionalitats en el futur així que el disseny haurà de permetre la ampliació en qualsevol moment a base d'actualitzacions.

Cercarem sempre que la usabilitat sigui intuïtiva, cosa que les eines que usarem ens facilitarà.

5. Tecnologies a usar

Un cop sabem les funcionalitats de l'aplicació, podem trobar quines son les tecnologies que més bé s'adaptaran. Així doncs, sabem que basant-nos en HTML5+CSS+JavaScript, les eines que usarem son:

jQuery i jQuery mobile: És la llibreria de javascript més usada arreu. Disposa d'una quantitat generosa de utilitats per a adaptar la imatge de la nostra web-app. Associant-ho amb CSS i les noves característiques de HTML5 les possibilitats son incomputables.

WebSQL: Tecnologia SQL local. L'avantatge d'aquesta tecnologia és que podem usar els nostres coneixements en SQL per a manejar la BBDD que necessita l'aplicació. La informació s'enmagatzema localment, i és compatible en tots els sistemes actuals.

Google Maps API: Les API de Google maps ens permet usar tota la potencia de l'eina de Google a través de javascript, així podrem adaptar-ho a les necessitats de l'aplicació.

PhoneGap(Cordova): Compilador que prepararà l'executable en sistema Android. Hem de tenir clar que tot el que hem preparat s'ha de poder veure correctament en un navegador compatible HTML5. Tot i això, si ho volem transformar en una aplicació, la mateixa haurà de fer de navegador, això és del que s'encarrega phonegap.

6. Maquinari

Ordinadors:

Per motius de logística i laborals, el projecte l'aniré desenvolupant en diferents ordinadors. El gran avantatge del sistema en el que treballarem és que en gran mesura tot es pot treballar amb un editor de text senzill, així doncs no son necessàries grans instal·lacions de programari.

Casa: Intel Core i3 4GB Ram – Windows 7

Portatil: Intel Core i3 4GB Ram – Windows 7

Feina: AMD – Windows 7

Dispositius mòbils:

El dispositiu on faré les proves és el personal, però disposaré d'altres dispositius per fer proves i poder ajustar d'aquesta manera l'aspecte visual i comprovar la compatibilitat:

Dispositiu principal: LG G2 16gb

Altres dispositius: Sony Xperia M2, Sony Xperia , Samsung Galaxy pro, Samsung ACE, HTC nexus one

7. Programari

Tot el programari necessari per desenvolupar aquest projecte es:

MS Office: Paquet de programari que inclou editor de textos (MS Word), Editor de presentacions (MS PowerPoint) i Bases de dades (MS Access) d'on exportarem les dades en XML. També incorpora altres per facilitar el desenvolupament de la documentació corresponent al programari.

ProjectLibre: Programari opensource per organitzar la planificació i el diagrama de Gantt.

Notepad++: Editor de textos per desenvolupar el codi font tant en HTML, CSS o Javascript.

Paquet CoreDraw X7: Paquet gràfic tant de retoc com de disseny gràfic.

Chrom: Aquest Navegador de Google és ideal per a testejar el desenvolupament i anar-lo deputant.

8. Planificació

Per tal de portar la feina degudament programada en el temps, tenim distintes dates que son de recomanable complementació.

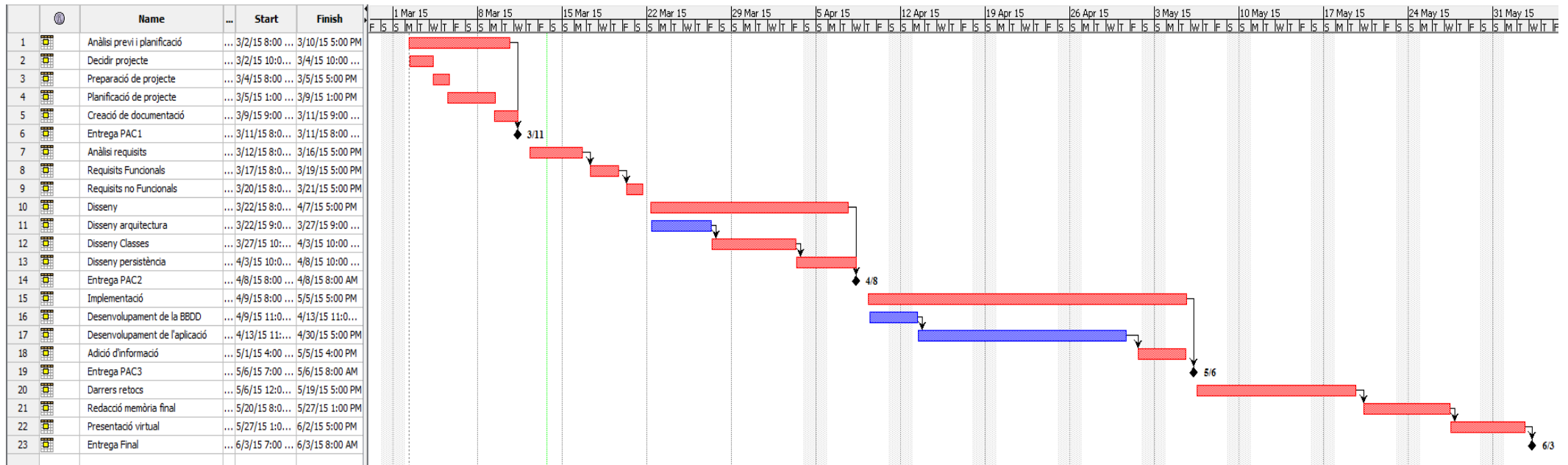
PAC1 – 11/03/2015 – Descripció i planificació de projecte

PAC2 – 08/04/2015 – Disseny de projecte

PAC3 – 06/05/2015 – Desenvolupament de projecte

PAC4 – 03/06/2015 – Entrega final de projecte i presentació

Seguint aquestes dates podem desenvolupar el següent diagrama de Gantt.



El projecte ens servirà per posar en pràctica tot allò treballat al llarg de la carrera, centrant-nos en

9. Histories d'Usuari

L'usuari pot seguir diferents camins. Exposarem una historia d'usuari on s'utilitzaran totes les funcionalitats. S'ha de tenir en compte que aquestes funcionalitats poden ser usades en l'ordre que es desitgi .

1er Seleccionar idioma de l'aplicació

2on Cercar en un mapa diferents localitzacions de botigues.

3re Veure un catàleg de productes classificats per Famílies

4rt Consultar un producte podent veure una imatge, les característiques més destacades, els ingredients i tots els seus valor nutricionals

5è Elegir una botiga on-line i accedir-hi

10. Pantalles

Les pantalles mostrades a continuació corresponen a les necessitats de l'App respecte a les seves funcionalitats i tenint en compte les Històries d'usuari.



Menu Inici:

Pantalla inicial on l'usuari podrà elegir l'opció a executar. Es disposarà de les opcions Productes, que obrirà el catàleg de productes; Botigues, que obrirà un mapa; On-line, que llistarà botigues on-line; I, Configuració, que permetrà opcions relacionades amb la App.



Menú persistent:

En la mayoría de las pantallas del sistema l'usuari podrà accedir directament a un menú persistent on es podran triar les mateixes opcions que en el menú d'inici. Estarà disponible en el Catàleg, configuració, mapa de botigues i el llistat de botigues on-line.



Configuració:

Es podrà triar l'idioma de l'aplicació. Això farà que totes les opcions del l'app i la informació del catàleg i tota la aplicació es mostri en l'idioma triat. Addicionalment, es podran veure els crèdits de l'aplicació i es podrà triar tancar l'aplicació i alliberar la memòria del sistema.



Catàleg:

El catàleg permetrà navegar per la llista de productes de la marca agrupats per famílies. Es podrà elegir la família y es mostraran els productes amb una petita imatge i el seus noms. Al elegir un producte s'obrirà una nova pantalla.



Producte:

On trobarem tota la informació pertinent al producte. Imatge, nom, descripció, ingredients, valor energètic. Haurà d'estar preparada per seguir navegant per el catàleg. Al menys per poder veure fàcilment tots els productes de la família.



Botigues On-line:

Aquest apartat mostrarà una sèrie de botigues on-line os es podrà adquirir els productes de la marca. L'usuari seleccionarà la botiga i dirigirà directament al usuari a la pagina web en qüestió, just a l'apartat on apareixen els productes.



Botigues:

Es mostrarà un mapa en el que es senyalitzaran les botigues que venen el producte de la marca. L'usuari es podrà posicionar en el mapa, veien d'aquesta manera quina és la botiga més propera. L'usuari podrà arribar a seleccionar una botiga i el sistema haurà de traçar un camí fins a la botiga.

11. Estructura de Dades

Les dades amb les que treballarà el sistema son només de consulta. En cap cas s'ha pensat en la possibilitat de que l'usuari pugui afegir dades pròpies o d'algun altre tipus al sistema. Així doncs haurem de tenir en compte 3 blocs de dades. Tot i això, les dades seran relacionals ja que hi haurà dades reutilitzades.

Idioma: Com s'ha presentat anteriorment el sistema ha de suportar diferents idiomes. La millor manera per poder gestionar-ho és a través d'una base de dades amb les diferents traduccions ben estructurades. Cada traducció estarà relacionada amb un idioma i amb un codi que l'identificarà.

Producte: Un producte consta d'una informació bàsica i única com poden ser la seu codi, quantitat de calories... però tindrà altes dades compartides com ara els tipus d'ingredients o els valors nutricionals.

Ingredients: Els ingredients son fonamentals per a informar del producte. S'informarà de l'ingredient i de la tipologia (Natural u orgànic).

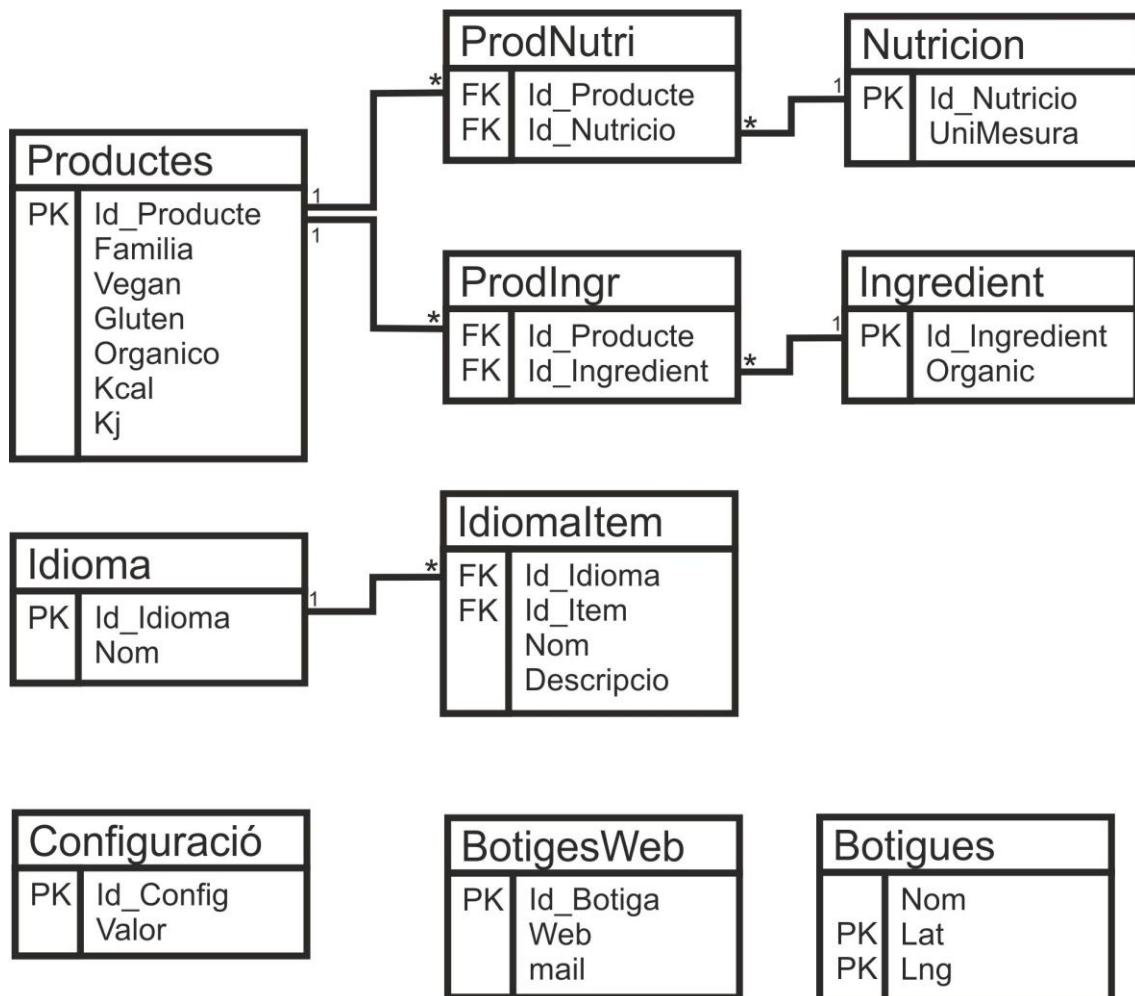
Valors nutricionals: Quan es parla de productes energètics es parla normalment de productes pensats per a ser consumits durant una activitat física. Per tant, cal informar correctament de totes les característiques del producte per que l'usuari sàpiga quan i com consumir-lo.

Botigues on-line: Serà la relació de botigues electròniques que venen els productes de la marca. Disposarà de toa la informació necessària per a poder enllaçar directament a la web en qüestió i adicionalment del contacte mail.

Botigues: Aquesta informació no es trobarà dintre de la web, si no que estarà disponible al servidor Google i l'usarem a través de la API gloogle Maps. Per tant, es crearà externament però no estarà gestionada per l'app.

Configuració: On mantindrem les dades imporants de l'aplicació, com la versió actual, o bé l'idioma actiu.

Diagrama Entitat-relació:



12. Arquitectura

Des de el primer moment tinc com a objectiu desenvolupar una aplicació ben estructurada i basada en MVC (Vista-Controlador-Model). Però les meves ganes d'aprendre a utilitzar webSql ho han impedit.

WebSql és un sistema de base de dades asíncron que s'executa un cop que tota la informació de la pàgina s'ha executat. Això no ens permet usar informació obtinguda en les consultes fora de la funció que ha fet la crida. L'únic que podem arribar a fer per usar les dades és passar-la a elements d'Html. És per això que he hagut de desenvolupar la i estructurar d'una manera diferent, però basant-me en el MVC.

He preparat totes les vistes en Html, dintre del codi es faran crides als a funcions que modificaran la informació dels elements corresponents. Les crides es faran en javascript a

funcions preparades en arxius locals. Per tant, totes les cerques a bases de dades es faran de forma local, que és el que cercava des de el principi, i asíncrona.

Per altre banda, només en 2 casos necessitarem connexió externa. La més important per la aplicació en quan, a través de *gmaps.js* sol·licitem veure el mapa en la nostra aplicació. Totes les dades de les botigues les trobem de manera local, però les imatges del mapa les cerca *gmaps* a través de la web.

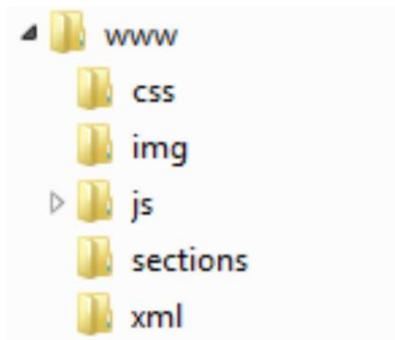
Finalment, també caldrà connexió externa quan l'usuari vulgui accedir a alguna de les botigues on-line que llistem en la secció corresponent. Com he comentat, no és necessari per el correcte funcionament de l'aplicació, però si per a permetre que l'usuari pugui continuar amb la seva experiència.

Tot això, doncs, queda mostrat en el següent diagrama.



13. Distribució dels arxius

Una aplicació preparada en html per a cordova ha de mantenir una distribució de arxius mínima. Tots els arxius *html*, *frameworks*, *javascript*, imatges i demés ha de estar contingut dintre del directori *www* del projecte creat en cordova o phonegap. Adicionalment, hem de saber que el primer arxiu que s'obrirà serà el *index.html*. A partir d'aquí som lliures de crear la distribució d'arxius que desitgem, però en el meu cas he decidit seguir la la distribució marcada per cordova. Adicionalment he creat els directoris *sections* i *xml*:



Cada un dels directoris té una funció.

-css: On posarem els arxius que modifiquin l'estil de la web, o aplicació en aquest cas.

-img: Totes les imatges que s'usaran en l'aplicació. En aquest cas totes les imatges de productes, els logotips, i altres.

-js: És el directori on s'instal·len els frameworks. En aquest cas *jquerymobile* i *gmaps*. Però també es troben els arxius necessaris que he desenvolupat per a poder utilitzar la base de dades. De fet, en aquest directori hi hauria d'haver tots els arxius codificats en javascript.

-sections: Totes les seccions que he preparat es troben aquí en arxius *html*.

-xml: El primer cop que la aplicació s'iniciï cercarà les dades dels arxius xml per posar al dia la nostra webSql.

14. Control de la BBDD

El treball que es du a terme a l'aplicació amb la base de dades es limita a la creació de taules, i la inserció, modificació i consulta d'aquestes. He creat dos arxius per a fer tot aquest treball.

Websqlinit.js

Aquest arxiu conté les funcions que crearan les taules i que introduiran les dades, o les modificaran.

Per començar cal crear una instància de crida a la base de dades amb la que treballarem:

```
var chimpdb = openDatabase('cbapp', '1.0', 'DB Chimpanzee Bar App', 2 * 1024 * 1024);
```

La funció de execució Sql:

```
chimpdb.transaction(function (tx) {  
    tx.executeSql('CREATE TABLE Languages (Id_Language PRIMARY KEY,  
    Language)');  
});
```

En aquest cas creem una taula de nom *Languages* amb els camps *id_Language*, que serà la clau primària, i *Language*. Com es pot veure, el codi fa una crida a un comandament Sql.

Aquesta serà la manera que usarem sempre per a executar totes les crides al websql.

En aquest arxiu no només es troba el codi per crear totes les taules, si no que també omplirem totes aquestes amb informació. Però primer hem d'agafar la informació que està en xml i tractar-la per poder inserir-la.

Una funció tipus que agafa les dades i les transmet a una funció per a que les insereixi a la base de dades és:

```
function getXMLIngredients(){  
  
    xmlhttp=new XMLHttpRequest();  
    xmlhttp.open("GET","xml/Ingredients.xml",false);  
    xmlhttp.send();  
    xmlDoc=xmlhttp.responseXML;  
  
    var data =xmlDoc.getElementsByTagName("Ingredients");
```

```
for(var i=0;i<data.length;i++)
{
    idind=data[i].getElementsByTagName("Id_ingredient")[0].textContent;
    org=data[i].getElementsByTagName("Organic")[0].textContent;

    insertTableIngredients(idind, org);

}
}
```

Aquesta funció obre l'arxiu xml i en fa un repàs de tots els seus elements, per cada un dels elements, passa per paràmetre les dades a una funció que les insertarà a la taula corresponent.

Websqlquery.js

Correspon a l'arxiu que efectuarà totes les consultes de dades sobre la base de dades. Un exemple de funció que podem trobar-hi és:

```
function getItemName(IdItem){
    chimpdb.readTransaction(function (tx) {
        tx.executeSql('SELECT LanguageName FROM LanguageItem WHERE
Id_Language=(SELECT CValue FROM Config WHERE Id_Config="Language") AND
Id_Item="'+IdItem+'"', [], function (tx, results) {
            results.rows.item(0).CValue);
            ret=results.rows.item(0).LanguageName;
            console.log("Language -->" + ret);
            document.querySelector('#'+IdItem).innerHTML = ret;
        });
    });
}
```

Com podem veure en aquest exemple es fa una crida de lectura a la base de dades on executem una consulta. En aquesta consulta hi podem afegir variables rebudes per paràmetre.

Sent conscients de les limitacions de webSql, al no poder retorna dades, l'única opció que ens queda és modificar els elements del html, tal com es veu en l'exemple.

15. Futures millores

L'aplicació pot ser més complerta del que ara és, pel que es poden dur a terme les següents millores

-Interfície gràfica: He experimentat amb el css de jquerymobile, però m'ha donat certs problemes al treballar amb els elements html. Per això caldria o bé canviar de sistema dinàmic de dades per tal de no treballar tant directament amb els elements html, o bé millorar i personalitzar el css.

-*Migrar a PHP+Mysql*: Al començar la aplicació, i després de la experiència personal d'usar un terminal mòbil amb molts problemes de connexió, vagi decidir que els usuaris de l'aplicació quan no tinguessin access a la xarxa poguessin seguir usant-la sense problemes. Però aquesta decisió ha afectat a l'hora de desenvolupar l'aplicació aportant certes limitacions. Així doncs, crec que el més aconsellable seria permetre una aplicació amb connexió constant que pugui gaudir de tot el potencial d'altres eines com el Php+Mysql.

-*Implementar gestió d'actualitzacions*: El Catàleg de productes, o el llistat de clients no és quelcom immòbil, pel que caldrà anar actualitzant les bases de dades. Si finalment no s'efectua la migració a PHP+Mysql, caldrà pensar en com aplicar-ho.

-*Llistar webs per idiomes*: Per ara el llistat de botigues online és fixe, i estiguis usant l'idioma que estiguis usant sempre apareixerà la mateixa llista. Això és un problema, ja que un usuari de llengua anglesa entrarà a una web de llengua castellana i farà que perdi el temps. Així doncs, hauré d'ajustar aquesta utilitat per a que les webs es mostrin segons idiomes.

-*Millorar part Gmaps*: Crec que no n'hi ha prou en mostrar totes les botigues que venen el producte en el mapa. Cal poder posicionar on està l'usuari i addicionalment establir un camí des de on es troba l'usuari fins a la botiga seleccionada.

-*Adaptar aplicacions a altres dispositius*: Phonegap ens permet compilar les aplicacions per a diferents dispositius, però disposa de algunes crides especials per a certs sistemes operatius, com per exemple l'obertura de finestres externes. Així doncs, si volem portar l'aplicació de android a ios, caldrà controlar aquestes crides i modifica-les

16. Bibliografia

-Jquermobile

< <http://www.jquermobile.com>>

-Cordova

< <http://cordova.apache.org>>

-Phonegap

< <http://www.phonegap.com>>

-Gmaps API

< <https://developers.google.com/maps/?hl=es>>

-Stackoverflow – Foro de dubtes de programació

< <https://stackoverflow.com/>>

-w3schools – Tutotials i exemples

< http://www.w3schools.com/html/html5_intro.asp>

-Chimpanzee Bar

< <http://www.chimpanzeebar.com>>