

**PFC - Entrega 20120612\_v3**

# **Projecte iURV**

**Memòria PFC**



**Consultor: Jordi Ceballos Villach**

**Alumne: Nuri Bordas i Garcia**

**Data entrega: 12/06/2012**

## Control del document

| Revisions del document |        |  |
|------------------------|--------|--|
| Data                   | Versió | Descripció   |
| 12/03/2012             | V0     | Pla de treball   |
| 09/04/2012             | V1     | Anàlisi, Disseny i Prototip  |
| 10/05/2012             | V2.0   | Correcció apartat Disseny segons proposta consultor: <ul style="list-style-type: none"><li>- Correccions apartat 5.1 de comparativa d'alternatives de desenvolupament</li><li>- Incorporació a l'apartat 5.2 de l'arquitectura lògica</li><li>- Ampliació apartat 5.3</li><li>- Nou apartat 5.4 Diagrames de seqüència</li></ul> |
| 21/05/2012             | V2.1   | S'afegeix la descripció de la implementació de la solució desenvolupada  |
| 12/06/2012             | V3.0   | Correccions a l'entrega anterior i complementació document PFC.  |

## Contingut

|           |   |    |
|-----------|---|----|
| 1         | Definició general del projecte .....                        | 6  |
| 1.1       | Justificació i context del projecte .....                   | 6  |
| 1.2       | Marc de referència .....                                    | 6  |
| 1.3       | Terminis establerts .....                                   | 8  |
| 2         | Estructura del projecte i descomposició en activitats ..... | 10 |
| 2.1       | Estructura del projecte .....                               | 10 |
| 2.2       | Descomposició en activitats .....                           | 10 |
| 2.2.1     | Detall de les activitats .....                              | 10 |
| 2.2.1.1   | Inici del projecte .....                                    | 10 |
| 2.2.1.2   | Estudi de viabilitat .....                                  | 11 |
| 2.2.1.3   | Anàlisi .....   | 11 |
| 2.2.1.4   | Disseny .....   | 11 |
| 2.2.1.5   | Implementació i proves .....                                | 12 |
| 2.2.1.6   | Finalització .....  | 12 |
| 2.2.2     | Temporització .....   | 13 |
| 2.3       | Eines utilitzades .....                                     | 15 |
| 2.4       | Entregables de la solució .....                             | 15 |
| 3         | Anàlisi del sistema .....                                   | 16 |
| 3.1       | Requeriments inicials .....                                 | 16 |
| 3.2       | Requeriments no funcionals .....                            | 17 |
| 3.3       | Diagrames de casos d'ús .....                               | 17 |
| 3.3.1     | Menú general .....  | 17 |
| 3.3.2     | Consultar novetats .....                                    | 18 |
| 3.3.3     | Consultar notícies .....                                    | 18 |
| 3.3.4     | Consultar Activitats .....                                  | 19 |
| 3.3.5     | Consultar persones .....                                    | 20 |
| 3.3.6     | Localitzar centres .....                                    | 21 |
| 4         | Disseny .....   | 22 |
| 4.1       | Alternatives desenvolupament .....                          | 22 |
| 4.1.1     | Aplicació nativa .....                                      | 22 |
| 4.1.2     | Aplicació web + contenidor .....                            | 23 |
| 4.1.2.1   | Contenidor: PhoneGap .....                                  | 23 |
| 4.1.2.2   | Avaluació frameworks .....                                  | 28 |
| 4.1.2.2.1 | jQueryMobile .....  | 28 |
| 4.1.2.2.2 | SenchaTouch .....   | 30 |
| 4.1.2.2.3 | Comparativa jQueryMobile vs SenchaTouch .....               | 32 |
| 4.1.3     | Solució escollida: PhoneGap + jQueryMobile .....            | 33 |
| 4.2       | Arquitectura de la solució .....                            | 35 |
| 4.2.1     | Arquitectura física .....                                   | 35 |
| 4.2.2     | Arquitectura lògica .....                                   | 36 |
| 4.3       | Altres aspectes tecnològics .....                           | 38 |
| 4.3.1     | SGBD .....  | 38 |
| 4.3.2     | Accés a les dades .....                                     | 39 |
| 4.3.3     | XML corporatius .....                                       | 40 |
| 4.3.4     | Mapes .....   | 41 |

|         |   |    |
|---------|---|----|
| 4.4     | Diagrames de seqüència .....                | 41 |
| 4.4.1   | Localitzar centres .....                    | 42 |
| 4.4.2   | Consultar persones.....                     | 43 |
| 4.4.3   | Consultar novetats .....                    | 44 |
| 5       | Implementació .....                         | 45 |
| 5.1     | Contenedor aplicació: PhoneGap.....         | 45 |
| 5.2     | Capa de presentació.....                    | 47 |
| 5.2.1   | Visió general jQuery Mobile .....           | 47 |
| 5.2.2   | Component menú .....                        | 48 |
| 5.2.3   | Component news .....                        | 49 |
| 5.2.4   | Component agenda .....                      | 50 |
| 5.2.5   | Component newsFoto .....                    | 50 |
| 5.2.6   | Component seleccioPersones .....            | 51 |
| 5.2.7   | Component seleccioCentres .....             | 53 |
| 5.2.8   | Component informaciInstitucional .....      | 54 |
| 5.2.9   | Component accés URV .....                   | 55 |
| 5.3     | Capa de negoci .....                        | 56 |
| 5.3.1   | Visió general PHP .....                     | 56 |
| 5.3.2   | Component Gestió Notícies .....             | 57 |
| 5.3.3   | Component Gestió Persones.....              | 57 |
| 5.3.4   | Component Gestió Centres .....              | 58 |
| 5.3.5   | Component Gestió Institucions.....          | 59 |
| 5.4     | Capa de dades .....                         | 60 |
| 5.5     | Implementació pantalles de l'aplicació..... | 62 |
| 6       | Conclusions .....                           | 73 |
| ANNEX 1 | – Glosari .....                             | 74 |
| ANNEX 2 | – Fonts consultades .....                   | 76 |

## Taula d'il·lustracions

|   |    |
|---|----|
| Il·lustració 1. Aplicacions d'exemple .....                       | 7  |
| Il·lustració 2. Cas d'ús Menú General.....                        | 17 |
| Il·lustració 3. Cas d'ús Consultar novetats .....                 | 18 |
| Il·lustració 4. Cas d'ús Consultar notícies.....                  | 18 |
| Il·lustració 5. Cas d'ús Consultar activitats.....                | 19 |
| Il·lustració 6. Cas d'ús Consultar persones.....                  | 20 |
| Il·lustració 7. Cas d'ús Localitzar centres.....                  | 21 |
| Il·lustració 8. Esquema MVC.....                                  | 22 |
| Il·lustració 9. Compatibilitats PhoneGap.....                     | 24 |
| Il·lustració 10. Estructura projecte XCode.....                   | 27 |
| Il·lustració 11. Selecció template XCode .....                    | 28 |
| Il·lustració 12. Codi + jQueryMobile.....                         | 29 |
| Il·lustració 13. Codi + SenchaTouch .....                         | 31 |
| Il·lustració 14. Simulador entorn desenvolupament iOS .....       | 34 |
| Il·lustració 15. Arquitectura física .....                        | 35 |
| Il·lustració 16. Arquitectura lògica .....                        | 36 |
| Il·lustració 17. Diagrama d'events jQueryMobile.....              | 37 |
| Il·lustració 18. Diagrama execució jQueryMobile .....             | 38 |
| Il·lustració 19. Esquema taules BD .....                          | 39 |
| Il·lustració 20. Diagrama seqüència Localitzar centres .....      | 42 |
| Il·lustració 21. Diagrama seqüència Consultar persones .....      | 43 |
| Il·lustració 22. Diagrama seqüència Consultar novetats.....       | 44 |
| Il·lustració 23. Selecció template PhoneGap.....                  | 45 |
| Il·lustració 24. Estructura de les taules .....                   | 61 |
| Il·lustració 25. Pantalla principal aplicació .....               | 62 |
| Il·lustració 26. Pantalla Novetats .....                          | 63 |
| Il·lustració 27. Pantalla Activitats .....                        | 64 |
| Il·lustració 28. Pantalla Notícies.....                           | 65 |
| Il·lustració 29. Detall notícia.....                              | 66 |
| Il·lustració 30. Pantalla introducció criteris de cerca .....     | 67 |
| Il·lustració 31. Pantalla Llista de persones .....                | 68 |
| Il·lustració 32. Pantalla Detall d'informació de la persona ..... | 69 |
| Il·lustració 33. Llistat de centres .....                         | 70 |
| Il·lustració 34. Mapa d'un centre .....                           | 71 |
| Il·lustració 35. Informació institucional.....                    | 72 |
| Il·lustració 36. Detall dades institucionals.....                 | 72 |

## 1 Definició general del projecte

### 1.1 Justificació i context del projecte

A l'hora de prendre la decisió de quin projecte es volia realitzar, es va tenir clar que l'objectiu era provar de desenvolupar una aplicació per a iPhone o iPad. Avui en dia gairebé tothom té un iPhone o un iPad o un iPod... i no deixa de ser sorprenent com, inclús els nens, utilitzen les aplicacions que estan desenvolupades per a aquest dispositius amb molta facilitat i naturalitat.

Inicialment es va plantejar, a l'hora de sol·licitar l'àrea del projecte, que es desenvoluparia una aplicació educativa per a nens, però posteriorment, a causa de la revisió de projectes estratègics per a la Universitat Rovira i Virgili (a partir d'ara URV), s'ha vist que es podria aprofitar aquest projecte per a fer un pas inicial i incorporar a aquest canal la URV i acabar definint quins es vol que siguin els estàndards per al desenvolupament de determinades aplicacions per a mòbil.

### 1.2 Marc de referència

El projecte iURV neix com un pas endavant per a la Universitat Rovira i Virgili per apropar la universitat als estudiants a través de les noves tecnologies. Es pretén crear una aplicació per a ser utilitzada des del mòbil que contingui diferents funcionalitats per als estudiants: localització dels campus, directori de persones, agenda... i amb capacitat de ser estesa més endavant.

Si fem un cop d'ull a l'àmbit de les universitats veiem que aquestes, a poc a poc, van desenvolupant les seves aplicacions per estar presents en aquests nous canals. Les universitats espanyoles també estan (en menor mesura que les americanes) començant a fer els seus desenvolupaments. La majoria d'aquestes universitats disposen d'aplicacions que un cop arrenquen ofereixen l'accés a diferents funcionalitats, de manera que per cada funcionalitat que es vol donar hi ha una icona a la pantalla inicial.

Tot seguit s'adjunten les captures de les aplicacions de Stanford i de Georgia:



Il·lustració 1. Aplicacions d'exemple

S'utilitzarà aquest estil d'aplicació per al desenvolupament del projecte.

Les funcionalitats que ofereix l'aplicació, són les següents:

- Notícies. Accés a les darreres notícies de la universitat. Es mostra un llistat amb les darreres notícies de la universitat i un cop se'n selecciona una se'n pot accedir al detall. En l'actualitat el servei de notícies de la universitat no disposa de cap mètode que entregui les darreres notícies, de manera que ha calgut fer un desenvolupament propi per a obtenir les dades.
- Localitzar centre. Permet ubicar en un mapa els diferents centres i facultats de que disposa la universitat. Es mostra la llista de centres i un cop el selecciones es visualitza el mapa d'on està ubicat a partir de les seves coordenades.
- Accés al directori de persones. Aquesta funcionalitat permet localitzar les persones donades d'alta al directori de la universitat. Al no disposar de cap servei de la universitat que proporcioni aquesta informació, s'ha creat una base de dades local de manera que es puguin consultar persones "fictícies".
- Agenda d'actes a la URV. Funcionalitat que ofereix la llista dels darrers actes publicats a l'agenda corporativa.
- Novetats de la URV. Funcionalitat que ofereix la llista de les darreres novetats publicades.
- Informació institucional. Aprofitant l'estructura de dades que s'ha hagut de crear s'ha afegit una funcionalitat per mostrar la informació classificada institucionalment.
- Accés a la web de la URV. Es tractarà d'una simple icona que proporcionarà accés a la web de la URV.

El que es tenia clar al moment de començar el projecte era per a quina plataforma es volia desenvolupar, iPhone, però es va plantejar una fase prèvia al desenvolupament per decidir si l'aplicació és desenvolupava en codi natiu per als dispositius comentats o bé es feia ús d'algun framework open source que permetés desenvolupar l'aplicació i fer-la compatible amb altres dispositius. En l'actualitat hi ha un gran debat obert entre els que consideren millor desenvolupar aplicacions natives i els que pensen que són millors les aplicacions híbrides implementades amb altres tecnologies HTML5, CSS, jQuery...

Com es veurà en l'apartat d'anàlisi, finalment l'opció escollida, ha descartat el desenvolupament de l'aplicació amb codi natiu, ja que per aquest tipus d'aplicació, s'ha considerat que l'ús de tecnologies més esteses com HTML5, CSS..., facilita la feina. És més fàcil trobar milers de planes amb informació d'aquestes tecnologies, que no pas fer-ho d'Objective-C, tot i que no cal oblidar que l'entorn de desenvolupament d'Apple està molt aconseguit. El fet de desenvolupar en Objective-C fa que siguin necessaris perfils molt especialitzats que per aquest tipus de projectes no semblen necessaris.

L'elecció del tipus de solució, també ha de servir per a establir els estàndards de desenvolupament d'aplicacions per a mòbils i altres dispositius per a la universitat.

Per aquest motiu, també s'ha fet l'anàlisi d'utilitzar algun framework existent per a facilitar la feina i la solució final ha sigut utilitzant un d'aquests frameworks.

Un dels riscos identificats a l'inici del projecte ha estat que hi ha determinats serveis que haurien de cobrir les funcionalitats ofertes, i que actualment no estan disponibles a la universitat. Per aquest motiu, ha estat necessari crear determinats serveis que, a priori, es pretenia que estiguessin disponibles.

### 1.3 Terminis establerts

El desenvolupament d'aquest projecte ve marcat per els terminis establerts a nivell de la pròpia assignatura per tal de portar-lo a terme.

| Calendari tasques           |            |            |
|-----------------------------|------------|------------|
| Descripció tasca            | Data inici | Data fi    |
| Pla de treball              | 01/03/2012 | 12/03/2012 |
| Anàlisi, Disseny i Prototip | 13/03/2012 | 09/04/2012 |
| Implementació               | 10/04/2012 | 21/05/2012 |
| Entrega final               | 22/05/2012 | 12/06/2012 |

**Fase Pla de treball:** és la fase en que es tracta de definir els objectius del projecte, les funcionalitats que s'oferiran, els terminis, els riscos... (Doc. nbordas\_PlaTreball.pdf)

**Fase Anàlisi, Disseny i Prototip:** fase en la que s'entregarà l'anàlisi (en el que s'haurà de prendre la decisió de quin tipus de solució volem desenvolupar), el disseny i un prototip amb la interfície gràfica de l'aplicació final. (Doc. nbordas\_PAC2\_AnalisiDissenyPrototip.pdf)

**Fase d'Implementació:** fase final en la que es portarà a terme el desenvolupament de l'aplicació. El resultat d'aquesta fase és l'entrega de l'aplicació final en funcionament. (Doc. nbordas\_PAC3\_Implementació.pdf)



**Entrega final:** fase durant la que es realitzarà la memòria final del projecte i la presentació per tal de ser exposada. (Doc. Nbordas\_Memoria.pdf).

Durant les diferents fase plantejades s'ha anat generant la documentació que ha sigut entregada en les diferents fites establertes en el pla de projecte, els continguts dels quals han estat condensats en aquesta memòria final de projecte.

## 2 Estructura del projecte i descomposició en activitats

### 2.1 Estructura del projecte

La metodologia que s'ha utilitzat en el desenvolupament del projecte és la de cicle de vida en cascada.

### 2.2 Descomposició en activitats

D'acord amb els estàndards de projectes de construcció de software les activitats que formen part d'aquest projecte són les següents:

| Descomposició estructural d'activitats |                           |                                  |
|--|---------------------------|----------------------------------|
| Codi d'activitat                       | Activitat nivell 1        | Activitat nivell 2               |
| 01                                     | Inici del projecte        |                                  |
| 02                                     | Construcció del software  |                                  |
| 02.01                                  |                           | Estudi de viabilitat             |
| 02.02                                  |                           | Anàlisi                          |
| 02.03                                  |                           | Disseny                          |
| 02.04                                  |                           | Implementació i proves unitàries |
| 02.05                                  |                           | Proves                           |
| 03                                     | Entrega Final de Projecte |                                  |

#### 2.2.1 Detall de les activitats

##### 2.2.1.1 Inici del projecte

En aquest fase s'ha intentat tenir una visió global del sistema a crear, així com establir una guia de les activitats a portar a terme al llarg del temps.

| Inici del projecte        |  |
|---------------------------|--|
| Activitat                 | Descripció   |
| Definició del projecte    | Conceptualització del projecte a realitzar, a nivell de requeriments i tecnologia a utilitzar. |
| Planificació del projecte | Establir terminis del projecte conforme a les fites de l'assignatura                           |

|                      |  |
|----------------------|--|
| Creació del document | Generació del Pla de Treball                         |
| Enviament PAC1       | Enviament del document corresponent a aquesta etapa. |

### 2.2.1.2 Estudi de viabilitat

En aquesta fase s'han analitzat les funcionalitats definides inicialment i s'han ampliat amb les proposades com a funcionalitats a estendre.

| Estudi viabilitat                             |   |
|---|---|
| Activitat                                     | Descripció  |
| Definició de les funcionalitats a implementar | Revisió i ampliació de les funcionalitats que ha d'oferir l'aplicació |

### 2.2.1.3 Anàlisi

Durant aquesta fase s'han detallat les necessitats a cobrir per l'aplicació tant des del punt de vista funcional com des del punt de vista operacional.

Les principals activitats han sigut:

| Anàlisi                                     |   |
|---|---|
| Activitat                                   | Descripció  |
| Especificació de requeriments funcionals    | Creació i documentació dels casos d'ús                        |
| Especificació de requeriments no funcionals | Identificació i documentació dels requeriments no funcionals. |

### 2.2.1.4 Disseny

En aquesta fase s'ha definit com s'haurà d'implementar la solució per a cobrir les necessitats i objectius plantejats en les fases anteriors, incloent aspectes concrets de tecnologia.

Les activitats realitzades han estat les següents:

| Disseny                           |  |
|-----------------------------------|--|
| Activitat                         | Descripció   |
| Estudi alternatives tecnològiques | Analitzar alternatives de desenvolupament i presa de decisió de l'arquitectura de la solució |

|                              |   |
|------------------------------|---|
| Disseny arquitectònic        | Disseny de l'arquitectura de la solució   |
| Disseny interfícies d'usuari | Creació de prototipus d'interfícies d'usuari per les diferents funcionalitat.             |
| Preparació del document      | Recopilació de la informació i preparació del document d'anàlisi i disseny de l'aplicació |
| Entrega PAC2                 | Enviament de la PAC2.   |

### 2.2.1.5 Implementació i proves

Durant aquesta fase s'ha fet el desenvolupament de l'aplicació i dels serveis necessaris per a realitzar les funcionalitats definides, així com les proves necessàries per a comprovar-ne el correcte funcionament.

Les activitats realitzades han sigut:

| Disseny                          |  |
|----------------------------------|--|
| Activitat                        | Descripció   |
| Desenvolupament capa presentació | Implementació dels components que componen aquesta capa: <ul style="list-style-type: none"> <li>• Planes</li> <li>• JS</li> <li>• CSS</li> </ul> i proves. |
| Desenvolupament capa de negoci   | Desenvolupament programes PHP accés a la base de dades i proves.   |
| Definició capa de dades          | Definició de les taules per donar suport a l'aplicació.  |
| Proves integrades                | Prova de tota la solució   |
| Preparació del paquet entregable | Confecció del document d'instruccions d'instal·lació, del documenta d'implementació i creació del paquet de software.                                      |

### 2.2.1.6 Finalització

Seria la fase de pas a producció de la solució, que en aquest cas es correspon a les tasques següents:

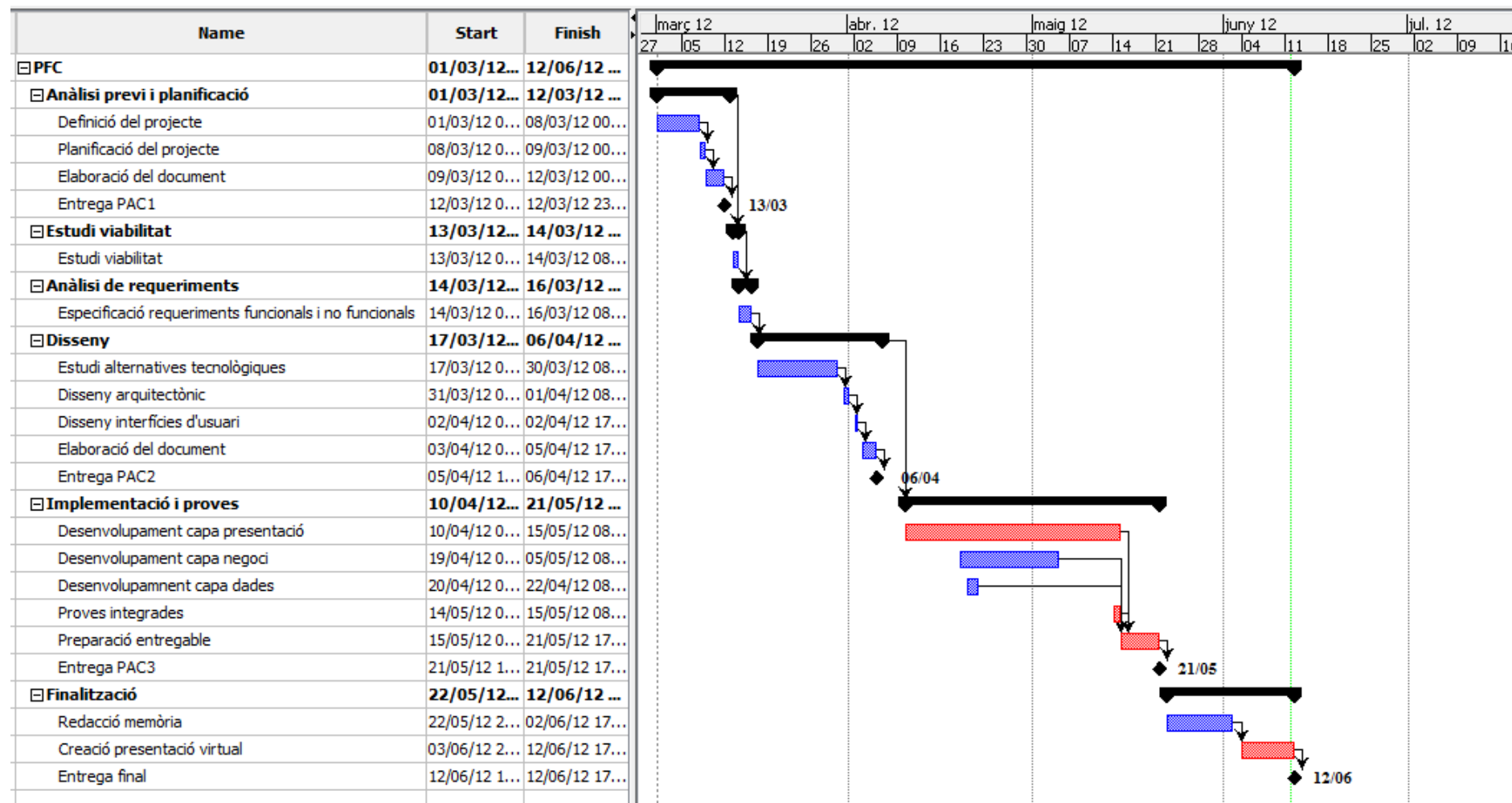
| Anàlisi                        |  |
|--------------------------------|--|
| Activitat                      | Descripció   |
| Redacció memòria final         | Recopilació de documents i confecció de la memòria final |
| Creació de presentació virtual | Creació de la presentació del projecte                   |
| Entrega final                  | Entrega de la documentació                               |

## 2.2.2 Temporització

Tot seguit es mostra el temps destinat a cadascuna de les activitats:

| Calendari tasques  |         |            |            |
|--|---------|------------|------------|
| Descripció tasca   | Dies    | Data inici | Data fi    |
| <b>Anàlisi previ i planificació</b>                          | 12 dies | 01/03/2012 | 12/03/2012 |
| <b>Definició del projecte</b>                                | 7 dies  | 01/03/2012 | 07/03/2012 |
| <b>Planificació del projecte</b>                             | 1 dia   | 08/03/2012 | 08/03/2012 |
| <b>Elaboració del document</b>                               | 3 dies  | 09/03/2012 | 11/03/2012 |
| <b>Entrega PAC1</b>  | Fita    | 12/03/2012 |            |
| <b>Estudi viabilitat</b>                                     | 1 dia   | 13/03/2012 | 13/03/2012 |
| <b>Anàlisi de requeriments</b>                               | 3 dies  | 14/03/2012 | 16/03/2012 |
| <b>Especificació requeriments funcionals i no funcionals</b> | 3 dies  | 14/03/2012 | 16/03/2012 |
| <b>Disseny</b>   | 20 dies | 17/03/2012 | 05/04/2012 |
| <b>Estudi alternatives tecnològiques</b>                     | 14 dies | 17/03/2012 | 30/03/2012 |
| <b>Disseny arquitectònic</b>                                 | 2 dies  | 31/03/2012 | 01/04/2012 |
| <b>Disseny interfícies d'usuari</b>                          | 1 dia   | 02/04/2012 | 02/04/2012 |
| <b>Elaboració del document</b>                               | 3 dies  | 03/04/2012 | 05/04/2012 |
| <b>Entrega PAC2</b>  | Fita    | 05/04/2012 |            |
| <b>Implementació i proves</b>                                | 42 dies | 10/04/2012 | 21/05/2012 |
| <b>Desenvolupament capa presentació</b>                      | 25 dies | 10/04/2012 | 15/05/2012 |
| <b>Desenvolupament capa negoci</b>                           | 6 dies  | 19/04/2012 | 05/05/2012 |
| <b>Desenvolupament capa dades</b>                            | 3 dies  | 20/04/2012 | 22/04/2012 |
| <b>Proves integrades</b>                                     | 2 dies  | 14/05/2012 | 15/05/2012 |
| <b>Preparació entregable</b>                                 | 6 dies  | 16/05/2012 | 21/05/2012 |
| <b>Entrega PAC3</b>  | Fita    | 21/05/2012 |            |
| <b>Finalització</b>  | 22 dies | 22/05/2012 | 12/06/2012 |
| <b>Redacció memòria</b>                                      | 12 dies | 22/05/2012 | 02/06/2012 |
| <b>Creació presentació virtual</b>                           | 10 dies | 03/06/2012 | 12/06/2012 |
| <b>Entrega final</b>   | Fita    | 12/06/2012 |            |

Diagrama de Gantt de la planificació i tasques del projecte:



## 2.3 Eines utilitzades

Per a la realització del PFC s'han utilitzat les següents eines:

- Xcode v4.2.1, entorn de desenvolupament per a Mac OS.
- MySQL per a disseny de la base de dades.
- Firefox / Chrome per a proves en entorn web.
- SmartDraw per elaboració de casos d'ús i altres gràfics.
- Microsoft Word / Open Office Writer per a l'elaboració dels documents.
- Microsoft Power Point / Open Office Impress per a l'elaboració de la presentació.
- OpenProj per a l'elaboració de la planificació del projecte.

## 2.4 Entregables de la solució

Durant la realització del present PFC s'han realitzat els següents entregables:

- Document del Pla de Treball
- Document d'Anàlisi i Disseny i Prototipus del Sistema.
- Aplicació
- Manual d'instal·lació de l'aplicació
- Memòria final del projecte
- Presentació virtual

## 3 Anàlisi del sistema

### 3.1 Requeriments inicials

En les fases inicials del projecte, s'han perfilat les funcionalitat que havia de tenir l'aplicació (inicialment, s'havia proposat una versió més reduïda de l'aplicació però finalment es van augmentar les funcionalitats).

Les funcionalitats que ofereix l'aplicació són les següents:

- **Accés a les notícies:** al incorporar aquesta funcionalitat es va haver de tenir en compte que era possible que no estiguessin disponibles determinats serveis a la URV que ens permetessin fer una gestió de presentació de notícies adequada. Com que els terminis en el que s'ha portat a terme aquest projecte han estat immediats, s'ha decidit implementar una solució pròpia per a l'accés a notícies (en aquest cas es tractarà de notícies fictícies).
- **Accés a les darreres novetats:** Com que la URV si que disposa d'una funcionalitat que permet obtenir les darreres tres notícies de la web corporativa, s'ha decidit accedir a aquesta informació per tal de proporcionar una aplicació més propera a la realitat.
- **Accés als darrers events:** De la mateixa manera que es poden obtenir les tres darreres notícies, també es poden obtenir els tres darrers actes publicats a l'agenda corporativa de la universitat. Com que no ens aportaria res implementar un servei d'accés a l'agenda fictici (que seria el mateix que per les notícies), únicament s'ha implementat la funcionalitat s'accedeix els darrers actes publicats per tal de proporcionar, com ja s'ha dit abans, una aplicació més real.
- **Localització de centres:** S'ofereix la possibilitat de localitzar en un mapa els diferents centres de que disposa la universitat. Per aquest cas, ens trobem amb les mateixes mancances respecte el sistema de notícies, de manera que ha calgut implementar un servei que retorni els centres de que disposa la universitat i les seves coordenades i dades de detall.
- **Cerca estudiants/PAS/PDI:** Aquesta funcionalitat permet localitzar les persones donades d'alta al directori de la universitat. Tot i que en el seu moment va ésser considerada una funcionalitat a estendre i que es va comentar que es podria implementar mitjançant fitxers en local, donat que ha estat necessari crear una base de dades per a les funcionalitats de consulta de notícies i de persones, s'ha aprofitat l'arquitectura creada per a proporcionar les dades de les persones.
- **Informació institucional:** Aquesta funcionalitat mostra la classificació de centres agrupats per campus, per facultats o escoles... També mostra informació les membres del rectorat, vicerectorats, secretaries i gabinets.
- **Accés a la web de la URV:** es tracta de proporcionar una link a la web de la URV. És un punt d'accés a la plana web, que s'obra en una nova finestra independent.



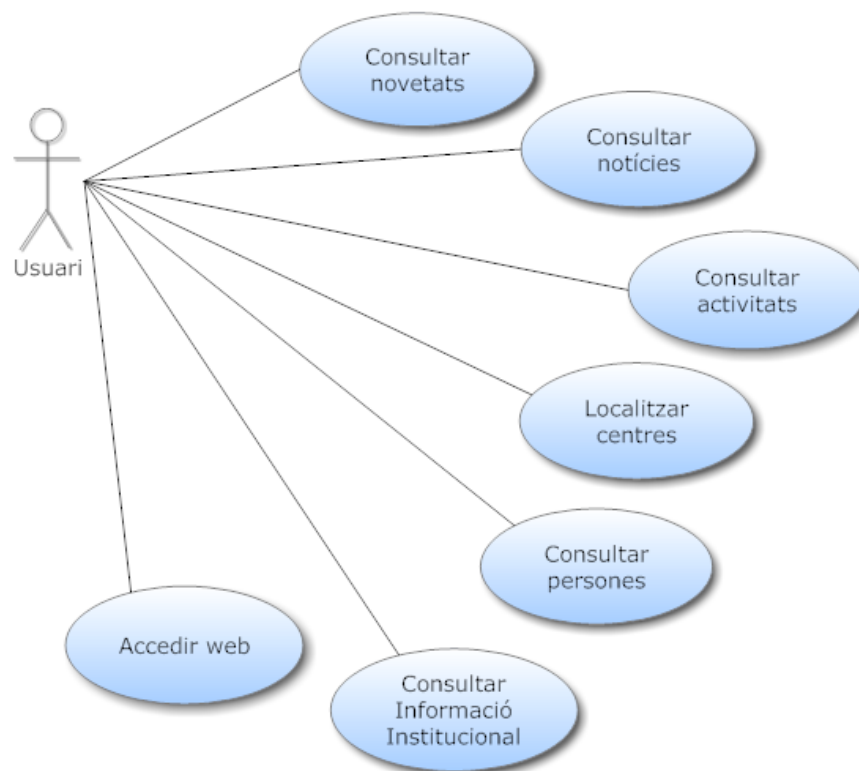
## 3.2 Requeriments no funcionals

Inicialment es va determinar que s'avaluarien les diferents alternatives per tal de desenvolupar l'aplicació de la manera més viable possible. Com es podrà veure en un dels apartats posteriors d'aquest document, s'ha optat per un tipus de solució que permet que una mateixa aplicació amb unes petites variàncies, pot ésser utilitzada en diferents plataformes, de manera que un únic desenvolupament, ens permet disposar d'una aplicació multiplataforma.

## 3.3 Diagrames de casos d'ús

### 3.3.1 Menú general

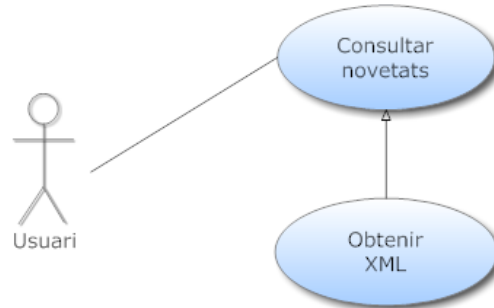
El següent diagrama mostra una vista global de l'aplicació que s'ha implementat, dels actors i dels casos d'ús que descriuen els requeriments funcionals de l'aplicació.



II·lustració 2. Cas d'ús Menú General

L'aplicació, al estar destinada a ser utilitzada mitjançant el mòbil, té com a únic actor l'usuari final, ja que no es contempla cap funcionalitat per al manteniment de dades (en el cas òptim haurien de ser serveis existents a la universitat i que no haurien de requerir cap tipus de desenvolupament).

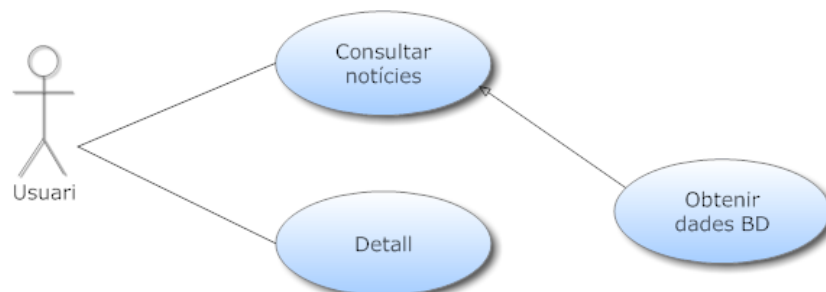
### 3.3.2 Consultar novetats



Il·lustració 3. Cas d'ús Consultar novetats

| Cas d'ús: CONSULTAR NOVETATS |   |
|------------------------------|---|
| <b>Actors</b>                | Usuari  |
| <b>Objectiu</b>              | Veure les darreres novetats publicades a la web de la universitat.  |
| <b>Descripció</b>            | L'usuari selecciona aquesta opció del menú principal i s'accedeix al diari digital per a obtenir les notícies de més recent publicació de la universitat. Com a resultat es mostra una pantalla on visualitza les darreres novetats publicades amb la data de publicació. Cadascuna d'aquestes novetats serà un link a la notícia publicada en el portal. |
| <b>Precondicions</b>         | Es necessari que existeixi el fitxer<br><a href="http://www.urv.cat/noticies/diari_digital/contenido.xml">http://www.urv.cat/noticies/diari_digital/contenido.xml</a>   |
| <b>Postcondicions</b>        |   |

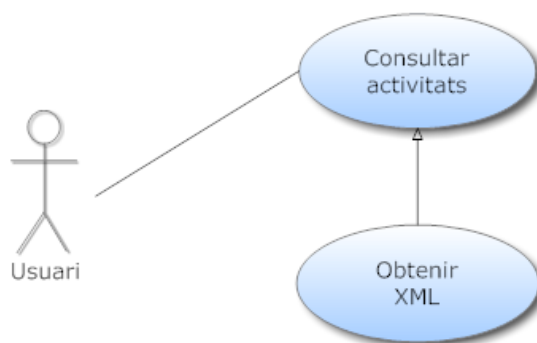
### 3.3.3 Consultar notícies



Il·lustració 4. Cas d'ús Consultar notícies

| Cas d'ús: CONSULTAR NOTÍCIES |   |
|------------------------------|---|
| <b>Actors</b>                | Usuari  |
| <b>Objectiu</b>              | Veure l'històric de notícies publicades a la web de la universitat.   |
| <b>Descripció</b>            | L'usuari selecciona aquesta opció des del menú principal i es mostra una pantalla on visualitzen les darreres notícies publicades. Un cop seleccionada la notícia podrà accedir-ne al detall. Per a la confecció de la llista de notícies, s'accedirà a la base de dades per a obtenir la informació a mostrar. |
| <b>Precondicions</b>         | Carregar la base de dades creada per a proporcionar aquesta informació.   |
| <b>Postcondicions</b>        |   |

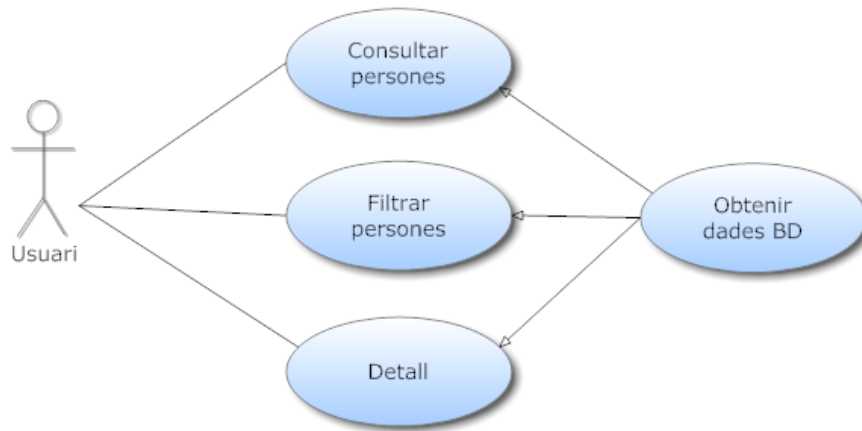
### 3.3.4 Consultar Activitats



II-lustració 5. Cas d'ús Consultar activitats

| Cas d'ús: CONSULTAR ACTIVITATS |  |
|--------------------------------|--|
| <b>Actors</b>                  | Usuari   |
| <b>Objectiu</b>                | Veure les darreres activitats publicades a la web de la universitat.   |
| <b>Descripció</b>              | L'usuari selecciona aquesta opció del menú principal i s'accedeix a l'agenda corporativa per a obtenir les darreres activitats publicades a la universitat. Com a resultat es mostra una pantalla on visualitza les darreres activitats publicades amb la data de publicació. Cadascuna d'aquestes activitats serà un link a la pròpia descripció de l'activitat publicada en el portal. |
| <b>Precondicions</b>           | Es necessari que existeixi el fitxer<br><a href="http://wwwa.urv.cat/noticies/diari_digital/contenido_actos.xml">http://wwwa.urv.cat/noticies/diari_digital/contenido_actos.xml</a>  |
| <b>Postcondicions</b>          |  |

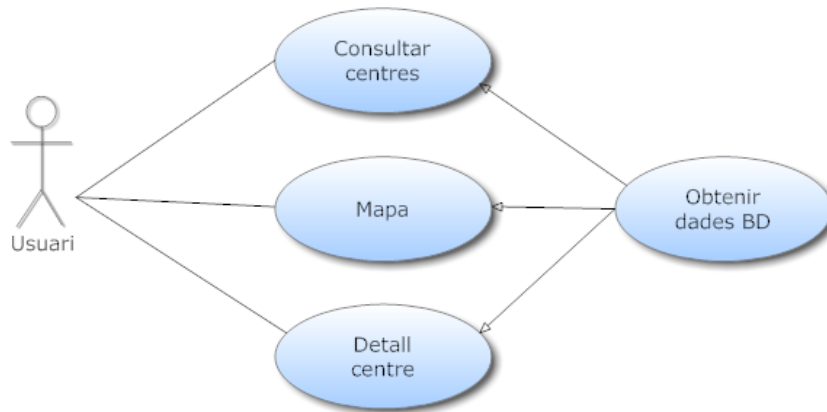
### 3.3.5 Consultar persones



II-lustració 6. Cas d'ús Consultar persones

| Cas d'ús: CONSULTAR PERSONES |   |
|------------------------------|---|
| <b>Actors</b>                | Usuari  |
| <b>Objectiu</b>              | Cercar una persona a la base de dades de la universitat   |
| <b>Descripció</b>            | L'usuari selecciona aquesta opció del menú i es mostrarà una pantalla on filtrar el tipus de persona que es vol buscar: estudiant, personal de docència o investigació... De la llista de persones retornades després de consultar la base de dades, l'usuari podrà accedir al detall de la persona seleccionada per veure'n les dades públiques. |
| <b>Precondicions</b>         | Caldrà que la base de dades creada per a aquest objectiu tingui la informació de les persones que es vulgui permetre consultar.   |
| <b>Postcondicions</b>        |   |

### 3.3.6 Localitzar centres



Il·lustració 7. Cas d'ús Localitzar centres

| Cas d'ús: LOCALITZAR CENTRES |   |
|------------------------------|---|
| <b>Actors</b>                | Usuari  |
| <b>Objectiu</b>              | Localitzar en un mapa el centre seleccionat per l'usuari.   |
| <b>Descripció</b>            | L'usuari selecciona aquesta opció del menú i s'accedeix a la base de dades per a obtenir el llistat de centres. Un cop l'usuari selecciona un centre, es mostra la localització d'aquest centre en un mapa (a partir de les coordenades obtingudes de la base de dades). Des d'aquesta pantalla l'usuari també podrà seleccionar l'opció de veure la informació relativa al centre. |
| <b>Precondicions</b>         | Caldrà tenir carregats els centres de la universitat en la base de dades creada per tal efecte.   |
| <b>Postcondicions</b>        |   |

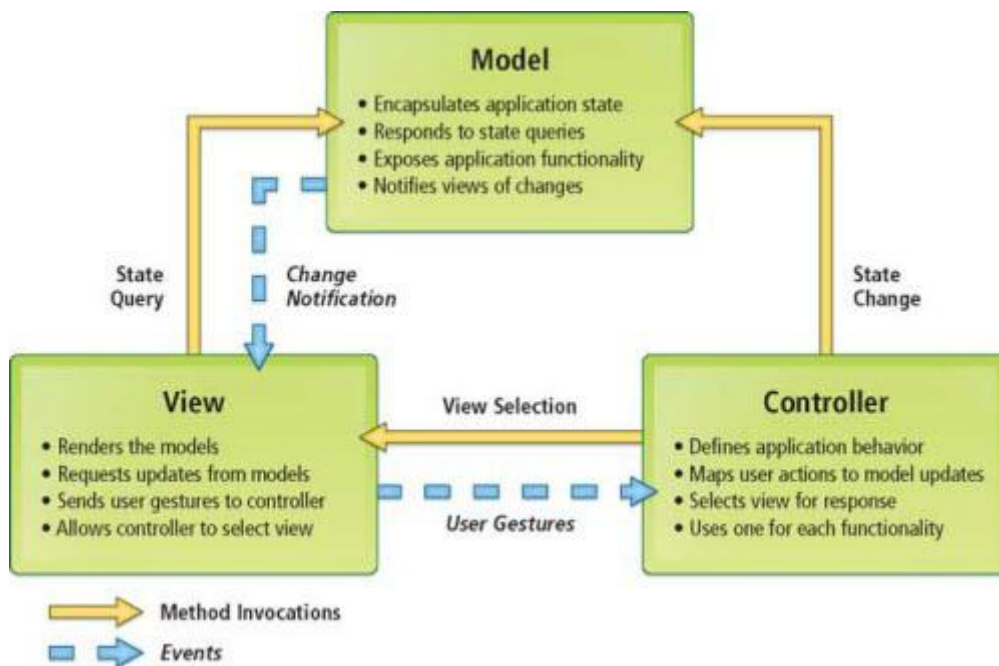
## 4 Disseny

### 4.1 Alternatives desenvolupament

Com ja es va comentar a l'inici del projecte, es presenten moltes alternatives tecnològiques per a realitzar aplicacions per a dispositius mòbils. En els següents apartats es farà una pinzellada de cadascuna de les alternatives més destacades.

#### 4.1.1 Aplicació nativa

La primera alternativa de desenvolupament, òbviament, és escollir una solució nativa implementada amb els SDK dels diferents fabricants i pels diferents tipus de mòbils. En el cas d'apple es pot dir que el seu entorn de desenvolupament (junt amb els simuladors que proporcionen) és molt intuïtiu. L'entorn de desenvolupament d'Apple es basa en el model MVC (Model, View, Controller). Xcode permet separar les funcions d'interfície i de visualització de les de lògica i de negoci (bé, de fet no hi ha cap altra alternativa per a fer-ho). L'esquema següent mostra com funciona aquest model de programació:



II·lustració 8. Esquema MVC

**Model:** conté totes les dades de nucli i lògica. Això inclou variables, connexions a fonts RSS externes o a imatges, funcions detallades.... Aquesta capa és totalment independent de les vistes de manera que es pot canviar la visualització de les dades sense fer cap canvi en l'origen d'aquestes.

**View:** s'encarrega de la part de visualització de les dades. Els estils, una llista, una taula, un reproductor d'àudio, un reproductor de vídeo... són molts dels diferents tipus de vistes que poden haver-hi en una aplicació.

**Controller:** actua d'intermediari entre els altres dos. Connecta objectes de les vistes a través d'un ViewController que passa la informació cap o des d'un model.

La principal avantatge que presenten és que aquestes solucions són òptimes per a molts casos, ja que s'utilitza l'SDK del propi fabricant del mòbil, de manera que es poden utilitzar de manera òptima les funcionalitats que els mòbils de darrera generació ofereixen.

El principal inconvenient, però, és que són dependents de la plataforma en que s'han desenvolupat, de manera que la solució només servirà per un tipus de família de mòbils. Cada cop més, les aplicacions per a mòbils es desenvolupen per a que siguin multi-plataforma de manera que ens hem basat en aquesta tendència per a descartar aquesta opció de desenvolupament.

#### 4.1.2 Aplicació web + contenidor

L'alternativa que es presenta, doncs, al desenvolupament de l'aplicació en codi natiu és fer aquest desenvolupament utilitzant HTML/JS/CSS. L'adopció de l'HTML5 i CSS3 per part dels navegadors dels mòbils ha millorat molt la navegació des d'aquests terminals. A banda, l'explosió de frameworks per a aplicacions per a mòbils, ve a significar que cada cop és més factible crear aplicacions de gran riquesa i interactivitat per dispositius mòbils, que ens han de permetre crear una aplicació gairebé nativa.

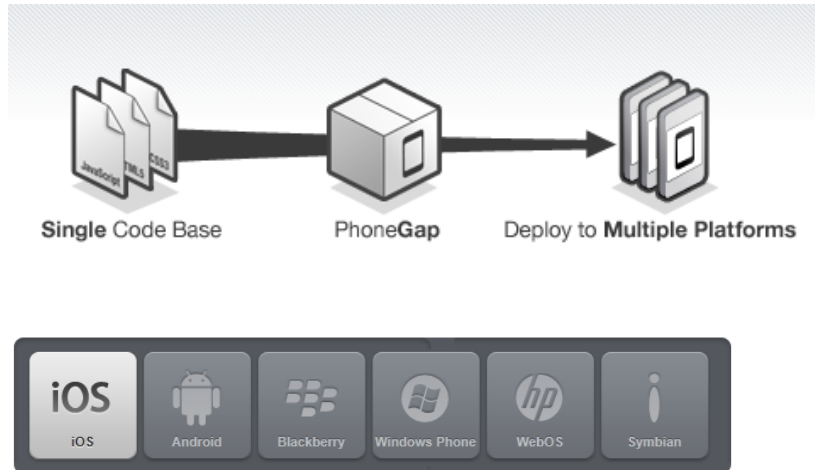
A banda, l'opció d'afegir un contenidor, ens permetrà generar una aplicació que sigui distribuïble a través de les plataformes que proveeixen els diferents fabricants per tal que els seus usuaris les puguin descarregar. Al tractar-se d'una capa d'abstracció entre el dispositiu i l'aplicació, ens hem decantat a escollir uns dels contenidors més coneguts: PhoneGap. Com ja s'ha dit, utilitzant un contenidor, es poden distribuir aquestes aplicacions a través de les tendes d'aplicacions natives per a iPhone, iPad i Android (dirigides a múltiples plataformes per a un codi únic).

##### 4.1.2.1 Contenedor: PhoneGap

###### Per què PhoneGap?

El desenvolupament per a dispositius mòbils és costós. Construir aplicacions per a cada dispositiu - Iphone, Android, Windows Mobile entre d'altres – requereix de diferents frameworks i llenguatges. PhoneGap permet crear aplicacions multiplataforma sense canviar res de codi. Utilitza tecnologies web estàndards per fer de ponts entre les aplicacions web i els dispositius mòbils. A banda, el fet d'utilitzar estàndards, garanteix que encara que aquests evolucionin, continuaran funcionant.

PhoneGap és una plataforma d'aplicació HTML5 que permet crear aplicacions natives amb tecnologies web, alhora que obté accés a les APIs de la plataforma sobre la que es desenvolupa. PhoneGap aprofita les tecnologies web més conegudes pels desenvolupadors web (HTML i JS) per poder desenvolupar aplicacions ràpidament. A banda, disposa d'un SDK que s'integra en diferents entorns de desenvolupament de manera que un únic desenvolupament pot executar-se en múltiples plataformes.



**II-lustració 9. Compatibilitats PhoneGap**

PhoneGap és una implementació de codi lliure d'estàndards oberts. Això vol dir que els desenvolupadors i les empreses poden utilitzar PhoneGap per a aplicacions mòbils ja siguin gratuïtes, comercials, de codi lliure o una combinació d'aquestes.

### Què fa PhoneGap?

PhoneGap proporciona un template per a cada entorn de desenvolupament de manera que quan es vol crear un nou projecte per a dur a terme una aplicació, s'escull aquest template que crea les classes necessàries per construir un pont entre l'aplicació nativa que s'està creant i l'aplicació web que s'executarà dins d'aquest projecte.

En funció del dispositiu pel que es vol construir l'aplicació, es necessiten determinats components. Al descarregar-se PhoneGap i extreure'n el contingut, ens trobem la carpeta lib que conté una carpeta per a cada tipus de dispositiu pel que es pot generar una aplicació: android, bada, blackberry, ios, symbian o windows. En cadascuna d'aquestes carpetes trobem les llibreries que creen el template per l'entorn de desenvolupament associat al dispositiu:

- Aplicació per Android:
  - Cal tenir instal·lat Eclipse
  - Cal tenir instal·lat Android SDK
- Aplicació Blackberry:
  - Cal tenir instal·lat el JDK
  - Cal tenir instal·lat l'Apache Ant
- Aplicació iOS:



- Cal tenir instal·lat l'Xcode

Un cop es tenen tots els components instal·lats es pot crear el projecte a partir del template que proporciona la seva llibreria si es que disposa d'entorn de desenvolupament, ja que, per exemple, per blackberry s'hauran de crear a mà els fitxers necessaris per a generar l'aplicació.

Per tant, PhoneGap, es pot utilitzar en múltiples plataformes.

Un cop es té l'aplicació desenvolupada, la distribució de la mateixa serà la que estigui definida pel propi fabricant del dispositiu pel què hem generat l'aplicació: AppStore per a aplicacions iOS, AndroidMarket per a Android, App World per a Blackberry...

### Altres contenidors

Hi ha altres contenidors que permeten crear aplicacions de manera similar a PhoneGap: Titanium Appcelerator, Corona o Adobe Air Mobile entre d'altres. Les principals avantatges de PhoneGap en front a qualsevol d'aquestes alternatives són bàsicament:

- És la solució que més plataformes mòbils suporta, ja que corre dins d'un navegador.
- És molt fàcil de desenvolupar i proporciona una gran llibertat als que tenen coneixements d'HTML i Javascript.
- Hi ha bastanta documentació i és una solució gratuïta.

Com a inconvenients podríem citar:

- Requereix Xcode per a aplicacions iOS.
- L'aplicació no és més que una pàgina web, de manera que l'aspecte dependrà del framework web utilitzat. Si s'utilitzen frameworks com SenchaTouch, jQuery Mobile XUI... si que s'aconsegueix la semblança a una aplicació nativa.
- No té el rendiment d'una aplicació nativa.

Tot seguit s'adjunta una taula comparativa amb altres contenidors:

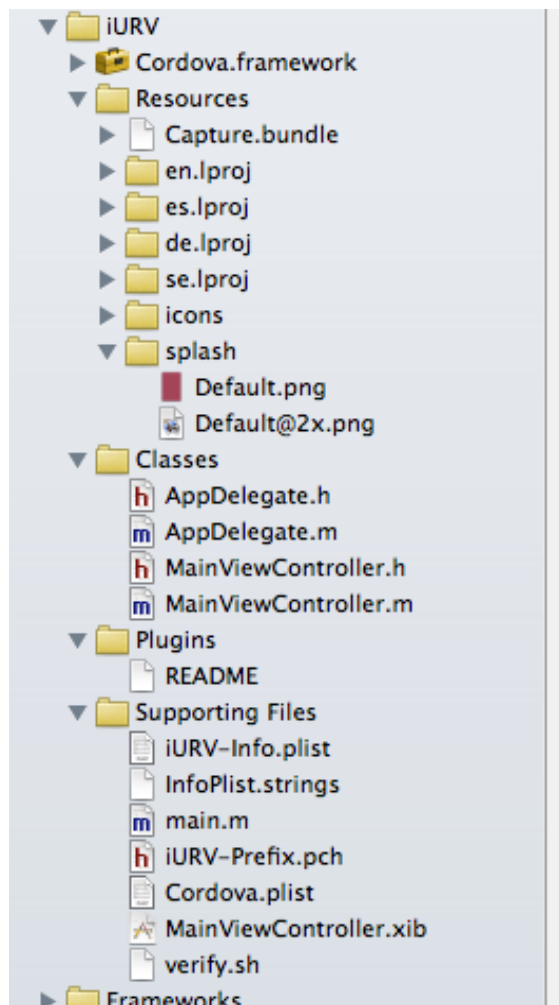
|                                      | Flex 4 - Adobe Air   |       | Appcelerator                        |     |                           | PhoneGap   |       |                        | Corona               |     |
|--------------------------------------|--|-------|-------------------------------------|-----|---------------------------|--|-------|------------------------|----------------------|-----|
|                                      | Android  | IOS   | Android                             | IOS | Win/Mac/Linux             | Android  | IOS   | BlackBerry             | Android              | IOS |
| Requiere Mac/Xcode                   | -  |       | -                                   | Si  | -                         | -  | Si    | -                      | -                    | Si  |
| Lenguaje                             | ActionScript 3.0   |       | Javascript                          |     |                           | Javascript   |       |                        | Lua                  |     |
| Layout                               | MXML   |       | -                                   |     |                           | HTML5/CSS3   |       |                        | -                    |     |
| Componentes                          | Spark (propio de Flex 4), resultado homogéneo en todos los dispositivos. |       | Nativos                             |     | HTML + framework externos | HTML + frameworks externos como Sencha Touch, jQuery Mobile, etc |       |                        | Nativos              |     |
| IDE                                  | Flash Builder 4.5 (solo Win/Mac, no Linux)                               |       | Titanium Studio (basado en Eclipse) |     |                           | Eclipse  | Xcode | No (BB SDK + Ant, JDK) | No (Corona SDK)      |     |
|                                      | FDT - FlashDevelop - IntelliJ IDEA                                       |       |                                     |     |                           |  |       |                        |                      |     |
| Rendimiento                          | Rápido, pero no "suave"  |       | Muy bueno                           |     |                           | Medio (corre en un navegador)                                    |       |                        | Muy bueno            |     |
| Comunidad, documentación, fiabilidad | Muy buena  |       | Regular                             |     |                           | Regular  |       |                        | Muy buena            |     |
| Precio                               | Flex SDK (basado en Java) Gratis   |       | Gratis                              |     |                           | Gratis   |       |                        | \$199 - \$349 al año |     |
|                                      | \$699 Flash Builder 4.5  |       |                                     |     |                           |  |       |                        |                      |     |
| Motor físico, soporte para juegos    | No   |       | No                                  |     |                           | No   |       |                        | Si                   |     |
| Compatibilidad                       | Solo con los gama alta con procesador Arm7                               | Todos | Todos los dispositivos              |     |                           | Si   |       |                        | No                   |     |

En el moment d'iniciar el projecte només s'havia sentit a parlar del contenidor PhoneGap a través de documentació de Forester i Gartner, i al iniciar el projecte es va buscar informació sobre el mateix. Analitzant les parts fortes i les parts febles, la decisió d'utilitzar aquest contenidor va ser molt senzilla ja que, dels inconvenients que es citen, ja es pretenia utilitzar un framework per a generar un aspecte natiu així com el rendiment no era una clau de la solució (ja que les funcionalitats a desenvolupar són bàsicament de consulta i ràpides d'execució).

### PhoneGap i Xcode

Quan ja tenim instal·lat PhoneGap ja es pot seleccionar el template a l'hora de desenvolupar el projecte (dins d'XCode o del compilador que s'estigui utilitzant) que proporcionarà les classes per delegar l'execució de l'aplicació a la part HTML que es desenvolupi.

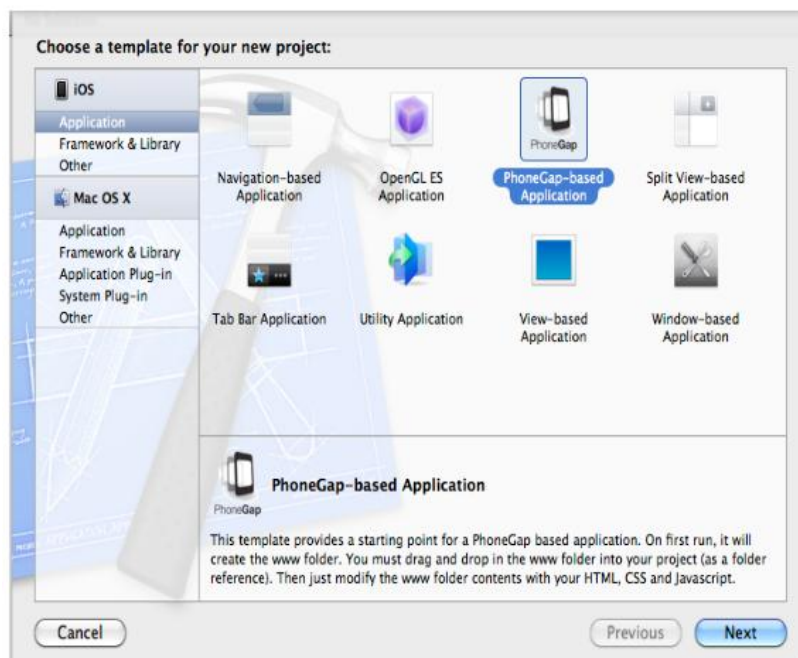
Al crear un projecte d'aplicació utilitzant el template de PhoneGap es crea una estructura bàsica de fitxers:



II-l·lustració 10. Estructura projecte XCode

Si ens centrem en les classes veiem que se'n creen dues diferents AppDelegate i MainViewController (amb les corresponents extensions .h (definició d'objectes) i .m (implementació dels objectes)). L'appdelegate es pot considerar el mòdul principal, el responsable del cicle de vida de l'aplicació mentre que el mainviewController és el que gestiona la connexió entre el model i la vista. Per tal de poder executar l'aplicació li indiquem al view controller que la plana d'inici serà nou\_menu.html de manera que passarà a desenvolupar-se la plana nou\_menu.html com a plana d'inici i a partir d'aquí el control passa a l'aplicació desenvolupada amb HTML/JS/CSS (malgrat que l'appdelegate s'estigui executant durant tota la “vida” de l'aplicació).

```
self.viewController = [[[MainViewController alloc] init]
autorelease];
self.viewController.useSplashScreen = YES;
self.viewController.wwwFolderName = @"www";
self.viewController.startPage = @"nou_menu.html";
self.viewController.view.frame = viewBounds;
```



II-l·lustració 11. Selecció template XCode

Un cop es disposa d'un contenidor que afegeix una capa entre l'aplicació i el dispositiu, només queda decidir quin dels molts frameworks que hi ha disponibles, volem utilitzar per a desenvolupar la nostra aplicació. L'avaluació s'ha fet entre Sencha Touch i jQueryMobile.

#### 4.1.2.2 Avaluació frameworks

##### 4.1.2.2.1 jQueryMobile

jQuery Mobile pot definir-se com un sistema d'interfície unificada que funciona en totes les plataformes dels dispositius mòbils més populars. Està construït a partir de jQuery i jQueryUI Foundation, que proporcionen una base molt sòlida al sistema. Permet amb poc codi, aconseguir grans resultats i proporciona un sistema de "tematització" molt flexible.

El framework inclou la capacitat de navegació en Ajax que proporciona transicions entre pàgines animades i un conjunt bàsic de widgetsUI (user interface): pàgines, diàlegs, toolbars (barres), listviews (l·listes), botons, elements de formularis...

Un de les característiques més potents de jQuery Mobile és que està destinat a ésser utilitzat en una àmplia varietat de plataformes mòbils per a que cap navegador o dispositiu es quedi enrere. Per garantir aquest suport entre dispositius, totes les planes que es desenvolupen amb jQuery Mobile es construeixen en HTML amb una semàntica molt neta per garantir la compatibilitat amb gairebé qualsevol dispositiu habilitat per a la web. En dispositius que interpreten CSS i javascript, jQuery Mobile aplica tècniques de

millora progressiva per a transformar la pàgina semàntica en una de rica i interactiva que aprofita la potència de jQuery i CSS.

Una altra de les característiques més importants, és la facilitat d'aprenentatge del sistema, basat en un sistema de marques que serveixen per a definir el comportament i els temes (estils).

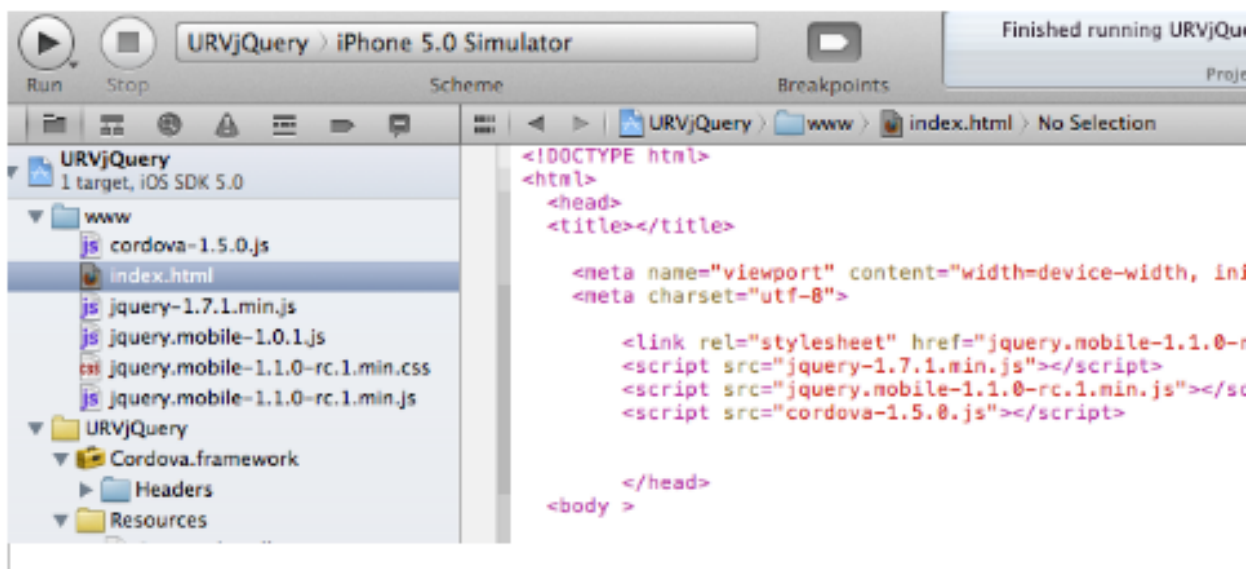
És una solució que prové d'una llarga trajectòria de la seva llibreria original: jQuery tot i que per ella mateixa porta poc temps en el mercat, va sortir a mitjans de 2011.

Les principals característiques, es poden resumir, doncs en:

- Està basat en HTML5
- Permeten simular efectes de transició entre planes
- Contenen temes per al disseny de la interfície molt potents i de fàcil configuració
- El propi framework proporciona les utilitats necessàries per a poder treballar en Ajax.
- En el seu nucli es en l'ús d'etiquetes HTML amb atributs definits pel framework.
- És sorprenent la quantitat de poc codi que cal per a aconseguir un objectiu.

Tot seguit es mostra una petita aplicació desenvolupada amb jQueryMobile (integrada amb PhoneGap):

En aquest cas l'estructura bàsica té el directori www on crearem les planes html a utilitzar. Es pot determinar d'incloure els js i els css i les imatges de manera ordenada en directoris per a cada funció.



Il·lustració 12. Codi + jQueryMobile

En l'html que es crea es necessari incloure el fitxer que permet la cobertura de PhoneGap (cordova.js) i dels fitxers JS que implementen jQuery.

Així, doncs, podem dir que les principals avantatges de jQueryMobile són que ens permet fer molt amb poc codi i que la seva corba d'aprenentatge és molt curta. Per contra ens trobem que és una solució que porta poc temps al mercat (tot i que és una evolució d'una de més antiga) i que pot presentar alguna mancança. De manera global, pel que fa a tots els frameworks, es pot considerar un inconvenient el fet que s'utilitzin llibreries d'aquest tipus que fan que la solució inclogui funcions, estils... que mai s'utilitzaran.

#### 4.1.2.2.2 SenchaTouch

És una solució que prové d'una llarga trajectòria de la seva llibreria original: ExtJS. Proporciona una aparença d'aplicació nativa que pots ser molt interessant en funció de l'aplicació que es vulgui desenvolupar. És una solució molt robusta i de certs anys en el mercat

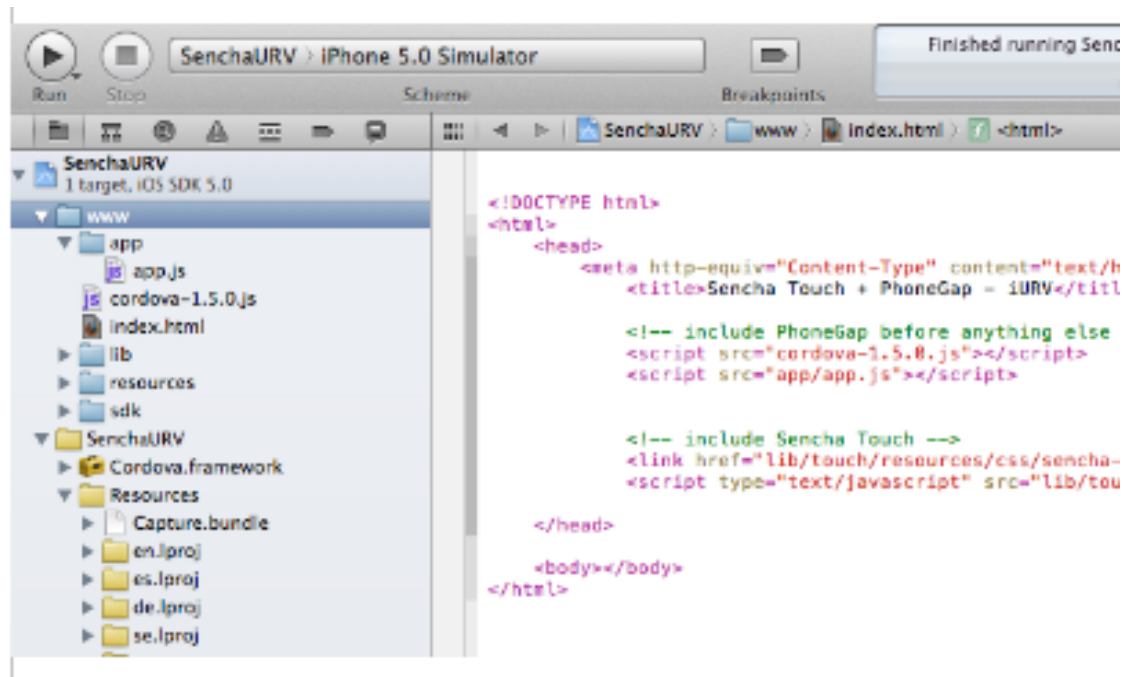
Principals característiques:

- Està basat en HTML5
- Permeten simular efectes de transició entre planes
- Contenen temes per al disseny de la interfície molt potents i de fàcil configuració
- El propi framework proporciona les utilitats necessàries per a poder treballar en Ajax.
- En el seu nucli, abandona l'orientació basada en DOM (Document Object Model), per passar a basar-se en components
- És necessari escriure molt de codi per tal d'aconseguir el que es vol.

Tot seguit es mostra una petita aplicació desenvolupada utilitzant Sencha Touch integrada amb PhoneGap (cordova.js)

L'estructura de les aplicacions creades en Sencha seran com la que es mostra (a partir del directori www que es crea per a integrar-ho amb PhoneGap i de la inclusió del fitxer cordova.js):

- En la carpeta app es guardaran els fitxers que faran la gestió de Model, Vista, Controlador i Gestió de dades de l'aplicació.
- App.js serà el punt d'entrada de l'aplicació.
- En la carpeta resources es guardaran els fulls d'estil i imatges de l'aplicació
- En la carpeta sdk, hi haurà una còpia de l'SDK de Sencha Touch.
- Index.html serà el fitxer HTML de l'aplicació



II-Il·lustració 13. Codi + SenchaTouch

A l'html de la plana únicament s'inclou el fitxer app.js que serà el punt d'entrada de l'aplicació.

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Sencha Touch + PhoneGap - iURV</title>
  <!-- include PhoneGap before anything else -->
  <script src="cordova-1.5.0.js"></script>
  <script src="app/app.js"></script>
  <!-- include Sencha Touch -->
  <link href="lib/touch/resources/css/sencha-touch.css" rel="stylesheet" type="text/css" />
  <script type="text/javascript" src="lib/touch/sencha-touch.js"></script>
</head>
<body></body>
</html>
```

Mentre que tota la “feina” es realitza en el js:

```
Ext.application({
  name: 'Prova URV',
  launch: function() {
    Ext.create("Ext.tab.Panel", {
      fullscreen: true,
      tabBarPosition: 'bottom',
      items: [
        {
          title: 'Home',
          iconCls: 'home',
          html: [
            '',

```

```
        '<h1>Benvingut a la Web de la URV</h1>',  
        "<p>Plana d'inici</p>",  
        '<h2>Sencha Touch (2.0.0)</h2>'  
    ].join(" ")  
    }]  
    });  
});
```

Així, doncs, podem dir que les principals avantatges de SenchaTouch són que es tracta d'una evolució d'ExtJs i que el propi framework porta molt temps en el mercat de manera que es pot considerar una solució molt sòlida. A banda, també permet aconseguir que les aplicacions tinguin un aspecte d'aplicacions natives, cosa que el fa òptim per determinades aplicacions. Per contra, la corba d'aprenentatge és bastant elevada. Tal i com s'ha dit pel jQuery Mobile, presenta el mateix inconvenient que la resta de frameworks, ja que es pot considerar un inconvenient el fet que s'utilitzin llibreries d'aquest tipus que fan que la solució inclogui funcions, estils... que mai s'utilitzaran.

#### 4.1.2.2.3 Comparativa jQueryMobile vs SenchaTouch

A mode de resum, es recullen les principals característiques de cadascun dels dos frameworks.

jQueryMobile:

- És una llibreria d'interfície d'usuari (UI) que es recolza en jQuery i jQuery UI per a la manipulació de DOM, Ajax i altres utilitats.
- Per crear components UI, generalment s'ha d'escriure el codi html corresponent, i la llibreria la proporciona el seu aspecte.
- Com que treballa per a millorar el codi HTML que es crea, permet reutilitzar l'html existent.
- Té un model d'objectes bastant reduït de manera que és més ràpid i senzill d'aprendre, especialment si s'està familiaritzat amb jQuery.
- No imposa una estructura de programació, cosa que proporciona més flexibilitat, però pot portar a dificultats de manteniment de les aplicacions.
- És fàcil d'integrar amb altres frameworks.
- Suporta més dispositius que Sencha Touch.
- No estan vinculat a cap proveïdor.

SenchaTouch:

- És una llibreria que prova de fer-ho tot: components UI, manipulació DOM, Ajax i altres utilitats. Per tant, no depèn d'altres llibreries.
- Es basa en centrar la programació en javascript, de manera que no és necessari escriure massa codi html.
- Té un model d'objectes molt complert, que proporciona més característiques implementades, però que també en dificulta l'aprenentatge.
- És necessari seguir una estructura de programació, de manera que normalment el codi està ben organitzat.



- Proporciona un servidor integrat i abstraccions d'emmagatzemament local que fan més fàcil realitzar operacions CRUD sobre dades relacionals.
- Suporta menys dispositius que jQueryMobile.

La següent taula mostra aquestes i altres característiques:

| Característica                | jQueryMobile      | SenchaTouch               |
|-------------------------------|-------------------|---------------------------|
| Necessitats altres llibreries | Sí                | No                        |
| Corba aprenentatge            | Ràpida            | Normal                    |
| Capacitat UI                  | Normal            | Molt bona                 |
| Llenguatge                    | Javascript + HTML | Gairebé tot Javascript    |
| Documentació                  | Bona              | Regular                   |
| Compatibilitat                | Molts dispositius | iOS, Android i Blackberry |
| Maduresa                      | Regular           | Bona                      |
| Preu                          | Gratuït           | Gratuït                   |

#### 4.1.3 Solució escollida: PhoneGap + jQueryMobile

La decisió sobre quin framework utilitzar, estarà més condicionada a una valoració subjectiva que no a una d'objectiva ja que cadascuna de les solucions pot ser bona en funció de l'aplicació que es vol desenvolupar o de l'experiència que es tingui en aquestes llibreries. Particularment, després de les proves realitzades, m'he decantat per jQueryMobile ja que aparentment té una corba d'aprenentatge més petita que el SenchaTouch i permet desenvolupar les funcionalitats definides.

L'ús de jQueryMobile és tant fàcil com la inclusió de les seves llibreries en el HEAD de les pàgines on es vol utilitzar:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
  <title>Prova</title>
  <link rel="stylesheet" href="http://code.jquery.com/mobile/latest/jquery.mobile.min.css" />
  <script src="http://code.jquery.com/jquery-1.6.4.min.js"></script>
  <script src="http://code.jquery.com/mobile/latest/jquery.mobile.min.js"></script>
  <script src="js/cordova.1.0.5.js"></script>
</head>
<body>
  <div id="details" data-role="page">
    <!-- Content -->
  </div>
</body>
</html>

```

llibreries jQueryMobile

El desenvolupament d'una aplicació amb jQuery Mobile és relativament senzill segons el grau de familiaritat amb HTML i JavaScript. jQuery envolta objectes HTML i classes amb un objecte contenidor general que s'utilitza per crear, buscar i manipular objectes dins de l'aplicació. Utilitza tags específics per a identificar determinats components dins de la pàgina:

```

<div id="details" data-role="page">
  <div data-role="header">
    <h1>iURV</h1>
  </div>
  <div data-role="content">
    <!-- Contingut -->
  </div>
  <div data-role="footer">
    <h3>Fi</h3>
  </div>
</div>

```

TAGs especials jQuery

El fet que PhoneGap estigui integrat amb els diferents compiladors (en funció de la plataforma) facilita l'execució en els simuladors que porten integrats.



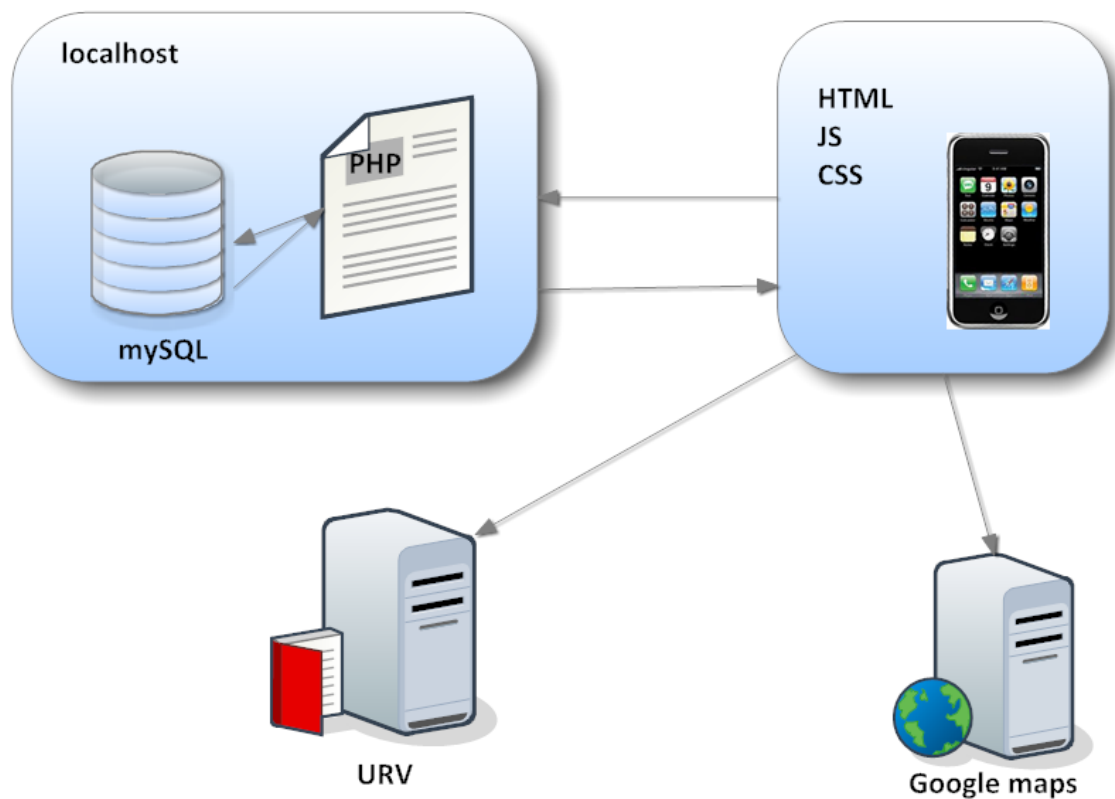
Il·lustració 14. Simulador entorn desenvolupament iOS

## 4.2 Arquitectura de la solució

### 4.2.1 Arquitectura física

L'arquitectura de la solució s'ha vist modificada amb la incorporació del servidor "localhost" que ens donarà les funcionalitats que no ens proporciona el sistema origen.

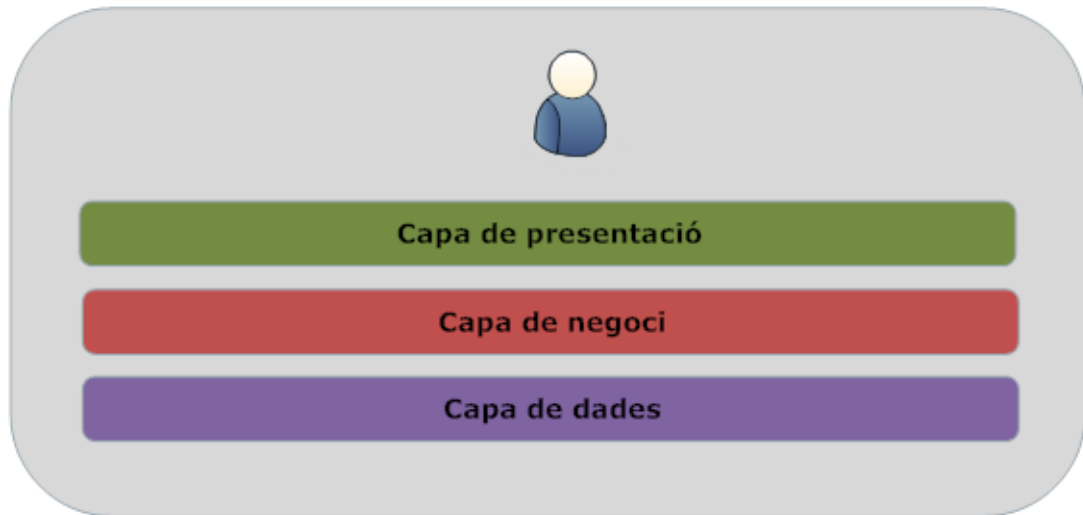
Des del terminal mòbil l'usuari podrà accedir a l'aplicació que en la seva majoria estarà implementada amb planes HTML (que incorporaran JS i CSS) per tal de proporcionar estils i lògica de presentació de les dades.



Il·lustració 15. Arquitectura física

#### 4.2.2 Arquitectura lògica

Les capes que estructuren la solució es recullen en el següent diagrama:



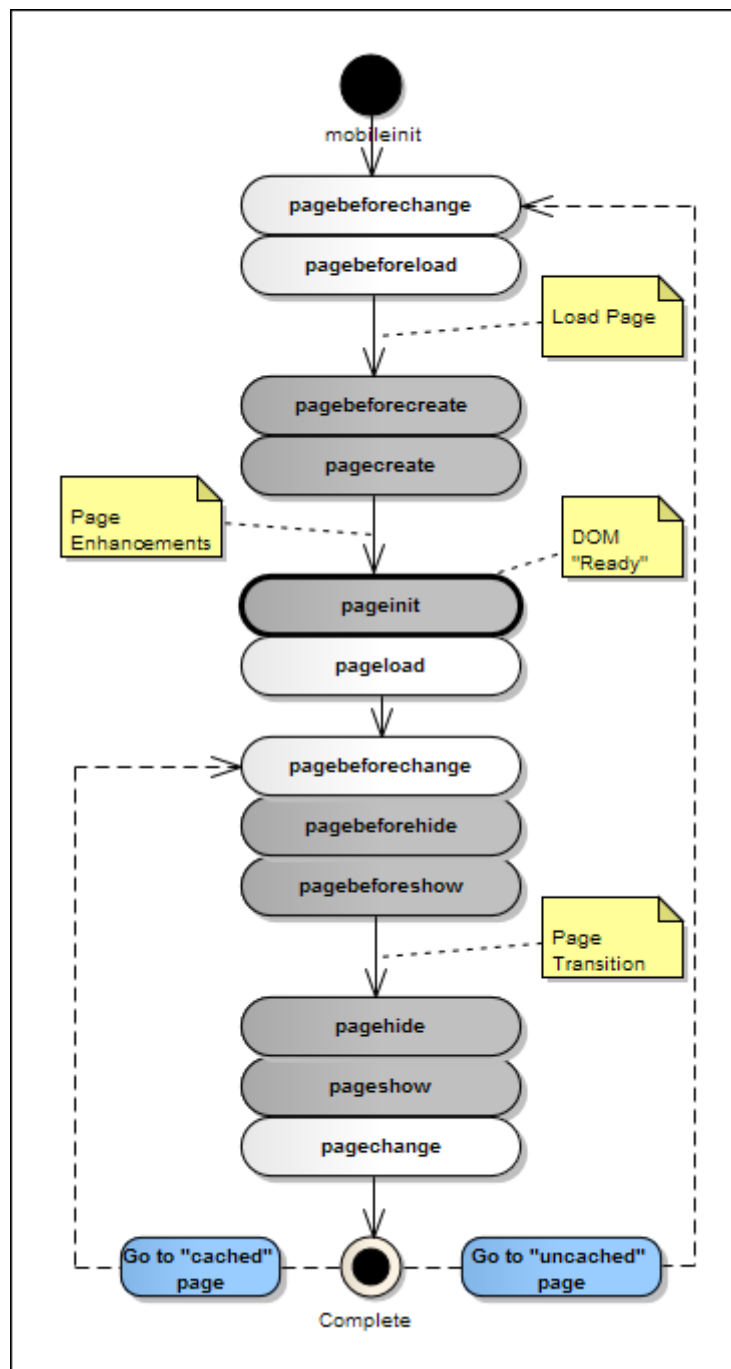
Il·lustració 16. Arquitectura lògica

On:

- Capa de presentació: és la capa més propera a l'usuari. Presenta el sistema a l'usuari, li comunica la informació i rep la informació de l'usuari en un mínim procés. Aquesta capa es comunica amb la capa de negoci.
- Capa de negoci: és la part de l'aplicació que rep les peticions de l'usuari i que envia les respostes després del procés. Es denomina capa de negoci (o inclús lògica de negoci) ja que és on es prenen les decisions dels processos a seguir. Es comunica amb la capa de presentació per a rebre sol·licituds i presentar resultats i amb la capa de dades.
- Capa de dades: és on es troben les dades i és la capa encarregada d'accedir a les mateixes. En aquest cas està formada per un únic gestor de bases de dades (tot i que es podrien tenir diferents gestors de BD) que és el responsable de retornar les dades guardades a la BD per l'aplicació.

La capa de presentació és la que s'executa en el propi dispositiu. Com s'ha dit, l'aplicació la componen plantilles html i codi javascript de la lògica de presentació. Les plantilles de l'aplicació tenen el codi html per presentar els components per pantalla, així com també inclouen fitxers javascript que seran els que tindran implementada la lògica de presentació.

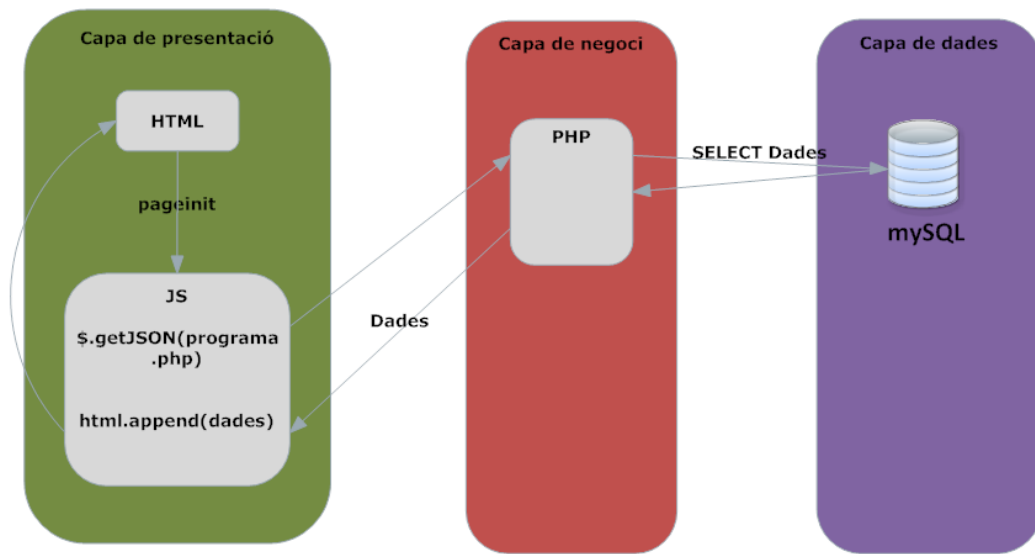
Al carregar una finestra de l'aplicació es pot produir qualsevol dels següents events:



II-lustració 17. Diagrama d'events jQueryMobile

El que es fa a nivell de javascript, és interceptar l'event que ens interessa per tal de executar la lògica desitjada. En la majoria de casos, com es veurà posteriorment en l'apartat d'implementació, l'event tractat és el **pageinit** que es produeix quan ja es té el model DOM de la plana. És en aquest moment en que es fa la crida cap a la capa de negoci (si és que la plana ho requereix), per obtenir, per exemple, una dada de la base de dades. Les crides són a programes PHP de la capa de negoci que accedeixen a la capa de dades per obtenir les dades de la base de dades i així retornar-les al client.

El següent diagrama mostra gràficament com és la seqüència d'execució (tot i que es pot veure en un apartat posterior el diagrama de seqüència de determinats casos d'ús).



II-lustració 18. Diagrama execució jQueryMobile

### 4.3 Altres aspectes tecnològics

Com ja es comentava en els riscos del projecte, hi ha determinats serveis que no es troben disponibles, en l'actualitat, a la universitat. Donades aquestes mancances dels serveis d'origen, i tal i com s'ha pogut llegir en l'anàlisi dels sistema, serà necessari que determinats serveis es desenvolupin per poder donar contingut a aquest projecte.

Per aquest motiu, per determinades funcionalitats (llistat de centres, llistat de persones i llistat de notícies) es crearà una base de dades en un servidor local per estendre les funcionalitats de l'aplicació i s'implementaran les funcionalitats necessàries per a accedir-hi.

Els següents apartats contenen una descripció més detallada d'algun dels components utilitzat per tal de portar a terme la solució.

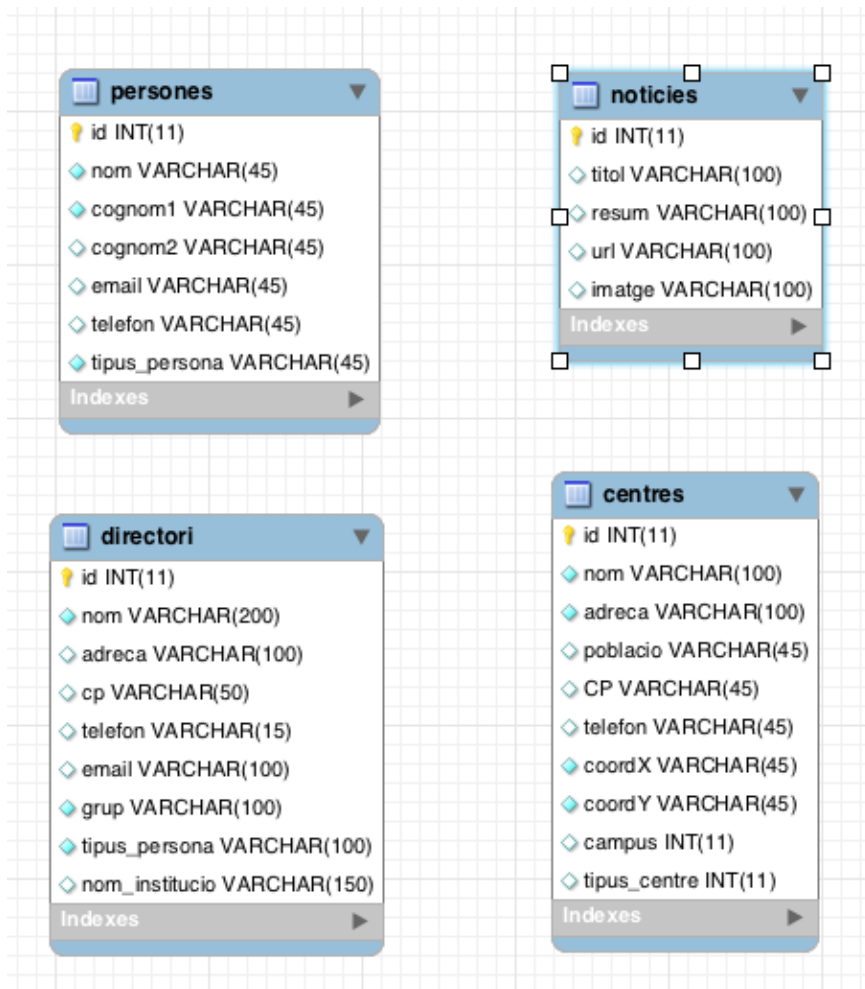
#### 4.3.1 SGBD

S'utilitzarà MySQL com a base de dades local on es definiran les taules d'estudiants, de centres i de notícies. A banda, i com es detalla més endavant, s'ha afegit una taula addicional per introduir-hi la informació del directori corporatiu (ja que s'ha volgut aprofitar la informació existent per donar-li una altra estructuració de visió).

MySQL és un sistema de gestió de bases de dades relacionals (anglès RDBMS – Relational DataBase Management System) multithread i multiusuari, que usa el llenguatge SQL (Structured Query Language) per a accedir a les dades.

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

Per dur a terme els serveis que de moment no estan implementats a la universitat, s'ha vist necessari definir les següents taules:



II-lustració 19. Esquema taules BD

No es mostra el diagrama de relacions de les taules ja que només cal una estructura de taula per a cada tipus d'informació.

#### 4.3.2 Accés a les dades

Per obtenir les dades de la base de dades, s'implementaran programes PHP que retornaran la informació en format JSON (Javascript Object Notation), de manera que des dels HTML de l'aplicació, s'obtingran fent ús de la funció `jQuery.getJSON()` (obté les dades fent una crida HTTP GET) o amb `jQuery.ajax()`. En ambdós casos el que s'aconsegueix és crear aplicacions interactives, de manera que mentre es mostra una

pantalla a l'usuari, en un segon terme es poden fer crides al servidor de manera asíncrona i sense que l'usuari se n'adoni.

**AJAX:** és l'acrònim de Asynchronous JavaScript and XML, un conjunt de tecnologies web que permeten que les aplicacions web des dels seus clients es comuniquin i sol·licitin informació al servidor de manera asíncrona sense interferir amb el contingut i comportaments de la pàgina actual, és a dir, sense refrescar (postback) la pàgina web per complet.

La seva implementació es basa en l'objecte XMLHttpRequest o XHR el qual, originàriament, va ésser desenvolupat per Microsoft i aparegué per primer cop com un ActiveX per a Internet Explorer, fins que posteriorment la resta de navegadors el van acollir implementant les seves pròpies implementacions.

En la implementació de l'aplicació s'utilitzaran les crides AJAX per a l'actualització del mapa i l'accés a la base de dades.

**JSON:** és l'acrònim de JavaScript Object Notation. És un estàndard obert basat en text dissenyat per a intercanvi de dades llegible per humans. Deriva del llenguatge script JavaScript, per a representar estructures de dades simples i llistes associatives.

El format JSON s'utilitza habitualment per serialitzar i transmetre dades estructurades en una connexió de xarxa. S'utilitza principalment per intercanviar dades entre un servidor i una aplicació web, sent una alternativa a l'XML.

### 4.3.3 XML corporatius

Per obtenir els XMLs que estan publicats a la web de la URV, farem ús de l'api <https://ajax.googleapis.com/ajax/services/feed/load> de google a la que se li indica la funció que s'executarà per a fer el render de l'XML obtingut.

El format dels XMLs respon als formats estàndards de continguts de sindicació de manera que quan s'obté l'xml [http://wwwa.urv.cat/noticies/diari\\_digital/contenido.xml](http://wwwa.urv.cat/noticies/diari_digital/contenido.xml) només cal accedir a l'ítem i a les propietats que es vulguin mostrar.



```
<?xml version="1.0" encoding="iso-8859-1" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:sy="http://purl.org/rss/1.0
/modules/syndication/" xmlns:admin="http://webns.net/mvcb/"
xmlns:content="http://purl.org/rss/1.0/modules/content/" xmlns="http://purl.org
/rss/1.0/">
  <channel rdf:about="http://obrador.antaviana.com">
    <title>Houdini v2.0 - XML News RSS/RDF Feed</title>
    <description>Notícies</description>
    <link>http://obrador.antaviana.com</link>
  </channel>
  <item><title>Segona Jornada de Portes Obertes a la URV el dimecres, 28 de
març</title><link>http://www.urv.cat/noticies/diari_digital
/cgi/principal.pl?fitxer=noticies/noticia012860.htm</link><pubDate>27
/03/2012</pubDate><author>institució</author></item><item><title>Una investigació
que proporciona les claus per decidir les prioritats en els esforços de conservació
de les espècies, a Science</title><link>http://www.urv.cat/noticies/diari_digital
/cgi/principal.pl?fitxer=noticies/noticia012844.htm</link><pubDate>23
/03/2012</pubDate><author>recerca</author></item><item><title>Trobada de premis
Nobel de Química al Campus d'Excel·lència Internacional Catalunya Sud</title>
<link>http://www.urv.cat/noticies/diari_digital/cgi/principal.pl?fitxer=noticies
/noticia012861.htm</link><pubDate>27/03/2012</pubDate><author>campus</author></item>
</rdf:RDF>
```

#### 4.3.4 Mapes

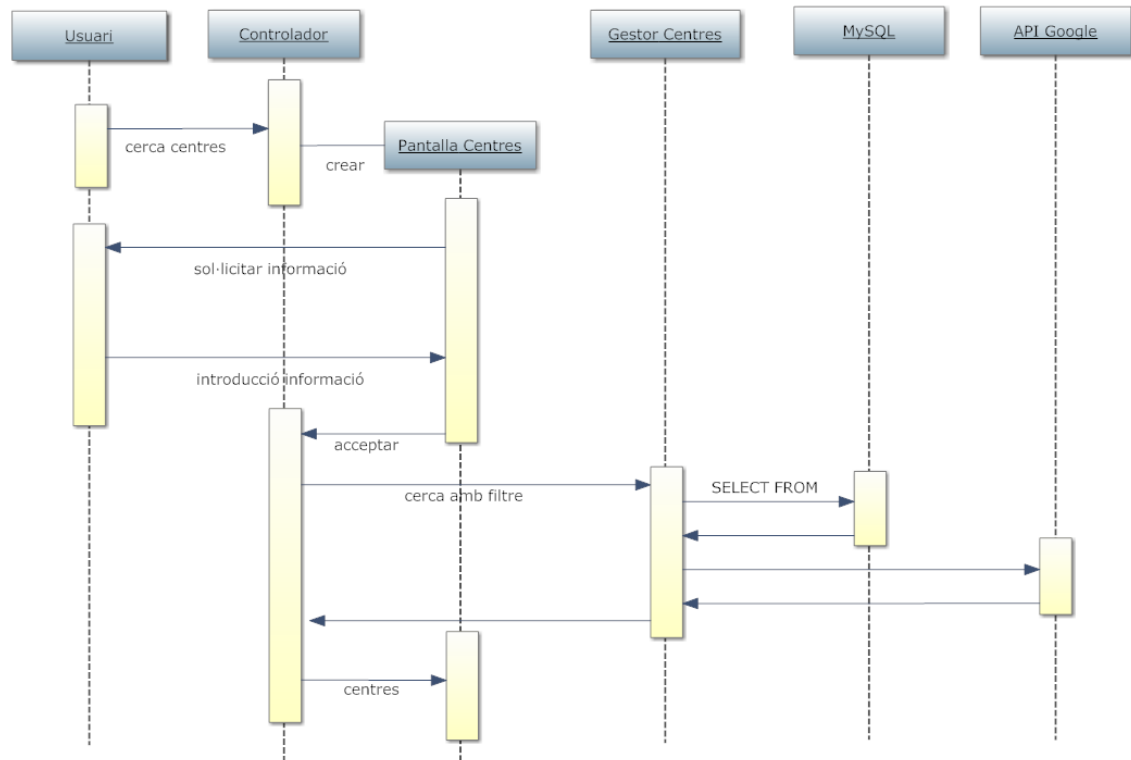
Per a la localització de campus i centres es fa ús de l'api de google <http://maps.google.com/maps/api/js>. Aquesta API permet, a partir de les coordenades de latitud i longitud d'un punt, localitzar-lo en un mapa. Les coordenades de localització estaran emmagatzemades a la base de dades i s'agafaran al accedir al detall del centre del que es vol conèixer la localització.

Un cop s'inclou el javascript corresponent, per crear un punt en unes determinades coordenades de longitud i latitud cal fer la crida a la funció `new google.maps.LatLng(41.118, 1.245)` (ex. coordenades ciutat Tarragona).

#### 4.4 Diagrames de seqüència

El principal objectiu del diagrama de seqüència és mostrar les interaccions entre els objectes d'un sistema. Tot seguit es mostren els diagrames de seqüència d'algun dels casos d'ús implementats,

#### 4.4.1 Localitzar centres

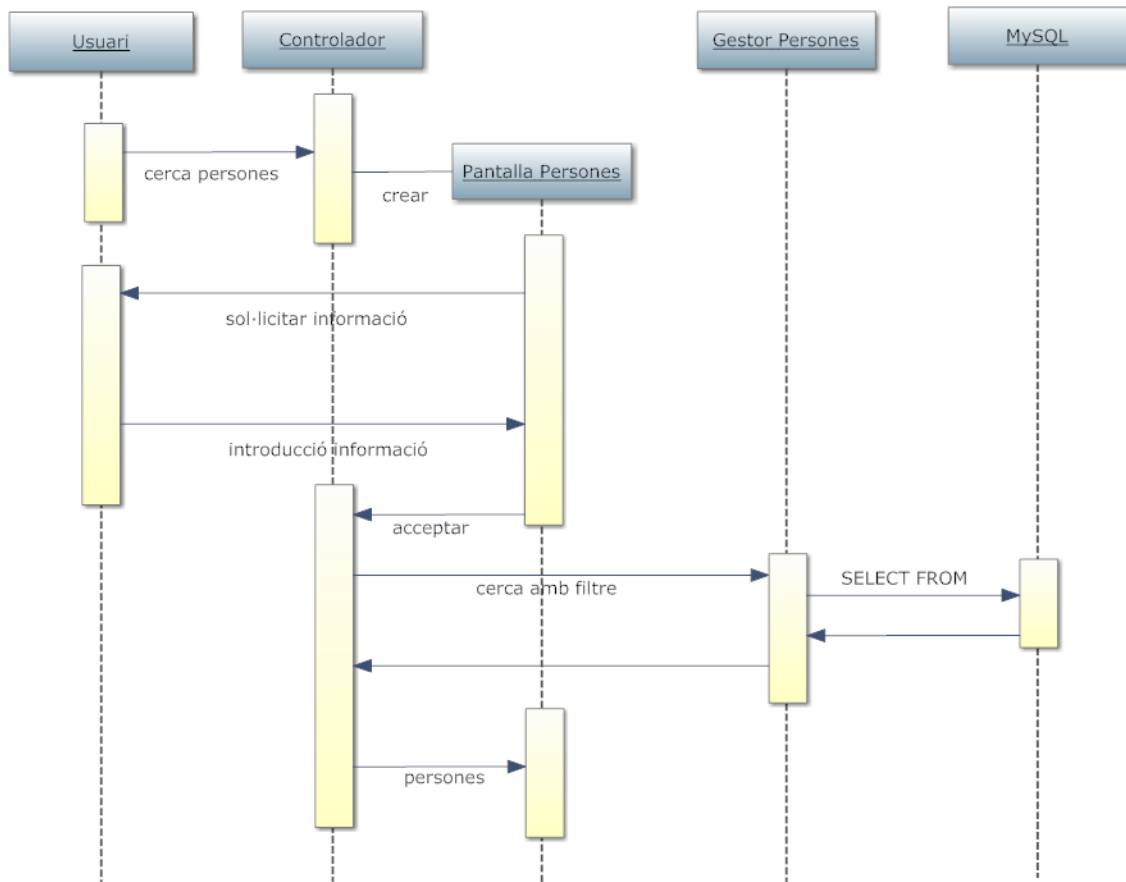


II-lustració 20. Diagrama seqüència Localitzar centres

La seqüència reflectida en el diagrama és la següent:

- L'usuari envia una senyal al Controlador de la interfície indicant la seva intenció de cercar els centres de la universitat. El framework de PhoneGap fa de controlador de la capa de presentació i pren decisions de què fer quan rep una petició des del terminal de l'usuari.
- El controlador crea una pantalla d'entrada d'informació.
- Es mostra la pantalla a l'usuari esperant que indiqui els criteris de cerca que vol aplicar.
- La pantalla notifica al controlador que l'usuari ha acceptat les dades introduïdes.
- El controlador envia al GestorDeCentres la cerca que es vol fer a la taula de centres.
- El GestorDeCentres envia la cerca cap al MySQL (en format SQL) i n'espera la resposta
- El GestorDeCentres fa la crida a l'API de google per tal d'obtenir el codi per pintar per pantalla el mapa que, finalment, es presenta per pantalla.

#### 4.4.2 Consultar persones

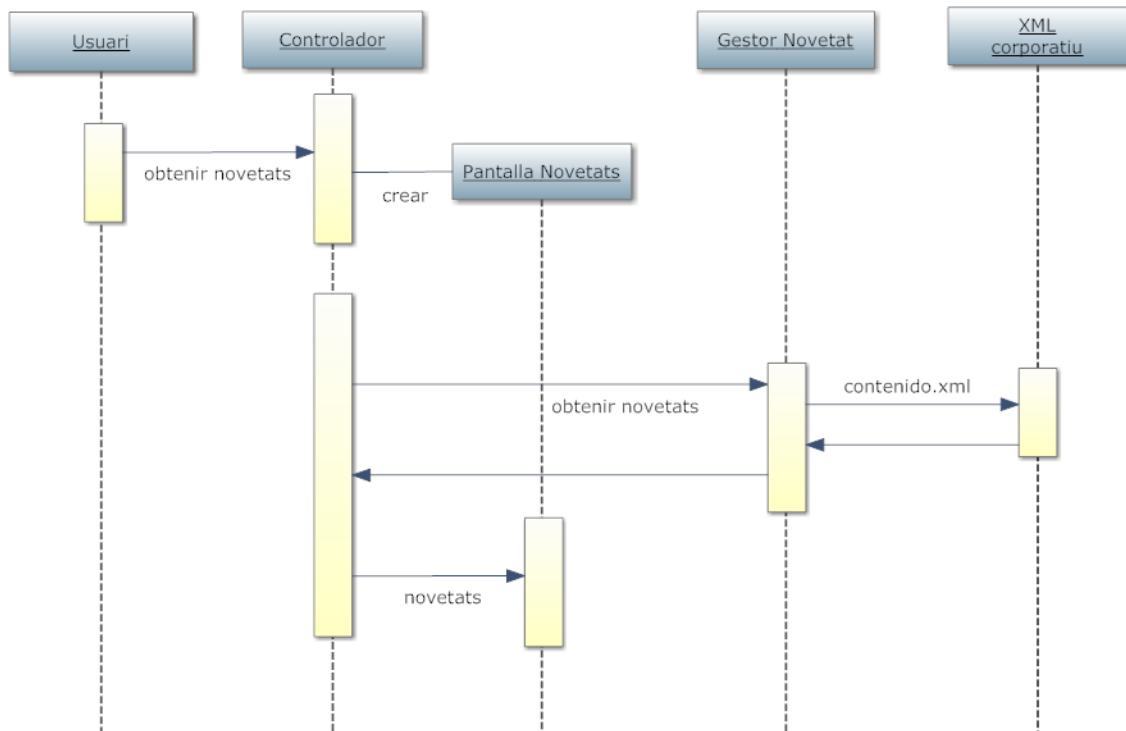


Il·lustració 21. Diagrama seqüència Consultar persones

La seqüència reflectida en el diagrama és molt similar a la seqüència comentada en el punt anterior. S'ha de dir que inicialment no estava previst que s'utilitzés una finestra per tal de filtrar dades però al final s'ha estimat oportú fer-ho. Així la seqüència queda de la següent manera:

- L'usuari envia una senyal al Controlador de la interfície indicant la seva intenció de cercar els persones de la universitat.
- El controlador crea una pantalla d'entrada d'informació.
- Es mostra la pantalla a l'usuari esperant que indiqui els criteris de cerca que vol aplicar.
- La pantalla notifica al controlador que l'usuari ha acceptat les dades introduïdes.
- El controlador envia al GestorDePersones la cerca que es vol fer a la taula de centres.
- El GestorDePersones envia la cerca cap al MySQL (en format SQL) i n'espera la resposta, que finalment es presenta per pantalla.

### 4.4.3 Consultar novetats



Il·lustració 22. Diagrama seqüència Consultar novetats

El darrer diagrama de seqüència que s'adjunta és el corresponent a l'obtenció de les novetats publicades a la web corporativa (l'obtenció de les novetats de l'agenda corporativa es fa amb la mateixa seqüència motiu pel qual no s'ha inclòs).

La seqüència queda de la següent manera:

- L'usuari envia una senyal al Controlador de la interfície indicant la seva intenció d'obtenir les darreres novetats publicades a la web corporativa de la universitat.
- El controlador crea una pantalla per mostrar les novetats.
- El controlador envia al GestorDeNovetats la necessitat de petició del fitxer de dades.
- El GestorDeNovetats envia la petició d'obtenció de l'XML i n'espera la resposta que, finalment, es presenta per pantalla.

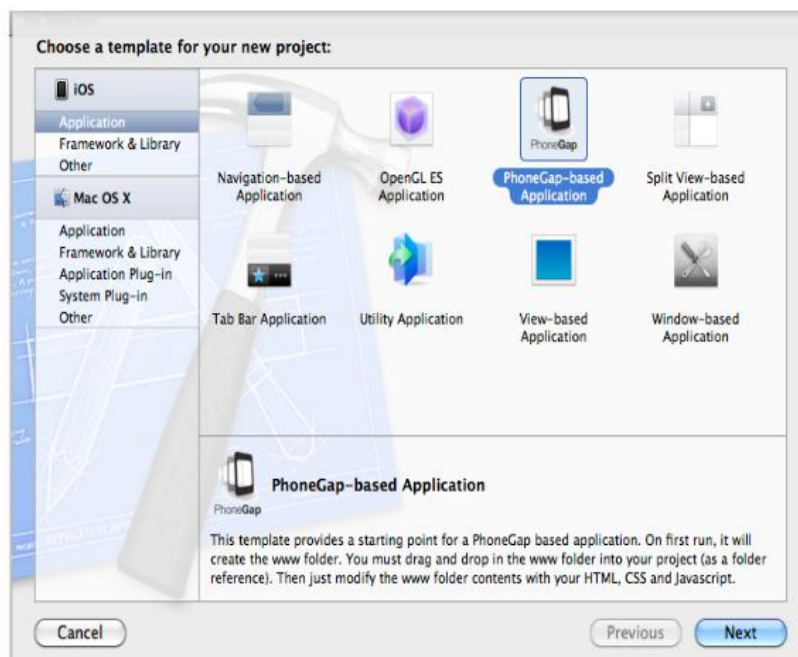
## 5 Implementació

Els següents apartats recullen les implementacions que s'han realitzat per tal de desenvolupar les diferents funcionalitats de l'aplicació iURV.

### 5.1 Contenidor aplicació: PhoneGap

Com ja s'ha comentat en l'apartat de disseny, la solució de l'aplicació s'ha desenvolupat sobre XCode amb el contenidor PhoneGap que permet crear aplicacions natives amb tecnologies web.

Per tal d'utilitzar PhoneGap cal instal·lar-se la el paquet que es pot descarregar de la seva web i que ens proporciona un template per crear aplicacions PhoneGap des d'XCode. Al seleccionar el template de PhoneGap (en les noves versions s'anomena Cordova) es genera una sèrie de codi que ens abstreu del codi natiu per tal de cedir l'execució a les planes html.



II-lustració 23. Selecció template PhoneGap

Al crear una aplicació amb aquest template es creen una sèrie de components dels que ara comentarem el més important. Es creen dues classes AppDelegate i MainViewController (amb les corresponents extensions .h (definició d'objectes) i .m (implementació dels objectes)).

## AppDelegate.m

Tot seguit es mostra una extracció del codi generat en aquesta classe:

```
#pragma UIApplicationDelegate implementation
/**
 * This is main kick off after the app inits, the views and
 * Settings are setup here. (preferred - iOS4 and up)
 */
- (BOOL) application:(UIApplication*) application
didFinishLaunchingWithOptions:(NSDictionary*) launchOptions
{
    NSURL* url = [launchOptions
objectForKey:UIApplicationLaunchOptionsURLKey];
    if (url && [url isKindOfClass:[NSURL class]]) {
        self.invokeString = [url absoluteString];
        NSLog(@"iURV launchOptions = %@", url);
    }
    CGRect screenBounds = [[UIScreen mainScreen] bounds];
    self.window = [[[UIWindow alloc] initWithFrame:screenBounds]
autorelease];
    self.window.autoresizesSubviews = YES;
    CGRect viewBounds = [[UIScreen mainScreen] applicationFrame];
    self.viewController = [[[MainViewController alloc] init]
autorelease];
    self.viewController.wwwFolderName = @"www";
    self.viewController.startPage = @"principal.html";
    self.viewController.view.frame = viewBounds;
    // over-ride delegates
    self.viewController.webView.delegate = self;
    self.viewController.commandDelegate = self;
    self.viewController.webView.scrollView.scrollEnabled=true;
    ...
}
```

L' `UIApplicationDelegate` declara mètodes implementats a partir de la delegació de l'objecte `UIApplication`. Els mètodes que conté proporcionen informació referent a events que es produeixen, que s'acaba el llançament de l'aplicació, quan es queda sense memòria...

En el nostre cas el mètode implementat és:

```
(BOOL) application:(UIApplication*) application
didFinishLaunchingWithOptions
```

Aquest mètode es crida quan l'aplicació s'ha acabat d'arrancar. És en aquest moment en el que se li indica quina serà la plana principal de l'aplicació (en aquest cas `principal.html`) i el directori on es trobaran les planes.

Per altra banda amb la implementació del mètode:

```
- (BOOL) webView:(UIWebView*) theWebView
shouldStartLoadWithRequest:(NSURLRequest*) request
navigationType:(UIWebViewNavigationType) navigationType
```

permetem la navegació a planes de fora l'aplicació. D'aquesta manera si la URL a la que s'intenta accedir comença per `http` o `https`, l'aplicació sabrà que s'ha d'obrir una URL externa en el navegador del dispositiu.

## 5.2 Capa de presentació

### 5.2.1 Visió general jQuery Mobile

Els recursos utilitzats en la capa de presentació són els estàndards de programació web HTML/CSS/JS amb la particularitat que s'utilitza la llibreria jQueryMobile per a la programació de les planes.

L'estructura de les pàgines ("page") de jQueryMobile estan optimitzades per a poder contenir pàgines simples o pàgines internes a les que s'accedeix des de la pròpia plana.

#### Estructura d'una plana per a mòbils:

- L'html ha de començar amb l'etiqueta '**doctype**' d'HTML5 per tal de poder treballar amb totes les característiques del framework. Tot seguit cal indicar, dins el **<head>** de la plana, el meta "**viewport**" tal i com es defineix tot seguit:  

```
<meta name="viewport" content="width=device-width, initial-scale=1.0;" />
```

d'aquesta manera l'amplada de la plana s'adaptarà a l'amplada del dispositiu on s'executa. Aquesta definició, però, no treu la possibilitat a l'usuari de fer zoom de les pàgines.
- Dins del tag **<body>** cada pàgina del dispositiu mòbil s'identifica amb un **div** amb l'atribut **data-role="page"** :

```
<div id="news" data-role="page">  
  ...  
</div>
```

Dins de l'estructura "page" es pot afegir qualsevol component HTML vàlid, però normalment les planes implementades amb jQuery Mobile acostumen a tenir com a fills del node "page" els elements amb els data-roles de "**header**" (per la capçalera), "**content**" (pel contingut) i "**footer**" (pel peu).

```
<div id="news" data-role="page">  
  <div data-role="header" >  
  </div> <!-- header -->  
  <div data-role="content">  
  </div> <!-- content -->  
  <div data-role="footer" >  
  </div> <!-- footer -->  
</div>
```

- Un únic HTML pot contenir múltiples "pages" que es carreguen conjuntament. Cada bloc de pàgina de de tenir un identificador únic (ex. id="news") que s'utilitzarà internament per a linkar les planes (ex. href="#news"). Quan es clica sobre el link, el framework comprova si internament hi ha alguna pàgina amb aquest ID i si és així fa la transició cap a aquest div.

#### Estils d'una plana per a mòbils:

- jQuery Mobile ofereix la possibilitat d'assignar una estètica ("theme") als components que integren una plana. JQuery Mobile ofereix 5 temes predefinits que contenen l'estètica dels diferents components que poden integrar una plana (capçalera, botons, llistes...). Els temes que incorpora jQuery Mobile s'identifiquen amb les lletres A, B, C, D i E.
- L'atribut data-theme es pot aplicar tant al header, al content com al footer, però és recomanable incloure'l en el div corresponent a la definició del data-role="page" per garantir que l'estil s'apliqui a tota la plana.
- El sistema de temes de jQueryMobile és similar al sistema ThemeRoller de jQuery UI amb unes incorporacions importants: utilitza propietats CSS3 per a utilitzar cantonades arrodonides, ombres en el text, degradats... Per a la definició dels estils propis de la universitat s'ha utilitzat l'eina ThemeRoller (<http://jquerymobile.com/themeroller/>). Aquesta eina permet definir l'estil gràficament i descarregar el CSS corresponent per tal d'incloure'l en l'aplicació.

### Inclusió llibreries i estils

- Dins del tag **head**, cal afegir la inclusió dels fitxers javascript de jquery i dels fitxers d'estils definits:

```
<link rel="stylesheet"
href="http://code.jquery.com/mobile/latest/jquery.mobile.min.css" />
<link rel="stylesheet" href="css/urv-theme.css"/>
<link rel="stylesheet" href="css/urv-theme.min.css"/>

<script src="js/jquery.js"></script>
<script src="js/jquery.mobile-1.0rc1.min.js"></script>
<script
src="http://maps.google.com/maps/api/js?sensor=true"
type="text/javascript"></script>
<script src="js/jquery.ui.map.full.min.js"
type="text/javascript"></script>
```

- Per últim al final de la plana es carreguen els javascripts propis de l'aplicació. Es podria haver creat un únic js per tota l'aplicació però s'ha decidit estructurar-ho en base a les funcionalitats implementades.

```
<!-- js de l'aplicació -->
<script src="js/novetats.js"></script>
<script src="js/persones.js"></script>
<script src="js/detallpersones.js"></script>
<script src="js/centres.js"></script>
<script src="js/detallcentres.js"></script>
```

## 5.2.2 Component menú

És el component descrit en el cas d'ús Menú General.



És el component d'accés a l'aplicació. Conté les icones corresponents a cada funcionalitat per tal de facilitar-ne l'accés. Està implementat utilitzant l'estructura ui-grid-b que crea una graella de tres columnes on s'ubiquen els diferents accessos.

Cadascun d'aquests accessos, a excepció de l'accés a la web de la URV, són links a la pròpia plana, és a dir, estaran definits dins del tag `data-role="page"`.

### 5.2.3 Component news

El component de news és el que s'ha descrit en el cas d'ús de Consultar novetats.

Dins la definició de la plana, en la secció `data-role="content"` s'afegeix una llista que s'omplirà dinàmicament amb el contingut de l'XML de novetats. En aquest cas es passa el control a la part de lògica de presentació. En el fitxer **novetats.js** s'han implementat les funcions de control sobre els components relacionats amb les notícies i activitats. Aquest fluxe de control s'aconsegueix pel fet de tenir implementada la funció:

```
$('#news').bind('pageinit',function(){
```

Al detectar que s'ha produït una inicialització de #news el control de l'execució passa a aquesta funció. Aquesta serà la manera en que es puguin executar crides a fitxers, bases de dades... quan es vol mostrar un component les dades del qual no són estàtiques.

En aquest cas s' farà la crida a la url:

```
var googUrl =  
'https://ajax.googleapis.com/ajax/services/feed/load?v=1.0&num='+n  
um+'&q='+url+'&callback=jQuery.rssReaderParser&context='+id;
```

on:

```
var num = "3";  
// Definim variables per recollir les darreres notícies publicades  
var rssUrl =  
"http://wwa.urv.cat/noticies/diari_digital/contenido.xml";  
var id = "post_results";
```

de manera que se li indica que quan obtingui el fitxer XML es faci la crida a la funció `jQuery.rssReaderParser` (inclosa en al propi javascript) per tal de fer el parsejat del fitxer obtingut.

Un cop es tenen les dades s'afegeixen a la llista de notícies que s'ha creat en el moment de crear la pàgina:

```
$('#noticies').append('<li data-role="list-  
divider">'+dia+dateTimeSplit[1]+mes+' de '...
```

i es fa el refresc de la llista, tornant el control a la plana.

### 5.2.4 Component agenda

El component de l'agenda és el descrit en el cas d'ús Consultar activitats.

L'estructura i el funcionament del component que obté les dades de l'agenda és la mateixa que la vista pel cas de les novetats, ja que es mostra també una llista de les darreres activitats publicades i aquesta informació s'obté de la mateixa manera.

De la mateixa manera que en el cas anterior, el control passa a la lògica de presentació al estar implementada la funció de control del pagecreate del component agenda:

```
$('#agenda').bind('pageinit', function() {
```

La crida a l'obtenció de l'XML es fa de la següent manera (a partir de les corresponents variables prèviament definides), afegint l'script de la crida:

```
// Definim l'espai on es farà la crida a l'api
var script = document.createElement('script');
script.setAttribute('type', 'text/javascript');
script.setAttribute('charset', 'utf-8');
script.setAttribute('src', googUrlAgenda);
document.body.appendChild(script);
```

Per fer el parsejat de l'XML retornat es recorren els ítems retornats i s'agafen les dades a mostrar per pantalla:

```
var entries = data.feed.entries;
for (var i=0; i<entries.length; i++) {
```

i agafem els valors dels camps a mostrar:

- entries[i].publishedDate -> Data de publicació
- entries[i].link -> URL assignada a l'event
- entries[i].title -> Descripció de l'event
- entries[i].author -> Qui n'és l'autor

### 5.2.5 Component newsFoto

El component newsFoto és el que s'ha descrit en el cas d'ús Consultar notícies.

Es va decidir implementar aquesta funcionalitat, donat que la possibilitat de mostrar les notícies corporatives es veia limitada, únicament, a mostrar les tres darreres novetats. S'ha implementat un component que en aquest cas obté les dades d'una base de dades local creada per a aquest objectiu.

En la capa de presentació es fa la gestió de preparar la plana per a mostrar les dades i es fa la crida cap a la capa de negoci per tal d'obtenir les dades.

De la mateixa manera que en els casos anteriors, l'estructura de la pàgina únicament crea el component `<ul id="llistaNoticies"...">` que s'omplirà amb les dades consultades a la base de dades.

Un cop es carrega la plana, el control passa a la següent funció:

```
$('#newsFoto').bind('pageinit', function() {  
  getNoticiesList();  
});
```

En aquest la funció `getNoticiesList()` serà l'encarregada de fer la crides cap a la capa de negoci per a obtenir la informació buscada.

Per obtenir aquestes dades s'utilitza la funció `$.getJSON`:

```
$.getJSON(serviceURL + 'getnoticies.php', function(data) {
```

Aquesta funció fa una petició de dades al servidor considerant que retorna la informació en notació JSON. En el segon paràmetre de la crida indiquen quina serà la funció que rebrà les dades, que en aquest cas s'implementa directament. El tractament que es fa un cop s'han rebut les dades és el següent:

```
// Recorrem els items retornats  
llistanoticies = data.items;  
$.each(llistanoticies, function(index, noticies) {  
  // Afegim a la llista de notícies la informació retornada  
  $('#llistaNoticies').append('<li style="white-space: normal;"><a  
href="' + noticies.url + '">' +  
'' +  
'<p>' + noticies.titol + '</p>' +...
```

I finalment s'actualitza la llista de notícies retornant el control a la pantalla.

Al afegir les notícies al component `<ul>` s'indica que el contingut sigui linkable cap a la URL, que està emmagatzemada a la base de dades, i que mostra el contingut estès de la notícia en el portal corporatiu.

## 5.2.6 Component seleccioPersones

El component `seleccioPersones` es correspon al cas d'ús descrit amb el nom de Cercar persones.

Quan l'usuari escull l'opció de consultar el directori de persones de la URV inicialment es mostra una pantalla on cal indicar els criteris amb els que es vol fer la cerca de persones.

Tal i com es va indicar ha sigut necessari implementar un servei que retornés les dades a mostrar en aquesta consulta ja que l'accés al directori corporatiu no ofereix serveis per a consultar-ne les dades. Per tant, s'ha creat una taula amb dades fictícies de persones assignades al PAS (personal d'administració i serveis) i al PDI (personal docent i investigador). Per una restricció de la LOPD no és possible accedir a dades dels estudiants fora del domini de la universitat, de manera que s'ha descartat, afegir les

dades d'estudiants a la base de dades ja que es tracta d'una implementació que no es portarà mai a terme.

Dins de la secció container es crea un formulari on li indiquem que com a acció obri la plana **#persones** que està implementada en la pròpia plana enlloc d'indicar una plana o programa remot. Per aquest motiu és necessari indicar que no es vol que s'utilitzi Ajax per a resoldre l'acció (cal indicar data-ajax="false"). En jQuery Mobile els submits dels formularis són gestionats directament utilitzant Ajax, creant una transició entre el formulari i la plana de destí.

Per altra banda s'incorpora el camp de text on es permetrà indicar la cadena de cerca en els camps de nom o cognom i un desplegable per indicar si es vol fer la cerca tenint en compte el tipus de personal. Al executar el botó de Cerca es llença l'acció i com a conseqüència s'activa la plana **#persones**.

De la mateixa manera que hem vist en els casos anteriors, al activar-se la plana **#persones** el control passa al javascript **persones.js** que és el que té implementada la funció:

```
$('#persones').bind('pageinit', function(event) {  
    var filtre = getUrlVars()["filtre"];  
    var cadena = getUrlVars()["search"];  
    getLlistaPersones(filtre, cadena);  
});
```

S'obtenen els paràmetres que s'han passat des de la plana de d'introducció dels paràmetres de cerca i es fa la crida a la funció que obtindrà la llista de persones.

En aquest cas també es fa la crida de la funció \$.getJSON indicant el PHP que s'ha d'executar amb els paràmetres recollits:

```
$.getJSON(serviceURL +  
'getpersones.php?filtre='+filtre+'&cadena='+cadena, function(data)  
{
```

D'igual manera es tracten els items obtinguts després de la crida afegint el tractament de comprovar que si canvia la inicial del cognom per tal d'afegir un separador per donar-li l'estètica d'agenda (al afegir el tag de llista <li> s'indica, cada cop que es vol crear una separació, que es tracta d'un **list-divider**: <li data-role="list-divider">).

```
$('#llistaPersones').append('<li data-role="list-divider">' +  
inicialactual.toUpperCase() + '</li>');
```

Al afegir a la llista les persones resultat de la consulta, s'indica que l'element serà linkable i la URL serà:

```
$('#llistaPersones'  
) .append('<li><a href="detall_persones.html?id=' + persones.id +  
'">...
```

de manera que quan es seleccioni alguna persona de la llista, es passarà a executar la plana **detall\_persones.html** passant-li com a paràmetre l'id de la persona seleccionada. Al cridar la plana **detall\_persones** passa a executar-se el javascript **detallpersones.js** com a conseqüència de tenir implementat el mètode

```
$('#detallsPersona').live('pageshow', function(event) {
```

que en aquest cas s'executa quan rep l'event **pageshow** ja que si es fa en el **pageinit** encara no s'han obtingut les dades de detall que es volen mostrar.

Un cop es recullen les dades, s'actualitza el contingut de la plana amb la informació continguda a la base de dades.

```
$('#nom').text(persona.nom + ' ' + persona.cognom1 + ' ' +  
persona.cognom2);  
$('#email').text(persona.email);
```

### 5.2.7 Component seleccioCentres

Aquest component es correspon al cas d'ús de Localitzar centres.

També ha estat necessari implementar el servei de consulta d'informació dels centres, ja que tampoc era un servei públic de la Universitat. Així doncs, s'ha creat una taula amb la informació dels centres per tal de poder-ne trobar la seva ubicació en un mapa. Un cop l'usuari estableix els criteris de filtrat dels centres que vol localitzar, o bé indicant la localitat o bé si els vols classificar pel campus on estan assignats, es procedeix a obtenir les dades dels centres i a llistar-les per pantalla. Quan l'usuari selecciona el criteri, el control de l'execució passa al javascript **centres.js** que s'activa al rebre l'event d'activació:

```
$('#paginaLlistaCentres').bind('pageinit', function(event)  
{  
  var filtre = getUrlVars()["radio-choice-1"];  
  getCentreList(filtre);  
});
```

Des de la pantalla de selecció de centres se li ha passat l'opció seleccionada per l'usuari que es la que es passarà cap al PHP encarregat de demanar les dades del centre

De manera similar a la que hem vist pel tractament de persones des de la funció **getCentreList** es genera una llista amb els centres resultat de la cerca. Els centres s'obtenen també en format JSON després de cridar al servei **getcentres.php**.

```
$('#llistaCentres').append('<li><a href="detall_centres.html?id=' +  
centres.id + '>' +  
<p>' + centres.nom + ' ' + centres.adreca +  
<br>' + centres.telefon + </p> +  
</a></li>');
```

Al crear la llista de centres, generem cada entrada de la llista com a link de manera que posteriorment es podrà seleccionar per accedir-ne al detall que trobem implementat en

la finestra **detall\_centres.html**. Al activar-se la plana de detall, el control passa al javascript **detallcentre.js** que tot seguit veurem com mostra el mapa on està ubicat el centre. Després de fer la crida al PHP que ens retornarà la informació del centre seleccionat a la pantalla anterior, indiquem que s'executi la funció `mapaCampus` que serà l'encarregada d'obtenir el mapa.

```
$.getJSON(serviceURL + 'getcentre.php?id='+id, mapaCampus);
```

El primer que fa la funció `mapaCampus`, a partir de la informació que obté de la base de dades, és crear un punt en coordenades geogràfiques (latitud i longitud) mitjançant la funció `google.maps.LatLng`. Un cop s'ha creat el punt s'utilitza el constructor `gmap` (proporcionat pel plugin de google map per a jquery i jquery mobile) per crear l'objecte on presentarem el mapa per pantalla. L'objecte el tenim definit a la plana:

```
<div id="map_canvas" style="width:270px;height:300px"></div>
```

A `gmap` li passem els següents paràmetres:

`Center` -> serveix per indicar en quin punt volem que es situï el centre del mapa.  
`Zoom` -> per indicar amb quin zoom inicial s'obre el mapa.  
`mapTypeId` -> serveix per indicar quina vista es vol per defecte del mapa: híbrida, satèl·lit o de carrers.  
`Callback` -> per indicar la funció que volem que es cridi un cop s'ha carregat el mapa. En el nostre cas un cop s'ha carregat el mapa, afegim un marcador (amb l'indicador de punt del google maps) que ens permetrà, que al fer click, s'obri a finestra amb la informació addicional del centre. Això s'aconsegueix amb la funció `openInfoWindow`.

### 5.2.8 Component informacióInstitucional

Donat el fet que s'han creat taules amb la informació de tots els centres de la universitat s'ha considerat que es podia afegir una funcionalitat que permetés consultar la informació de diverses maneres (i així aportar una visió més institucional de les dades. Si es consulten webs de diferents universitats, es pot veure que es dona molta importància (o si més no visibilitat) de la informació institucional. De manera que s'agrupen els centre tecnològics, els d'investigació... per donar una visió més acurada del que s'està fent en cada centre.

A més, a les webs, també s'incorpora l'estructura jeràrquica dels responsables de la universitat, dins del mateix apartat fet que ens ha portat a afegir part de la informació institucional corporativa en una nova taula per tal de proporcionar-la també des de l'aplicació.

Al accedir a aquest apartat d'informació institucional es carrega la pàgina institució (dins de la pàgina principal), que mostra la llista de la informació que es pot consultar:

```
<div data-role="content">  
<ul data-role="listview" class="llista" data-insets="true"  
data-theme="a" id="itemInstitucio">
```

```
<li><a href="#rektorat"><p>Òrgans de Govern</p></a></li>
<li><a href="#paginaLlistaInstitucions"><p>Campus</p></a></li>
<li><a href="#paginaLlistaInstitucions"><p>Facultats i
Escoles</p></a></li>
<li><a href="#paginaLlistaInstitucions"><p>Instituts i Centres
de Recerca</p></a></li>
</ul>
</div>
```

La primera de les opcions que es mostra, Òrgans de govern, permet accedir a la classificació de les dades de persones del directori institucional, que estaran agrupades en Rectorat / Vicerectorat / Secretaria General.

Al seleccionar l'opció de Rectorat, apareix una nova llista de selecció per tal s'escollir si es vol accedir a la informació del Rector, Secretaria del rektorat...

El que es veu és que un cop s'ha arribat a l'últim nivell dins de l'estructura institucional, es crida la plana **info\_directori.html** indicant-li per paràmetre el grup del que se'n volen obtenir les persones que en formen part.

```
$('#infoPersones').live('pageshow', function(event) {
var g = getUrlVars()["grup"];
getDadesGrup(g);
});
```

Al mostrar-se la plana es produeix l'event pageshow que fa que el control passi al fitxer **infodirectori.js** que és on estan implementades les funcions per fer les crides per a obtenir les dades:

```
function getDadesGrup(grup) {
$.getJSON(serviceURL + 'getfromdirectori.php?grup='+grup,
function(data) {
llistapersones = data.items;
var cadena = "";
$.each(llistapersones, function(index, persones) {
cadena = cadena + '<p>' + persones.nom + '<br><b>' +
persones.tipus_persona + '</b><br>' +
persones.adreca + '<br>' + persones.cp + '<br>' +
persones.email + '</p><br><br>';
});
$('#infoPersonesDetall').append(cadena);
});
}
```

Es recorre la llista de persones retornades del grup i es mostren totes en el div **infoPersonesDetall**.

### 5.2.9 Component accés URV

Proporciona un enllaç a la web de la universitat: [www.urv.cat](http://www.urv.cat)

## 5.3 Capa de negoci

La capa de negoci, que és on resideixen els programes que s'executen, és on es reben les peticions de l'usuari. Com ja hem vist en la descripció de la implementació de la capa de presentació hi ha varis programes PHP per satisfer les peticions de l'usuari que tot seguit es descriuen.

### 5.3.1 Visió general PHP

PHP és un llenguatge de programació orientat al desenvolupament d'aplicacions web dinàmiques amb accés a informació emmagatzemada en una base de dades. El codi font escrit en PHP és visible al navegador web i al client ja que és el servidor el que s'encarrega d'executar el codi i enviar el resultat HTML al navegador.

En exemple molt senzill és el següent:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>PHP Test</title>
  </head>
  <body>
    <?php
      echo 'Hello World';
    ?>
  </body>
</html>
```

Ofereix capacitat de connexió amb la majoria dels motors de bases de dades que s'utilitzen actualment, destacant la seva connectivitat amb MySQL i PostgreSQL.

En el nostre cas els programes desenvolupats per a l'aplicació requereixen accedir a la base de dades on s'han definit les taules amb la informació dels centres, notícies... de manera que s'ha definit un fitxer de configuració config.php que inclouran tota la resta de fitxers per contenir la definició de l'accés a la base de dades.

```
<?php
$dbhost = '127.0.0.1';
$dbuser = 'root';
$dbpass = '';
$dbname = 'iURV';
?>
```

on:

\$dbhost és el nom del servidor MySQL,  
\$dbuser serà l'usuari de la base de dades,  
\$dbpass serà la password de l'usuari indicat  
\$dbname és el nom de la base de dades.



### 5.3.2 Component Gestió Notícies

Des de la capa de presentació es fa la sol·licitud de la llista de notícies a través de la crida a **getnoticies.php**.

Aquest programa accedeix a la taula notícies i en retorna els seu contingut.

```
<?php
header('Content-type: application/json');

include 'config.php';

$sql = "SELECT titol, resum, imatge, url FROM noticies";

try {
    $dbh = new PDO("mysql:host=$dbhost;dbname=$dbname", $dbuser,
    $dbpass,array(PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES
    'utf8'"));
    $dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $stmt = $dbh->query($sql);
    $noticies = $stmt->fetchAll(PDO::FETCH_OBJ);
    $dbh = null;
    echo '{"items":'. json_encode($noticies) .'}';
} catch(PDOException $e) {
    echo '{"error":{"text":'. $e->getMessage() .'}}';
}

?>
```

Per tal d'accedir a les dades s'utilitza l'extensió d'Objectes de Dades de PHP (PDO) que defineix una interfície per a tenir accés a les bases de dades des de programes implementats en PHP. Proporciona una capa d'abstracció del motor de bases de dades al que s'accedeix de manera que l'accés a les dades és independent de la base de dades que s'està utilitzant.

Un cop s'ha connectat a la base de dades sobre la que es vol accedir s'executa la query amb la que volem obtenir les dades. Am la funció `fetchAll` carreguem en l'string de notícies els files obtingudes i finalment convertim l'string a format JSON amb la funció `json_encode`.

### 5.3.3 Component Gestió Persones

Des de la capa de presentació es fan dues crides relacionades amb la gestió de persones. La crida **getpersones.php** retorna la llista de persones a partir dels filtres introduïts i la crida **getpersona.php** retorna el detall de la persona seleccionada de la llista anterior.

El primer que es fa dins del programa `getpersones.php` és recollir els paràmetres de filtrat que es volen aplicar a la cerca de manera que es pugui construir la query que permeti obtenir les persones que compleixen el filtre indicat.

```
$tipus_persona = $_GET['filtre'];
$cadena = $_GET['cadena'];
```

Un cop es té la query que s'ha d'executar, el procediment serà el mateix que hem vist en l'apartat anterior, és a dir, s'estableix la connexió a la base de dades, s'executa la query i finalment es retorna el resultat en format JSON per tal d'ésser interpretat a la capa de presentació.

Per altra banda, si l'usuari decideix veure el detall d'alguna persona, passa a executar-se el programa `getpersona.php` que rebrà com a paràmetre l'identificador (id) de la persona seleccionada i l'utilitzarà directament com a filtre de la query.

```
try {
    $dbh = new PDO("mysql:host=$dbhost;dbname=$dbname", $dbuser,
    $dbpass,array(PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES
    'utf8'"));
    $dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $stmt = $dbh->prepare($sql);
    $stmt->bindParam("id", $_GET[id]);
    $stmt->execute();
    $persona = $stmt->fetchObject();
    $dbh = null;
    echo '{"item":'. json_encode($persona) .'}';
} catch(PDOException $e) {
    echo '{"error":{"text":'. $e->getMessage() .'}}';
}
```

En aquest cas per recollir el resultat de la cerca es fa amb la funció `fetchObject` ja que només ens podrà retornar una fila que compleixi la cerca.

### 5.3.4 Component Gestió Centres

Des de la capa de presentació es fan dues crides relacionades amb la gestió de centres. La crida **`getcentres.php`** retorna la llista de centres a partir dels filtres introduïts i la crida **`getcentre.php`** retorna el detall del centre seleccionat de la llista anterior. El funcionament, per tant, és molt similar al del component de gestió de persones que hem vist en l'apartat anterior.

Tal i com hem descrit en la descripció d'implementació de la capa de presentació, s'ha afegit una funcionalitat, per tal d'aprofitar les dades que hi han introduïdes a la base de dades per mostrar la informació d'una manera més "institucional". D'aquesta manera el programa `getcentres.php` també ens servirà per obtenir els centres agrupats per criteris: Facultats i Escoles, Instituts de Recerca...

Per tant, a l'hora de definir els filtres de selecció de centres haurem de considerar els dos orígens.

```
$poblacio = $_GET['filtre'];

if ($poblacio=='Totes') {
    $sql = "select p.id, p.nom, p.adreca, p.telefon, p.poblacio,
    p.campus, p.tipus_centre " .
    "from centres p order by p.nom";
```

```
}
else {
if ($poblacio=='Campus') {
$sql = "select p.id, p.nom, p.adreca, p.telefon, p.poblacio,
p.campus, p.tipus_centre " .
"from centres p where p.campus <> 'null' order by p.campus";
}
else if ($poblacio=='Facultats') {
$sql = "select p.id, p.nom, p.adreca, p.telefon, p.poblacio,
p.campus, p.tipus_centre " .
"from centres p where p.tipus_centre='1' or p.tipus_centre='2'
order by p.tipus_centre";
}
else if ($poblacio=='Instituts'){
$sql = "select p.id, p.nom, p.adreca, p.telefon, p.poblacio,
p.campus, p.tipus_centre " .
"from centres p where p.tipus_centre='4' or p.tipus_centre='5' or
p.tipus_centre='6' order by p.tipus_centre";
}
else {
$sql = "select p.id, p.nom, p.adreca, p.telefon, p.poblacio,
p.campus, p.tipus_centre " .
"from centres p where p.poblacio='$poblacio'";
}
}
```

Per últim, el retorn de la llista de centres, serà tal i com ja hem vist descrit anteriorment:

```
echo '{"items":'. json_encode($centres) .'}';
```

El detall del centre ens servirà per obtenir les coordenades que utilitzarem per mostrar el mapa i la informació que es mostrarà en la finestra de detall.

```
<?php
include 'config.php';

$sql = "select p.id, p.nom, p.adreca, p.poblacio, p.CP, p.telefon,
p.coordX, p.coordY " .
"from centres p where p.id=:id";
```

### 5.3.5 Component Gestió Institucions

Per la nova funcionalitat d'informació de la institució, a banda de mostrar els centres classificats segons la seva tipologia, també mostra informació del directori institucional. Aquesta informació s'ha guardat en una taula nova per tal de facilitar-ne tant la consulta com el manteniment (malgrat sigui una informació relativament estàtica).

Amb aquest objectiu s'ha desenvolupat el programa **getfromdirectori.php** que és l'encarregat d'accedir a la taula directori per a obtenir les dades a buscar. Es crida a aquest PHP amb dos objectius, buscar les persones assignades a un grup o buscar grups per tal de poder-los llistar:

```
if ($grup=='Vice') {
$sql = "SELECT p.nom_institucio, p.grup FROM directori p where
p.grup like \"%Vice%\"";
```

```
}  
else {  
$sql = "SELECT p.nom, p.email, p.adreca, p.telefon, p.cp,  
p.tipus_persona FROM directori p where p.grup='$grup';"  
}
```

## 5.4 Capa de dades

Les taules que s'han creat per donar suport a les funcionalitats implementades són les de notícies, persones, centres i directori. Les sentències de creació d'aquestes taules són les que es mostren a continuació.

L'accés que es fa a aquestes dades és, però, únicament de consulta i es fa a través de les funcions de PHP implementades i comentades en l'apartat de capa de negoci.

Tot seguit s'adjunta la definició d'aquestes taules:

```

delimiter $$

CREATE TABLE `centres` (
  `id` int(11) NOT NULL,
  `nom` varchar(100) NOT NULL,
  `adreca` varchar(100) NOT NULL,
  `poblacio` varchar(45) DEFAULT NULL,
  `CP` varchar(45) DEFAULT NULL,
  `telefon` varchar(45) DEFAULT NULL,
  `coordX` varchar(45) NOT NULL,
  `coordY` varchar(45) NOT NULL,
  `campus` int(11) DEFAULT NULL,
  `tipus_centre` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1$$

delimiter $$

CREATE TABLE `noticies` (
  `id` int(11) NOT NULL,
  `titol` varchar(100) COLLATE utf8_spanish_ci DEFAULT NULL,
  `resum` varchar(100) COLLATE utf8_spanish_ci DEFAULT NULL,
  `url` varchar(100) COLLATE utf8_spanish_ci DEFAULT NULL,
  `imatge` varchar(100) COLLATE utf8_spanish_ci DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci$$

delimiter $$

CREATE TABLE `persones` (
  `id` int(11) NOT NULL,
  `nom` varchar(45) NOT NULL,
  `cognom1` varchar(45) NOT NULL,
  `cognom2` varchar(45) DEFAULT NULL,
  `email` varchar(45) DEFAULT NULL,
  `telefon` varchar(45) DEFAULT NULL,
  `tipus_persona` varchar(45) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1$$

delimiter $$

CREATE TABLE `directori` (
  `id` int(11) NOT NULL,
  `nom` varchar(200) COLLATE utf8_spanish_ci NOT NULL,
  `adreca` varchar(100) COLLATE utf8_spanish_ci DEFAULT NULL,
  `cp` varchar(50) COLLATE utf8_spanish_ci DEFAULT NULL,
  `telefon` varchar(15) COLLATE utf8_spanish_ci DEFAULT NULL,
  `email` varchar(100) COLLATE utf8_spanish_ci DEFAULT NULL,
  `grup` varchar(100) COLLATE utf8_spanish_ci NOT NULL,
  `tipus_persona` varchar(100) COLLATE utf8_spanish_ci NOT NULL,
  `nom_institucio` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci$$

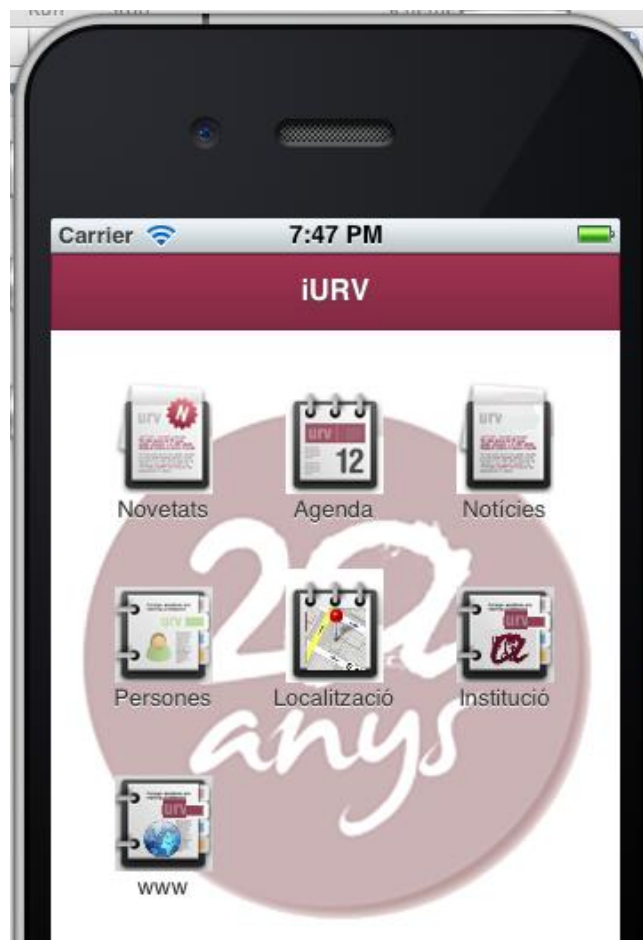
```

Il·lustració 24. Estructura de les taules

## 5.5 Implementació pantalles de l'aplicació

Tot seguit s'adjunten una sèrie de captures de pantalla de l'aplicació en execució.

Pantalla principal de l'aplicació: és el punt d'entrada a la mateixa i des d'on l'usuari selecciona la funcionalitat a executar: consulta novetats, consulta notícies, consulta agenda, consulta de persones, localització de centres, informació institucional i accés a la web corporativa.



Il·lustració 25. Pantalla principal aplicació

**Pantalla Novetats:** Mostra de la pantalla de consulta de les darreres novetats. Es mostra la llista confeccionada a partir d'un xml, mostrant la data de publicació, la descripció o resum de la mateixa i el departament o grup responsable de la publicació.



II-lustració 26. Pantalla Novetats

**Pantalla Agenda:** Mostra de la pantalla de consulta de les darreres activitats publicades. Es mostra la llista confeccionada a partir d'un xml, mostrant la data de publicació, la descripció o resum de la mateixa i el departament o grup responsable de la publicació.



II-lustració 27. Pantalla Activitats



**Pantalla Notícies:** mostra la llista de notícies que es tenen emmagatzemades a la base de dades del servidor local.



II-lustració 28. Pantalla Notícies

**Pantalla Detall notícia:** un cop es mostra la llista de notícies es pot accedir al detall de cadascuna d'elles a partir de la URL que també està emmagatzemada a la base de dades.



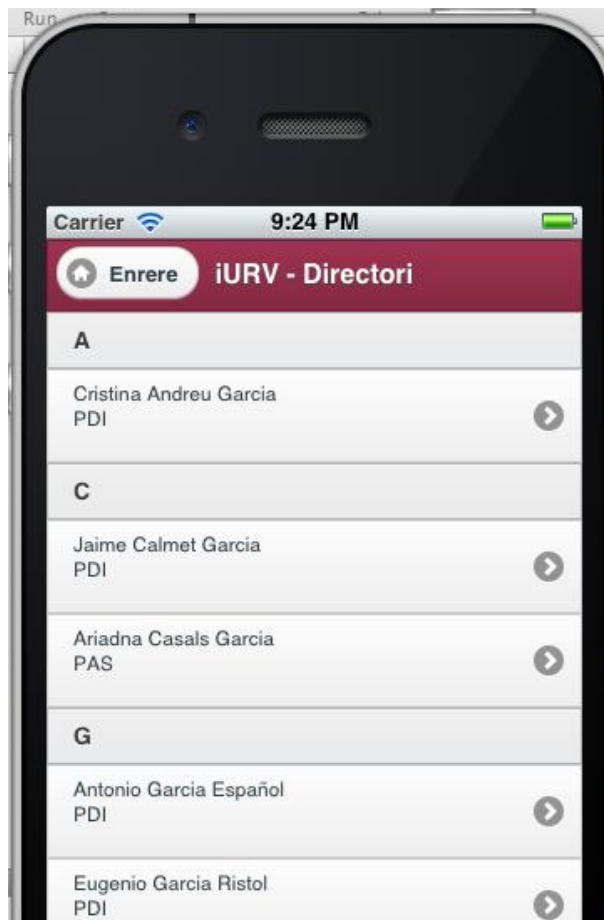
II-lustració 29. Detall notícia

**Pantalla d'introducció dels criteris de cerca de persones:** mostra la pantalla on es pot determinar quins criteris de cerca es volen utilitzar per localitzar persones de la institució. O bé localitzant alguna cadena o bé indicant el tipus de personal que es cerca o ambdós casos.



II-lustració 30. Pantalla introducció criteris de cerca

**Pantalla llista de persones:** mostra la llista de persones consultades a la base de dades a partir dels criteris introduïts.



II-Il·lustració 31. Pantalla Llista de persones

**Pantalla de detall de la informació de les persones:** mostra la informació guardada a la base de dades de la persona seleccionada.



II-lustració 32. Pantalla Detall d'informació de la persona

**Pantalla llista de centres:** mostra la llista de centres obtinguts de la base de dades.



II-lustració 33. Llistat de centres

**Pantalla de detall de la informació d'un centre:** un cop es selecciona un centre de la llista de centres, es passa a la finestra de detall on es pot visualitzar el mapa de l'ubicació del centre així com obtenir-ne el detall al clicar sobre el punt de localització.



Il·lustració 34. Mapa d'un centre

**Pantalla informació institucional:** pantalla on es mostra la informació de la universitat agrupada en criteris institucionals. Proporciona l'accés als centres i la informació relativa als òrgans de govern que també està emmagatzemada a la base de dades.



Il·lustració 35. Informació institucional



Il·lustració 36. Detall dades institucionals



## 6 Conclusions

El present projecte ha servit per veure la complexitat de crear aplicacions per a mòbil i per a assentar les bases per a futurs desenvolupaments a portar a terme per part de la universitat.

L'estudi dels diferents frameworks i l'ús dels mateixos, ha proporcionat una visió de les possibles arquitectures a utilitzar en aquests àmbits, aprofundint en un dels mateixos.

La capacitat de desenvolupar aplicacions web amb HTML5, CSS3 i JavaScript permet preveure possible qualsevol desenvolupament a realitzar. A més, la integració amb altres serveis a través d'AJAX, en destaca la navegabilitat i usabilitat.

A banda de la part d'aplicació per a mòbil, també ha estat necessari desenvolupar serveis addicionals per donar suport a determinades funcionalitats inexistents en l'actualitat que al estar desenvolupades amb eines open source es considera factible integrar-les en els sistemes de la pròpia universitat (creant una infraestructura específica pel canal mòbil, que inclouria la rèplica de les dades corporatives sobre la nova infraestructura).

Queden com a línies obertes, doncs, traslladar l'aplicació a l'entorn d'explotació de la pròpia universitat així com la compilació de l'aplicació per a altres plataformes (que seria un requeriment en el moment de posar l'aplicació a producció).

## ANNEX 1 – Glosari

Tot seguit es mostren els termes, abreviatures i acrònims necessaris per a la correcta comprensió d'aquest document.

**AJAX** – Asynchronous Javascript And Xml, és un conjunt de tecnologies que permeten actualitzar continguts web sense haver de tornar a carregar la pàgina. Permet crear aplicacions web asíncrones.

**CSS**- Cascading Style Sheets, és un llenguatge de fulls d'estil utilitzat per descriure la semàntica de presentació d'un document escrit en un llenguatge de marques.

**HTTP** - HyperText Transfer Protocol, protocol de transferència hipertext, utilitzat principalment per a l'accés a contingut web des de navegadors.

**IDE** – Integrated Development Environment, és una eina informàtica per al desenvolupament de programari de manera còmoda i ràpida.

**JS** – Javascript, és un llenguatge de programació interpretat. S'utilitza principalment en la seva forma del costat del client (client-side).

**JSON** – JavaScript Object Notation, és un format lleuger per a l'intercanvi de dades. És un estàndard obert basat en text dissenyat per a intercanvi de dades llegibles per humans.

**LOPD** – Llei Orgànica de Protecció de Dades.

**MVC** – Model-View-Controller, patró d'arquitectura software model-vista-controlador, que proposa la separació de la presentació, lògica de control i de negoci en components independents.

**MySQL** – és un sistema de gestió de bases de dades relacional, multithread i multiusuari.

**PAS** – Personal d'Administració i Serveis.

**PDI** – Personal Docent i Investigador.

**PDO** – PDP Data Objects, és una extensió que proveeix una capa d'abstracció d'accés a dades per a PHP5, de manera que s'usen les mateixes funcions per a fer consultes i obtenir dades de diferents manegadors de base de dades.

**PHP** – PHP Hypertext Pre-Processor, és un llenguatge de programació interpretat, dissenyat originàriament per a la creació de pàgines web dinàmiques.

**SGDB** – Sistema Gestor de Bases de Dades, software especialitzat en les tasques d'emmagatzemament i recuperació d'informació, principalment segons el model relacional.

**SQL** – Structured Query Language, llenguatge estructurat de consulta utilitzat per a la manipulació d'informació emmagatzemada en un SGDB.

**UI** – User Interface, és el medi amb què l'usuari pot comunicar-se amb una màquina, un equip o una computadora, i comprèn tots els punts de contacte entre l'usuari i l'equip; normalment solen ser fàcils d'entendre i fàcils d'accionar.

**UML** – Unified Model Language, llenguatge unificat de modelat de sistemes d'informació i components.

**URL** – Uniform Resource Locator, localitzador uniforme de recursos o adreça única d'un element a Internet.

**URV** – Universitat Rovira i Virgili. Universitat pública de Tarragona.

**XML** – Extensible Markup Language, llenguatge extensible de marques multipropòsit, molt utilitzat per a l'intercanvi d'informació entre sistemes heterogenis.

## ANNEX 2 – Fonts consultades

<http://www.apple.com>

<https://developer.apple.com>

<http://www.sencha.com>

<http://www.wikipedia.com>

<http://www.phonegap.com>

<http://www.stackoverflow.com>

<http://m.stanford.edu/>

<https://developers.google.com/maps/documentation/javascript/reference>

<http://www.slideshare.net>

<http://jquerymobile.com/themeroller/>