

**Projecte Fi de Carrera**

**Meteo Andorra**

**{ Dades meteorològiques del Principat d'Andorra }**

**Universitat Oberta de Catalunya - UOC**

**Enginyeria en Informàtica**

**Autor: Sergi Sogas Rodríguez**

**Consultor: Sebastià Cortes Herms**

**Gener de 2014**

## **Agraïments i dedicatòries**

En primer lloc m'agradaria mostrar la meva gratitud al professor i responsable de l'assignatura Xavier Vilajosana i al consultor Sebastià Cortes de la Universitat Oberta de Catalunya, per haver-me deixat desenvolupar aquest projecte personal, així com pels seus consells i ajuda atorgats al llarg de tot aquest semestre.

També vull agrair personalment la tutora Montserrat Pellicer per tots els seus consells i suggeriments. No només pels mostrats durant el transcurs d'aquest segon cicle, sinó també pels dispensats durant els estudis d'Enginyeria Tècnica en Informàtica de Gestió que vaig cursar a la Universitat d'Andorra.

Finalment, vull dedicar aquest treball, als meus pares, a la meva germana i a la meva parella, per la seva paciència, ànims, i suport que m'han fet arribat durant tota aquesta etapa. Sense ells aquesta comesa no hauria estat possible.

## **Índex**

### **Memòria PFC**

1 - Introducció.....	pàg.	4
2 - Descripció i objectius del projecte.....	pàg.	5
2.1 - Desenvolupament i funcionament.....	pàg.	5
2.2 - Esquema de funcionament.....	pàg.	7
2.3 - Maquinari i programari utilitzat.....	pàg.	8
3 - Planificació del projecte.....	pàg.	9
3.1 - Planificació 1a part - Fase 1.....	pàg.	10
3.2 - Planificació 2a part - Fase 2.....	pàg.	10
3.3 - Planificació 2a part - Fase 3.....	pàg.	11
3.4 - Planificació 2a part - Fase de documentació.....	pàg.	12
3.5 - Diagrames de Gantt.....	pàg.	12
4 - Anàlisi i disseny.....	pàg.	14
4.1 - Actors de les aplicacions.....	pàg.	14
4.2 - Casos d'ús de l'aplicació web.....	pàg.	15
4.3 - Casos d'ús de l'aplicació Android.....	pàg.	17
4.4 - Classes d'entitat.....	pàg.	20
4.5 - Diagrames de seqüència de l'aplicació web.....	pàg.	22
4.6 - Diagrames de seqüència de l'aplicació Android.....	pàg.	26
5 - Implementació.....	pàg.	30
5.1 - Publicació les dades meteorològiques a la xarxa social Twitter.....	pàg.	30
5.2 - Desenvolupament de l'aplicació web.....	pàg.	31
5.3 - Desenvolupament del servei web.....	pàg.	35
5.4 - Desenvolupament de l'aplicació Android.....	pàg.	39
6 - Conclusions i valoracions.....	pàg.	45
6.1 - Aspectes a millorar.....	pàg.	46
6.2 - Autoavaluació.....	pàg.	47
7 - Bibliografia i pàgines web consultades.....	pàg.	48
8 - Glossari.....	pàg.	50

### **Annexos**

Annex 1: Aplicació web.....	pàg.	51
Annex 2: Aplicació Android.....	pàg.	55
Annex 3: Instal·lació d'un entorn de desenvolupament LAMP.....	pàg.	59
Annex 4: Programari wview.....	pàg.	65
Annex 5: Comprovació del sistema implementat.....	pàg.	70
Annex 6: Valoració econòmica.....	pàg.	75

## **1 - Introducció**

Aquest projecte consisteix en recuperar les dades d'una estació meteorològica i desenvolupar una sèrie d'aplicacions i serveis en diferents tecnologies, per consultar l'evolució d'aquestes dades en temps real.

Els serveis i aplicacions desenvolupats han estat:

- Publicació de les dades de l'estació a la xarxa social Twitter.
- Aplicació web per consultar les dades recollides per l'estació, mostrar els valors mínims i màxims recuperats, mitjanes, gràfics lineals comparatius, resums diaris, resums anuals, etc.
- Aplicació per a dispositius amb sistema operatiu Android, amb funcionalitats similars a l'aplicació web.

Al llarg d'aquest treball es descriu de forma detallada els objectius del projecte i el seu funcionament, el maquinari utilitzat, s'explica la seva planificació i les fases en què ha estat dividit, es mostra l'anàlisi i el disseny de totes les aplicacions i serveis desenvolupats, així com les tecnologies utilitzades i els aspectes més rellevants de la seva implementació.

Finalment, en el darrer capítol s'exposa les conclusions, valoracions personals, i es realitza una autoavaluació del projecte.

## **2 - Descripció i objectius del projecte**

Bàsicament el projecte consisteix en recuperar i emmagatzemar les dades recollides pels diferents sensors d'una estació meteorològica (sensors de temperatura, humitat i pressió atmosfèrica), i desenvolupar tot un seguit d'aplicacions i serveis que permeten consultar l'històric d'aquestes dades, així com seguir l'evolució en temps real.

Les aplicacions i serveis desenvolupats han estat:

1) Servei de publicació de les dades recollides per l'estació meteorològica a la xarxa social Twitter.

2) Aplicació web multilingüe (en català, castellà, francès i anglès) que permet:

- Consultar les darreres dades recollides pels sensors de l'estació meteorològica,
- Consultar un resum diari de les dades recollides per l'estació, amb gràfics comparatius respecte al dia anterior per veure'n l'evolució, i presentació dels valors mínims i màxims diaris,
- Consultar un resum anual de les dades recollides per l'estació, amb gràfics de les mitjanes mensuals, i presentació dels valors mínims i màxims mensuals.

3) Aplicació per a terminals amb sistema operatiu Android, amb funcionalitats similars a l'aplicació web.

### **2.1 - Desenvolupament i funcionament**

#### **1) Recollida i emmagatzematge de les dades meteorològiques**

L'estació meteorològica està connectada a un equip que funciona 24/7 (24 hores al dia, 7 dies a la setmana), per tal de poder recuperar les dades de forma ininterrompuda.

Dit equip funciona amb un sistema operatiu GNU/Linux i té instal·lat el programari per a sistemes UNIX wview (<http://www.wviewweather.com/>), que s'encarrega de connectar-se a l'estació meteorològica i arxivar les dades dels sensors en una base de dades SQLite3.

#### **2) Publicació de les darreres dades emmagatzemades a la xarxa social Twitter**

S'ha programat un script en llenguatge Python que es connecta a la base de dades SQLite3, recupera les darreres lectures i publica els valors recuperats a la xarxa Twitter.

Dit script està instal·lat a l'equip connectat a l'estació meteorològica i s'executa de forma periòdica (per exemple, cada quatre hores) mitjançant l'administrador de processos *cron*.

### 3) Desenvolupament de l'aplicació web (entorn de proves)

Aquest punt ha estat dividit en dues fases:

- a) Instal·lació d'un servidor de proves similar al servidor de producció (entorn real), per tal de poder desenvolupar l'aplicació web de forma local.

El servidor de proves consisteix en el servidor de pàgines web Apache2 amb el mòdul PHP5, i el servidor de bases de dades MySQL, entre d'altres. El procés d'instal·lació d'aquest entorn està descrit en el darrer annex d'aquest treball.

- b) Desenvolupament de l'aplicació web utilitzant el llenguatge PHP i l'eina de treball per a aplicacions web Zend Framework.

### 4) Publicació de l'aplicació web en un entorn de producció (entorn real)

Una vegada acabat el desenvolupament de l'aplicació web, aquesta ha passat a publicar-se en un allotjament web professional que garanteix el bon funcionament de l'aplicació, i que compleix amb uns bons requisits de disponibilitat. L'adreça real per poder consultar l'aplicació web és: <http://www.meteoandorra.com/>.

Per tal que l'aplicació web pugui mostrar les dades actuals de l'estació meteorològica, s'ha muntat un sistema de sincronització entre les bases de dades del servidor local i del servidor de producció. D'aquesta forma l'entorn de producció disposa de les darreres dades recollides per l'estació.

### 5) Desenvolupament d'un servei web per poder consultar les dades de l'estació

S'ha programat un servei web SOAP que permet que d'altres dispositius puguin connectar-se a l'aplicació de l'entorn de producció, i puguin consultar així les dades meteorològiques.

El servei web també s'ha programat en llenguatge PHP i l'eina de treball Zend Framework. D'aquesta manera pot funcionar en el mateix entorn de producció que s'executa l'aplicació web.

### 6) Desenvolupament de l'aplicació per a terminals amb sistema operatiu Android

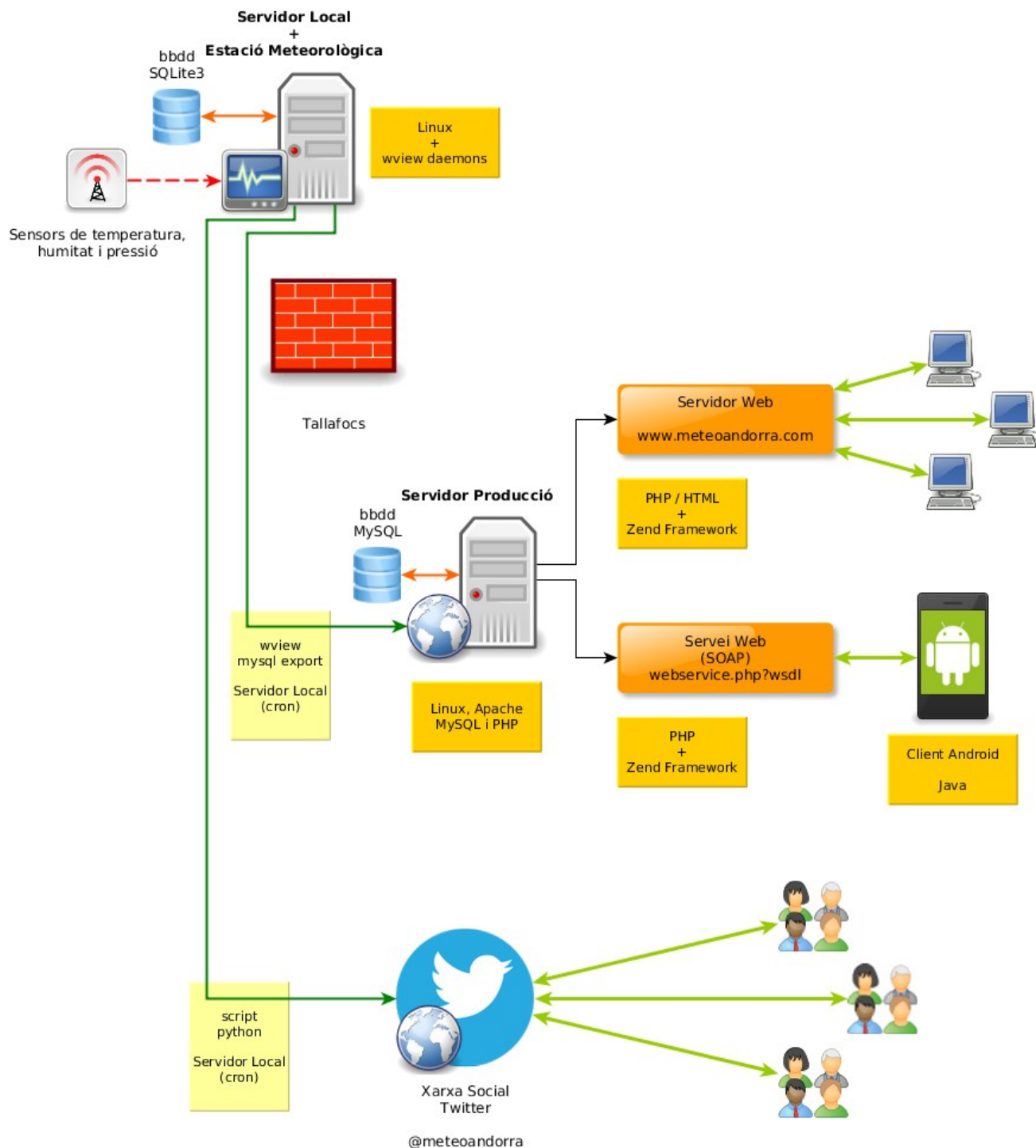
Finalment, s'ha desenvolupat una aplicació per a terminals Android amb funcionalitats similars a l'aplicació web.

De la mateixa manera que el desenvolupament de l'aplicació web, aquest punt també s'ha dividit en dues fases:

- a) Instal·lació de l'entorn de desenvolupament Android SDK (entorn de proves).
- b) Desenvolupament de l'aplicació i comprovació del correcte funcionament en un dispositiu Android real.

## 2.2 - Esquema general

Tots els punts descrits en l'apartat anterior poden resumir-se mitjançant el següent esquema de funcionament:



## **2.3 - Maquinari i programari utilitzat**

Tot seguit es detallen els diferents equips de maquinari i programari utilitzats durant el desenvolupament del projecte:

### Estació Meteorològica:

- Estació meteorològica Oregon Scientific WMR88 amb:
  - sensor de pressió atmosfèrica,
  - sensor remot exterior de temperatura i humitat.



### Entorn de desenvolupament:

- Portàtil Apple MacBook "Core Duo" - 2GB de RAM,
- Sistema Operatiu GNU/Linux Mint 14 "Nadia" 64-bit,
- Intèrpret de llenguatge Python,
- Servidor Apache2 + PHP5 i MySQL,
- Eclipse + PHP Development Tools,
- Eclipse + Android SDK.



### Terminal Android:

- Tablet ASUS Nexus 7 - 16GB,
- Sistema Operatiu Android 4.3 "Jelly Bean".



Tanmateix, els servidors local i de producció tindran la configuració següent:

### Servidor Local:

- Sistema operatiu GNU/Linux Ubuntu Server 12.04 LTS 64-bit,
- Intèrpret de llenguatge Python,
- Programari per a sistemes UNIX wview 5.20,
- Sistema de gestió de bases de dades SQLite3.

### Servidor Producció:

- Sistema operatiu GNU/Linux Ubuntu Server 12.04 LTS 64-bit,
- Servidor web Apache2 amb mòdul PHP5,
- Sistema de gestió de bases de dades MySQL.



### **3 - Planificació del projecte**

De forma general es pot dir que aquest projecte ha estat dissenyat tenint en compte dues parts ben diferenciades:

#### 1a part - Fase 1:

Aquesta primera part, que he anomenat Fase 1, va ser desenvolupada l'octubre de 2012 i va servir com a estudi inicial, per tal de mesurar l'impacte i l'interès del projecte.

Bàsicament consisteix en, una vegada adquirida l'estació meteorològica, connectar-la a un servidor local per començar a emmagatzemar les dades, i programar un petit script en llenguatge Python que recupera les darreres dades meteorològiques i les publica a la xarxa social Twitter amb el nom d'usuari *@meteoandorra*.

En el decurs d'un any el compte de *@meteoandorra* ha suscitat un interès considerable, superant la xifra 1000 seguidors interessats en conèixer les dades meteorològiques del Principat d'Andorra, i provocant un cert moviment i participació dins la xarxa social via retweets, favorits, consultes, etc.

Aquest interès ha promogut la idea de desenvolupar tot d'un seguit de serveis addicionals, per tal de poder consultar les dades meteorològiques recuperades. Aquests serveis addicionals han estat desenvolupats enguany durant la segona part del projecte.

#### 2a part - Fase 2, 3 i Documentació:

La primera millora consisteix en desenvolupar una pàgina web, que permeti consultar les dades meteorològiques des de qualsevol ordinador connectat a Internet. És el que he anomenat Fase 2 del projecte.

La segona millora consisteix en desenvolupar un servei web públic per a que d'altres dispositius, com per exemple *smartphones* o *tablets*, puguin consultar l'històric de les dades meteorològiques mitjançant una aplicació client. És el que he anomenat Fase 3 del projecte, la qual inclou també la programació d'un client per a dispositius amb sistema operatiu Android.

Per últim, durant la Fase de Documentació s'ha documentat tot el procés de realització del projecte, mitjançant una memòria de treball i una presentació resum.

Tot seguit es detalla la planificació temporal de cadascuna de les fases, així com les diferents tasques en què s'han dividit.



### 3.1 - Planificació 1a part - Fase 1

Com acabo de relatar la Fase 1 ha consistit en connectar l'estació meteorològica a un servidor local, i en programar un petit script per publicar les darreres dades recuperades a la xarxa social Twitter. Va ser desenvolupada a l'octubre de 2012 i la seva durada va ser d'aproximament de dues setmanes.

En el següent esquema de planificació podem veure les tasques en que fou dividida:

Name	Begin date	End date	Duration
[-] • Fase 1	10/1/12	10/17/12	13
• Estudi de requisits	10/1/12	10/1/12	1
• Llegir documentació wview	10/2/12	10/5/12	4
• Comprar estació meteorològica	10/4/12	10/5/12	2
• Instal·lació wview/estació (local)	10/8/12	10/8/12	1
• Proves funcionament wview/estació (local)	10/9/12	10/10/12	2
• Exportació de dades estació MySQL (local)	10/9/12	10/9/12	1
• Programació script dades twitter	10/10/12	10/11/12	2
• Proves script dades twitter (local)	10/12/12	10/15/12	2
• Publicar dades estació a twitter (real)	10/16/12	10/17/12	2

### 3.2 - Planificació 2a part - Fase 2

Les tasques previstes per a la Fase 2 i la seva planificació han estat:

Name	Begin date	End date	Duration
[-] • Fase 2 - Programació web	7/30/13	10/13/13	76
• Estudi requisits desenvolupament web	7/30/13	7/31/13	2
• Instal·lació entorn desenvolupament web	8/1/13	8/4/13	4
• Proves Zend Framework	8/5/13	8/6/13	2
• Proves llibreria pChart (gràfics i imatges suavitzats)	8/5/13	8/8/13	4
• Disseny gràfic aplicació web	8/9/13	8/18/13	10
[-] • Desenvolupament web (local)	8/19/13	9/22/13	35
• Programació plantilla	8/19/13	8/25/13	7
• Programació pàgina 'Resum diari'	8/26/13	8/31/13	6
• Programació pàgina 'Resum anual'	9/1/13	9/5/13	5
• Programació pàgina 'Contactar'	9/6/13	9/9/13	4
• Programació pàgina 'Legal i Crèdits'	9/10/13	9/13/13	4
• Programació bloc 'Últimes dades de l'estació'	9/14/13	9/17/13	4
• Programació bloc 'Últims tweets publicats'	9/18/13	9/22/13	5
• Proves desenvolupament web (local)	9/23/13	9/29/13	7
• Publicació web entorn de producció i proves (real)	9/30/13	10/13/13	14

La Fase 2 no només ha consistit en el desenvolupament de la pàgina web i els seus apartats, si no que també ha suposat:

- 1) La instal·lació d'un entorn de desenvolupament web local (servidor de noms, servidor de bases de dades MySQL i servidor web Apache2) per poder desenvolupar més ràpid.
- 2) La publicació de la pàgina web en un servidor de producció (entorn real) per fer accessible la pàgina web des de qualsevol ordinador connectat a Internet.

### **3.3 - Planificació 2a part - Fase 3**

Les tasques previstes per a la Fase 3 i la seva planificació han estat:

Name	Begin date	End date	Duration
• Fase 3 - Programació servei web i client android	9/30/13	11/10/13	42
• Desenvolupament web service	9/30/13	10/26/13	27
• Programació funció 'Últimes dades de l'estació'	9/30/13	10/1/13	2
• Programació funció 'Resum diari'	10/19/13	10/22/13	4
• Programació funció 'Resum anual'	10/23/13	10/26/13	4
• Desenvolupament client de proves python	9/30/13	10/26/13	27
• Programació funció 'Últimes dades de l'estació'	9/30/13	10/1/13	2
• Programació funció 'Resum diari'	10/19/13	10/22/13	4
• Programació funció 'Resum anual'	10/23/13	10/26/13	4
• Desenvolupament client android	9/30/13	10/28/13	29
• Instal·lació entorn desenvolupament android sdk	9/30/13	10/1/13	2
• Proves android sdk	10/2/13	10/7/13	6
• Disseny gràfic client android	10/8/13	10/13/13	6
• Programació 'Últimes dades de l'estació'	10/14/13	10/18/13	5
• Programació 'Resum diari'	10/19/13	10/22/13	4
• Programació 'Resum anual'	10/23/13	10/26/13	4
• Programació 'Legal i Crèdits'	10/27/13	10/28/13	2
• Proves client android en dispositiu nexus 7	10/30/13	11/10/13	12

La Fase 3 ha suposat:

- 1) El desenvolupament del servei web per a que les dades meteorològiques es puguin consultar des d'altres dispositius que no siguin navegadors web.
- 2) El desenvolupament d'un client molt lleuger i senzill realitzat en llenguatge Python, per tal d'obtenir els resultats del servei web i poder-lo desenvolupar més ràpidament (així no ha calgut esperar a tenir fet el client Android per comprovar el correcte funcionament del servei web).
- 3) El desenvolupament d'un client per a dispositius amb sistema operatiu Android, i la comprovació del correcte funcionament de l'aplicatiu en un dispositiu físic real.

### 3.4 - Planificació 2a part - Fase de documentació

Les tasques previstes i la planificació per a la Fase de documentació han estat:

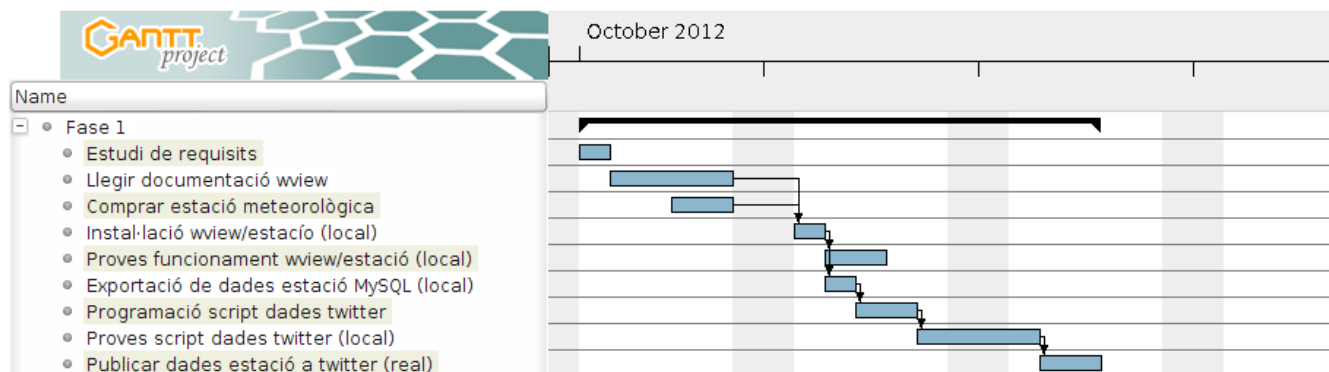
Name	Begin date	End date	Duration
Documentació / Memòria / Presentació	9/23/13	1/16/14	116
• Proposta: Descripció i Objectius del Projecte	9/23/13	9/25/13	3
• Decisió del Projecte	9/26/13	9/26/13	1
• PAC1: Planificació, tasques i objectius	9/30/13	10/11/13	12
• Lliurament PAC1	10/12/13	10/12/13	1
• PAC2: Anàlisis, disseny, casos d'ús, diagrames de classes	10/21/13	11/18/13	29
• Lliurament PAC2	11/19/13	11/19/13	1
• PAC3: Explicació, implementació, canvis	11/25/13	12/16/13	22
• Lliurament PAC3	12/17/13	12/17/13	1
• Elaboració memòria	12/18/13	1/9/14	23
• Lliurament memòria	1/10/14	1/10/14	1
• Elaboració presentació	12/25/13	1/15/14	22
• Lliurament presentació	1/16/14	1/16/14	1

Aquesta fase ha suposat la realització d'un document amb la descripció i els objectius del projecte, un document amb les planificacions de les diferents tasques del projecte (PAC1), un document amb els anàlisis, dissenys, casos d'ús i diagrames de classes del sistema implementat (PAC2), un document amb les explicacions del sistema implementat (PAC3), i per últim, la elaboració de la memòria de tot el treball i de la presentació final del projecte.

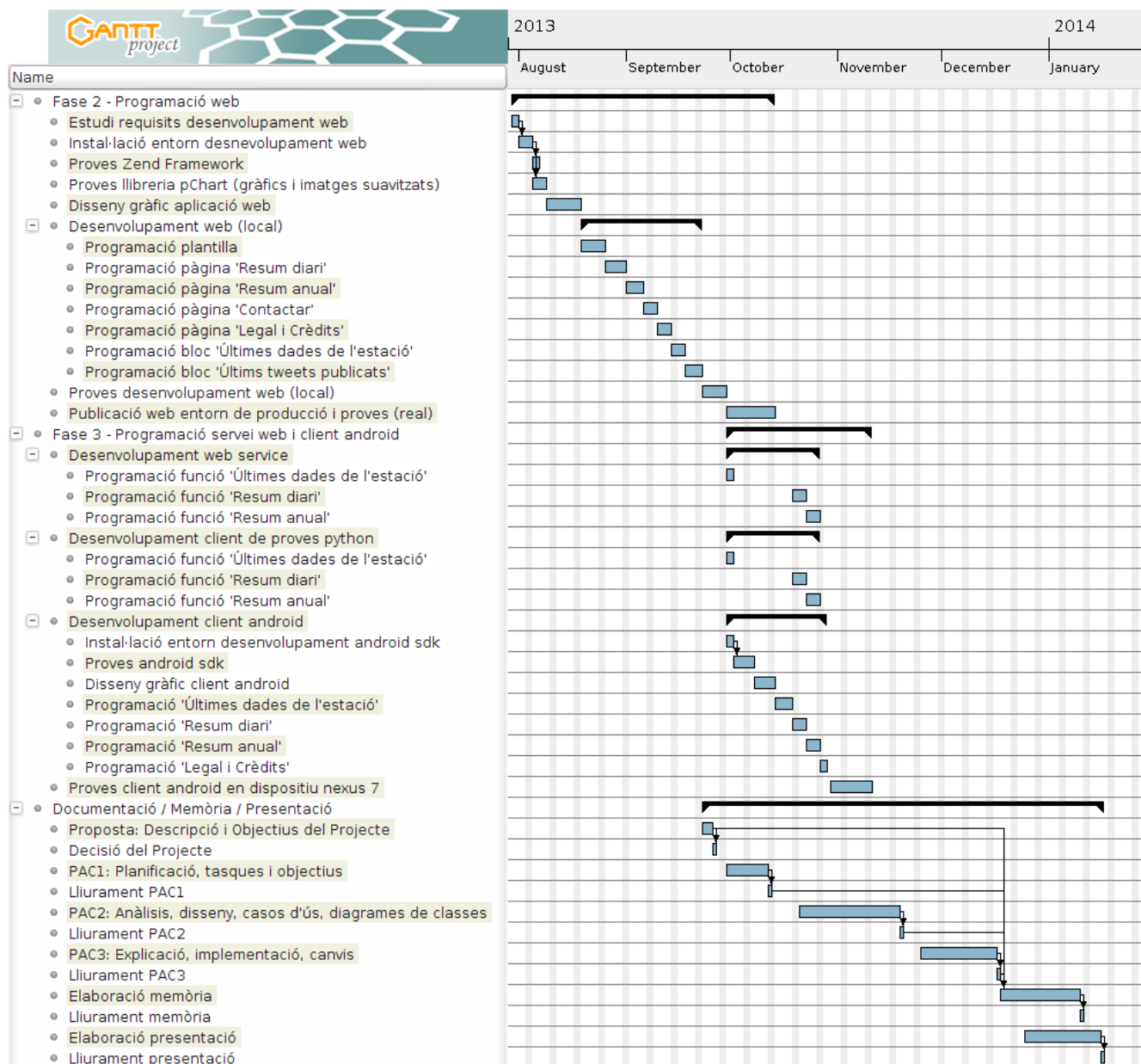
### 3.5 - Diagrames de Gantt

Per finalitzar aquesta part de la memòria, es pot veure un resum de cadascuna de les parts del projecte, amb els respectius diagrames de Gantt:

1a part - Fase 1:



**2a part – Fases 2, 3 i Documentació:**



## **4 - Anàlisi i disseny**

En aquest capítol de la memòria em centraré en el disseny de les aplicacions que s'han desenvolupat i que a la vegada permeten interactuar amb els diferents usuaris. Per tant, es tractaran els dissenys de les aplicacions web i Android, deixant així de banda els scripts que permeten publicar dades a la xarxa social Twitter, ja que el nivell de programació és molt senzill i des del punt de vista de disseny no és gaire interessant.

Començaré per identificar els actors principals de les aplicacions i els seus casos d'ús, i posteriorment detallaré les classes d'entitat i els diagrames de seqüència de les operacions.

### **4.1 - Actors de les aplicacions**

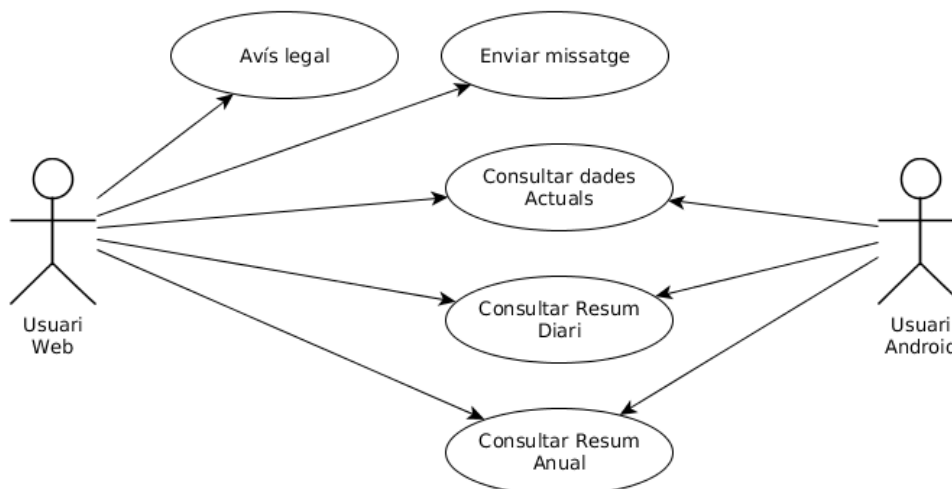
S'han distingit dos tipus d'actors diferents:

- a) Usuaris web: seran els visitants que realitzaran les consultes sobre les dades meteorològiques a través de la pàgina web.
- b) Usuaris Android: seran els visitants que realitzaran les consultes sobre les dades meteorològiques a través de l'aplicació desenvolupada pel sistema operatiu Android.

Ambdós usuaris podran realitzar les mateixes operacions de consulta, és a dir, podran consultar les dades actuals, podran consultar les dades i gràfiques diàries, i podran consultar les dades i gràfiques anuals. No obstant, els usuaris web podran realitzar, a més a més, dues operacions addicionals:

- 1) Consultar la informació sobre la titularitat de la pàgina web i els seus continguts,
- 2) Enviar missatges de correu electrònic mitjançant un formulari de contacte.

Tot això ho es pot resumir mitjançant el següent diagrama UML de casos d'ús:



## **4.2 - Casos d'ús de l'aplicació web**

<b>Cas d'ús: Avís legal (web)</b>	
<b>Descripció:</b>	Accedir a la condicions de legalitat de l'aplicació.
<b>Actors:</b>	Usuari web.
<b>Casos d'ús relacionats:</b>	Cap.
<b>Precondició:</b>	Accedir a l'apartat "Avís legal".
<b>Postcondició:</b>	Es mostra un text amb informació sobre la titularitat de la pàgina web i sobre els continguts de la mateixa.
<b>Flux d'esdeveniments</b>	
1. L'usuari web accedeix a l'apartat "Avís legal".	
2. El sistema mostra un amb informació sobre la titularitat de la pàgina web i els seus continguts.	
<b>Flux alternatiu</b>	

<b>Cas d'ús: Enviar missatge (web)</b>	
<b>Descripció:</b>	Accedir al formulari de contacte i poder enviar un correu electrònic.
<b>Actors:</b>	Usuari web.
<b>Casos d'ús relacionats:</b>	Cap.
<b>Precondició:</b>	Accedir a l'apartat "Contactar".
<b>Postcondició:</b>	Es mostra un formulari de contacte on l'usuari ha de completar uns camps amb les dades corresponents. Una vegada es completen les dades s'enviarà un correu electrònic amb el missatge a una adreça de correu electrònica determinada.
<b>Flux d'esdeveniments</b>	
1. L'usuari web accedeix a l'apartat "Contactar".	
2. El sistema mostra un formulari amb diferents camps a completar (nom, assumpte, missatge, ...).	
3. L'usuari completa els camps del formulari i fa clic al botó d'enviar.	
4. Es validen les dades que ha entrat l'usuari i s'envien per correu electrònic a una adreça determinada.	
<b>Flux alternatiu</b>	
1. Si l'usuari no ha entrat algun dels camps obligatoris, o algun dels camps conté un valor incorrecte, el sistema mostra missatge d'error indicant que es revisin les dades indicades.	

<b>Cas d'ús: Consultar dades actuals (web)</b>	
<b>Descripció:</b>	Veure les darreres dades de temperatura, humitat i pressió recollides per l'estació meteorològica.
<b>Actors:</b>	Usuari web.
<b>Casos d'ús relacionats:</b>	Cap.
<b>Precondició:</b>	Accedir a l'apartat "Dades actuals".
<b>Postcondició:</b>	S'obtenen les darreres dades recollides per l'estació.
<b>Flux d'esdeveniments</b>	
<ol style="list-style-type: none"> <li>1. L'usuari accedeix a l'apartat "Dades actuals".</li> <li>2. El sistema mostra les darreres dades recollides per l'estació.</li> </ol>	
<b>Flux alternatiu</b>	

<b>Cas d'ús: Consultar resum diari (web)</b>	
<b>Descripció:</b>	Veure els mínims i màxims diaris de temperatura, humitat i pressió recollides per l'estació. Veure la progressió diària de les dades mitjançant gràfics lineals.
<b>Actors:</b>	Usuari web.
<b>Casos d'ús relacionats:</b>	Cap.
<b>Precondició:</b>	Accedir a l'apartat "Resum diari". Seleccionar el dia pel qual es vol consultar el resum.
<b>Postcondició:</b>	S'obtenen els mínims i màxims diaris de temperatura, humitat i pressió, i els gràfics lineals amb les progressions diàries.
<b>Flux d'esdeveniments</b>	
<ol style="list-style-type: none"> <li>1. L'usuari accedeix a l'apartat "Resum diari".</li> <li>2. L'usuari selecciona el dia pel qual vol consultar el resum diari.</li> <li>3. El sistema mostra els mínims, els màxims i els gràfics lineals pel dia seleccionat.</li> </ol>	
<b>Flux alternatiu</b>	
<ol style="list-style-type: none"> <li>1. Per defecte si no se selecciona un dia, el sistema mostra el resum diari pel dia actual.</li> <li>2. Si l'usuari selecciona una dia pel qual no hi ha dades emmagatzemades, el sistema mostra un missatge d'avís indicant que no hi ha dades emmagatzemades per aquell dia en concret.</li> </ol>	



<b>Cas d'ús: Consultar resum anual (web)</b>	
<b>Descripció:</b>	Veure les mitjanes mínimes i màximes mensuals de temperatura, humitat i pressió recollides per l'estació. Veure la progressió anual de les dades mitjançant gràfics lineals.
<b>Actors:</b>	Usuari web.
<b>Casos d'ús relacionats:</b>	Cap.
<b>Precondició:</b>	Accedir a l'apartat "Resum anual". Seleccionar l'any pel qual es vol consultar el resum.
<b>Postcondició:</b>	S'obtenen les mitjanes mínimes i màximes mensuals de temperatura, humitat i pressió, i els gràfics lineals amb les progressions anuals.
<b>Flux d'esdeveniments</b>	
<ol style="list-style-type: none"><li>1. L'usuari accedeix a l'apartat "Resum anual".</li><li>2. L'usuari selecciona l'any pel que vol consultar el resum anual.</li><li>3. El sistema mostra les mitjanes mínimes i màximes mensuals, així com els gràfics lineals de l'any seleccionat.</li></ol>	
<b>Flux alternatiu</b>	
<ol style="list-style-type: none"><li>1. Per defecte si no se selecciona un any, el sistema mostra el resum anual per l'any actual.</li><li>2. Si l'usuari selecciona un any pel qual no hi ha dades emmagatzemades, el sistema mostra un missatge d'avís indicant que no hi ha dades emmagatzemades per aquell any en concret.</li></ol>	

### **4.3 - Casos d'ús de l'aplicació Android**

Tot i que, tant l'aplicació web com l'aplicació Android tenen les mateixes funcionalitats bàsiques (consultar les dades actuals, i consultar el resum diari i el resum anual), aquestes no obtenen les dades de la mateixa manera (l'aplicació Android ho fa mitjançant un servei web). A més, en l'aplicació Android és interessant tenir algun tipus mecanisme per registrar els accessos dels usuaris que accedeixin al sistema des de terminals Android, i obtenir així dades estadístiques sobre l'ús del sistema desenvolupat.

Això amb l'aplicació web no és necessari desenvolupar ja que, per exemple, es pot fer ús de l'eina AWStats (que analitza els fitxers de log del servidor web Apache2 per obtenir dades estadístiques avançades) o de l'eina Google Analytics (que permet obtenir dades estadístiques molt més avançades). En canvi amb l'aplicació Android aquesta necessitat no la tenim coberta de manera senzilla, per tant, és necessari introduir-la en algun punt de l'aplicació. Òbviament no es pot fer aquesta mena de registre de dades als terminals dels usuaris, pel que el més pertinent és fer-ho durant les crides realitzades al servei web, el qual queda explicat més endavant.

És per aquest motiu que, tot i les semblances entre les funcionalitats d'una aplicació i l'altra, els casos d'ús, diagrames de seqüència, etc. seran lleugerament diferents entre una aplicació web i l'aplicació Android.

<b>Cas d'ús: Consultar resum diari (Android)</b>	
<b>Descripció:</b>	Veure els mínims i màxims diaris de temperatura, humitat i pressió recollides per l'estació. Veure la progressió diària de les dades mitjançant gràfics lineals.
<b>Actors:</b>	Usuari Android.
<b>Casos d'ús relacionats:</b>	Cap.
<b>Precondició:</b>	Accedir a l'apartat "Resum diari". Seleccionar el dia pel qual es vol consultar el resum.
<b>Postcondició:</b>	S'obtenen els mínims i màxims diaris de temperatura, humitat i pressió, i els gràfics lineals amb les progressions diàries.
<b>Flux d'esdeveniments</b>	
<ol style="list-style-type: none"><li>1. L'usuari accedeix a l'apartat "Resum diari".</li><li>2. L'usuari selecciona el dia pel qual vol consultar el resum diari.</li><li>3. L'aplicació fa una crida al servei web demanant el resum diari, passant com a paràmetre el dia seleccionat.</li><li>4. El servei web registra l'operació (per obtenir dades d'ús i fer un tractament estadístic posterior) i obté les dades del resum diari pel dia rebut.</li><li>5. El servei web envia les dades obtingudes a l'aplicació.</li><li>6. L'aplicació Android mostra els mínims, els màxims i els gràfics lineals pel dia seleccionat.</li></ol>	
<b>Flux alternatiu</b>	
<ol style="list-style-type: none"><li>1. Si no se selecciona un dia, l'aplicació demana al servei web les dades de resum diari pel dia actual.</li><li>2. Si l'usuari selecciona una dia pel qual no hi ha dades emmagatzemades, el sistema mostra un missatge d'avís indicant que no hi ha dades emmagatzemades per aquell dia en concret.</li></ol>	

### **Observació:**

El servei web és l'encarregat de recuperar les dades i d'enviar-les a l'aplicació.

Si la crida és "Resum diari" o "Resum anual", el servei web també és l'encarregat de generar els gràfics lineals amb les progressions de les dades. No obstant, els gràfics lineals no s'envien de retorn a l'aplicació juntament amb la resta de les dades, sinó que s'emmagatzemen en una carpeta coneguda del servidor on està allotjada la pàgina web.

Quan l'aplicació Android rep la resposta del servei web amb les dades resultants, podrà mostrar els gràfics fent una crida directament a la URL de les imatges dels gràfics corresponents.

<b>Cas d'ús: Consultar resum anual (Android)</b>	
<b>Descripció:</b>	Veure les mitjanes mínimes i màximes mensuals de temperatura, humitat i pressió recollides per l'estació. Veure la progressió anual de les dades mitjançant gràfics lineals.
<b>Actors:</b>	Usuari Android.
<b>Casos d'ús relacionats:</b>	Cap.
<b>Precondició:</b>	Accedir a l'apartat "Resum anual". Seleccionar l'any pel qual es vol consultar el resum.
<b>Postcondició:</b>	S'obtenen les mitjanes mínimes i màximes mensuals de temperatura, humitat i pressió, i els gràfics lineals amb les progressions anuals.
<b>Flux d'esdeveniments</b>	
<ol style="list-style-type: none"><li>1. L'usuari accedeix a l'apartat "Resum anual".</li><li>2. L'usuari selecciona l'any pel qual vol consultar el resum anual.</li><li>3. L'aplicació fa una crida al servei web demanant el resum diari, passant com a paràmetre l'any seleccionat.</li><li>4. El servei web registra l'operació (per obtenir dades d'ús i fer un tractament estadístic posterior) i obté les dades del resum diari per l'any rebut.</li><li>5. El servei web envia les dades obtingudes a l'aplicació.</li><li>6. L'aplicació mostra les mitjanes mínimes i màximes mensuals, així com els gràfics lineals per l'any seleccionat.</li></ol>	
<b>Flux alternatiu</b>	
<ol style="list-style-type: none"><li>1. Si no se selecciona un any, l'aplicació demana al servei web les dades de resum anual per l'any actual.</li><li>2. Si l'usuari selecciona un any pel qual no hi ha dades emmagatzemades, el sistema mostra un missatge d'avís indicant que no hi ha dades emmagatzemades per aquell any en concret.</li></ol>	

### **Observació:**

Com s'acaba d'exposar, quan es fa la crida d'aquesta operació, el servei web s'encarrega de generar els gràfics lineals amb les progressions de les dades. Una vegada creats i quan l'aplicació rep la resposta del servei web amb les dades obtingudes (només les dades, els gràfics no s'envien juntament amb la resposta del servei web), aquesta podrà mostrar els gràfics mitjançant una crida directa a la URL de les imatges dels gràfics.

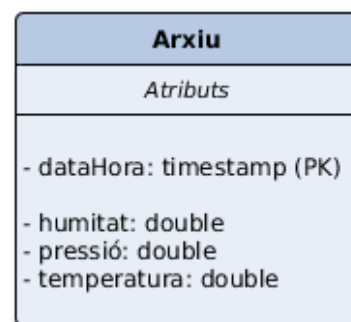
<b>Cas d'ús: Consultar dades actuals (Android)</b>	
<b>Descripció:</b>	Veure les darreres dades de temperatura, humitat i pressió recollides per l'estació meteorològica.
<b>Actors:</b>	Usuari Android.
<b>Casos d'ús relacionats:</b>	Cap.
<b>Precondició:</b>	Accedir a l'apartat "Dades actuals".
<b>Postcondició:</b>	S'obtenen les darreres dades recollides per l'estació.
<b>Flux d'esdeveniments</b>	
1. L'usuari accedeix a l'apartat "Dades actuals". 2. L'aplicació fa una crida al servei web demanant les dades actuals. 3. El servei web registra l'operació (per obtenir dades d'ús i fer un tractament estadístic posterior) i recupera les dades actuals. 4. El servei web envia les dades recuperades a l'aplicació. 5. El sistema mostra les dades actuals lliurades pel servei web.	
<b>Flux alternatiu</b>	

#### **4.4 - Classes d'entitat**

Si es realitza un anàlisi dels casos d'ús anteriors, es pot concloure que principalment es necessiten dues classes d'entitat diferents per a les aplicacions: una classe arxiu que serà on es guardaran les dades meteorològiques, i una altra classe accessos que serà on es guardaran els diferents accessos fets des de dispositius amb sistema Android.

##### 1) Classe Arxiu

- És utilitzada tant pel sistema de l'aplicació web, com pel sistema de l'aplicació Android.
- Serveix per emmagatzemar les dades recollides per l'estació meteorològica, és a dir: ha de permetre enregistrar, per cada lectura de l'estació, la marca de temps o *timestamp*, la temperatura, la humitat i la pressió.
- S'utilitza la marca de temps com a clau primària ja que com les lectures es realitzen en intervals de temps diferents, aquestes tenen una marca de temps única.



- Pel que fa als mètodes de la classe arxiu, he cregut necessari definir els mètodes següents:

- `getUltimesDades()`  
// per recuperar les últimes dades meteorològiques registrades per l'estació
- `getArxiuSegonsMarcaDeTemps(marcaDeTemps)`  
// per recuperar les dades meteorològiques a partir d'una marca de temps
- `getPressioMinSegonsInterval(interval1, interval2)`  
// per recuperar la pressió mínima d'un interval de temps
- `getPressioMaxSegonsInterval(interval1, interval2)`  
// per recuperar la pressió màxima d'un interval de temps
- `getTemperaturaMinSegonsInterval(interval1, interval2)`  
// per recuperar la temperatura mínima d'un interval de temps
- `getTemperaturaMaxSegonsInterval(interval1, interval2)`  
// per recuperar la temperatura màxima d'un interval de temps
- `getHumitatMinSegonsInterval(interval1, interval2)`  
// per recuperar la humitat mínima d'un interval de temps
- `getTemperaturaMaxSegonsInterval(interval1, interval2)`  
// per recuperar la humitat màxima d'un interval de temps
- `getMitjanaSegonsInterval(interval1, interval2)`  
// per recuperar la pressió, temperatura i humitat mitjana d'un interval de temps

## 2) Classe Accessos

- És utilitzada només pel sistema de l'aplicació Android mitjançant les crides fetes al servei web.

- Serveix per emmagatzemar totes les dades relacionades amb els accessos al sistema realitzats des de sistemes Android, és a dir: ha de permetre registrar, per cada accés, la seva marca de temps o *timestamp*, la operació que es demana al servei web, els paràmetres de la operació i la IP des de la qual es crida al servei web.

- Com a clau primària s'utilitza un identificador d'accés ja que crec que, per una mateixa marca de temps es pot arribar a fer més d'un accés, per tant, en aquest cas no es pot utilitzar la marca de temps com a identificador únic.

- Pel que fa als mètodes de la classe accessos, he cregut que només és necessari definir un mètode que permeti registrar els accessos a la base de dades (les consultes dels accessos es poden fer directament mitjançant una eina de gestió, com per exemple, phpMyAdmin):

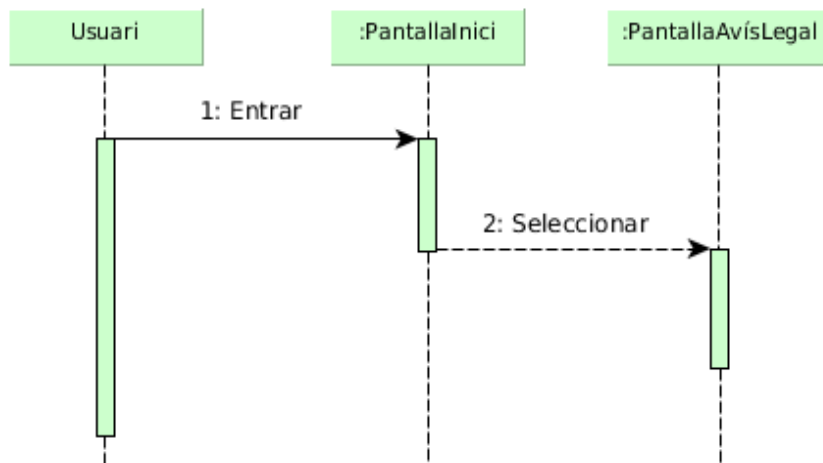
- `afegirAccess()`  
// permet al servei web registrar dades de les diferents crides fetes des de dispositius amb sistema operatiu Android

Accessos
Atributs
- idAccess: integer (PK)
- dataHora: timestamp
- hostName: string
- operació: string
- paràmetres: string
- idioma: string

## 4.5 - Diagrames de seqüència de l'aplicació web

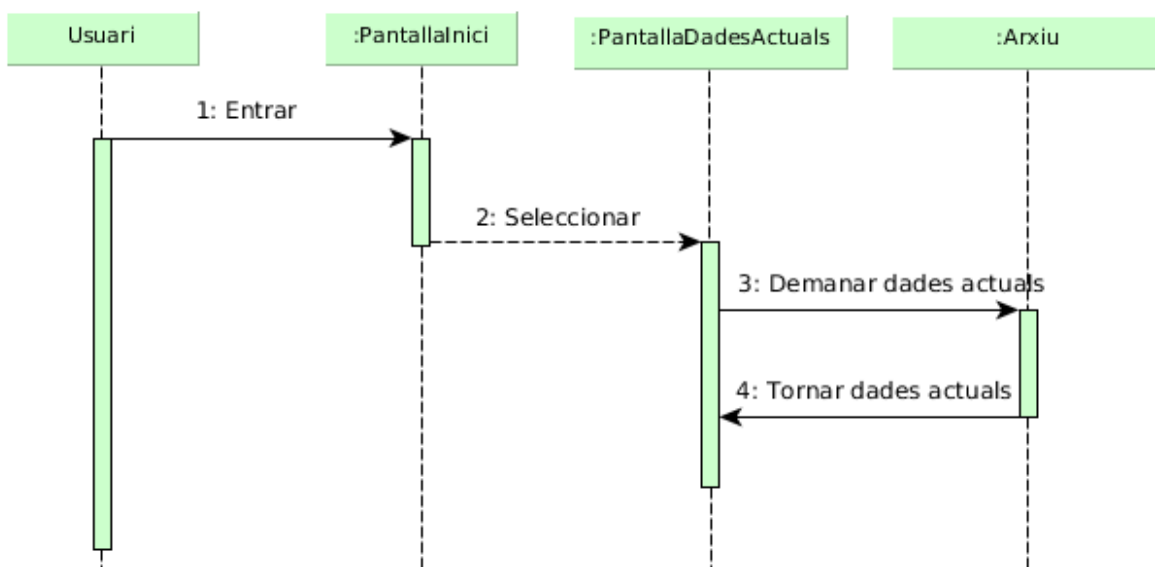
Tot seguit es mostren els diagrames de seqüència de les operacions de l'aplicació web, detallades en els apartats anteriors:

### Diagrama de seqüència "Avís legal"



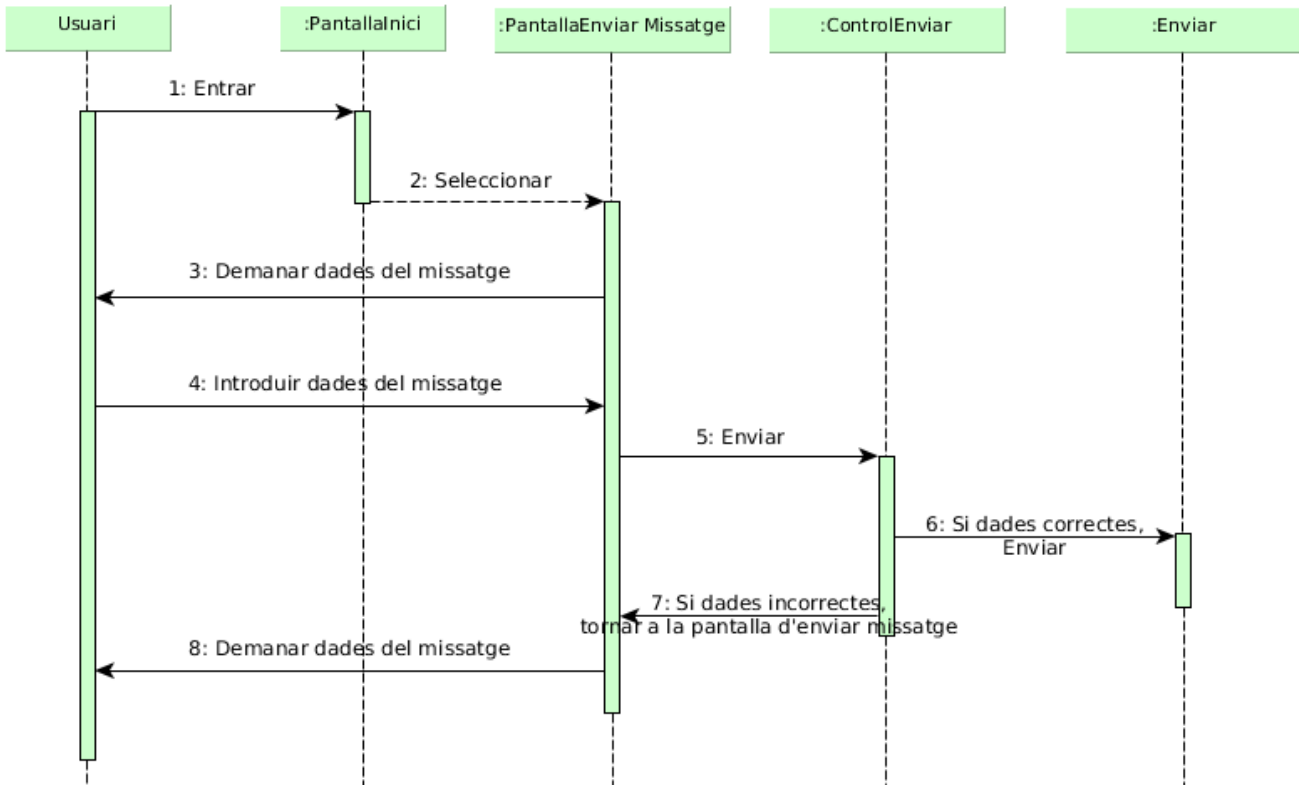
- La pantalla "Avís legal" mostra un text amb informació sobre la titularitat de la pàgina web i sobre els continguts de la mateixa.

### Diagrama de seqüència "Dades actuals"



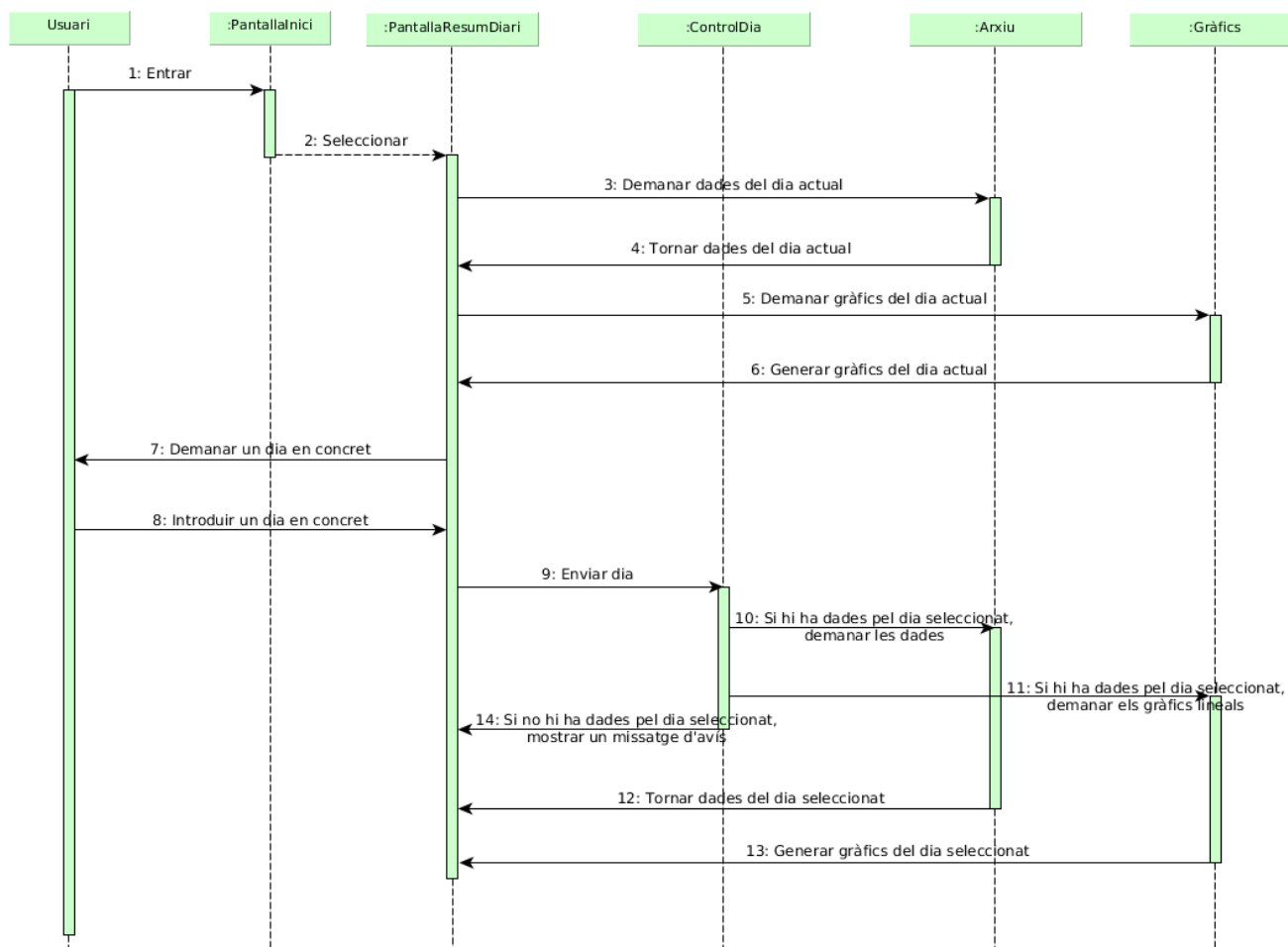
- Des de la pantalla “Dades actuals” es crida a la classe “Arxiu” demanant les darreres dades emmagatzemades per l'estació,
- La classe “Arxiu” torna les darreres dades de temperatura, humitat i pressió a la pantalla “Dades Actuals”.

### Diagrama de seqüència “Enviar missatge”



- La pantalla “Enviar missatge” conté un formulari amb els camps que s'hauran d'enviar per correu electrònic (nom i cognoms de l'emissor, assumpte del missatge, missatge, ...),
- L'usuari haurà d'emplenar els camps del formulari i fer clic al botó d'enviar,
- Al moment de fer enviar es fa un control sobre les dades emplenades per l'usuari,
- Si les dades són correctes, s'envien les dades del formulari per correu electrònic a una adreça determinada,
- Si les dades no són correctes, es torna a la pantalla “Enviar missatge” i es mostra un missatge d'avís per tal d'alertar a l'usuari que algun dels camps és incorrecte.

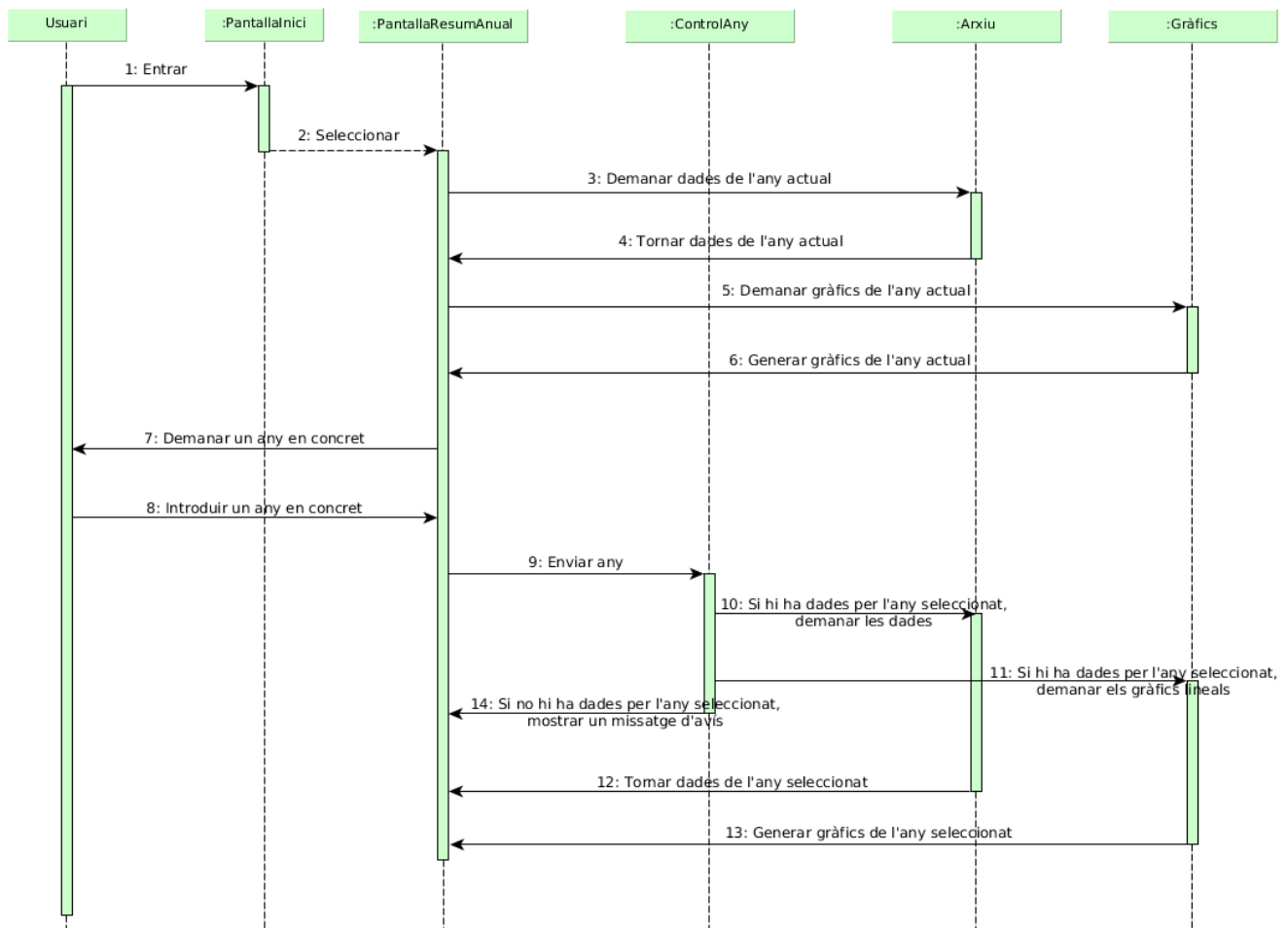
## Diagrama de seqüència "Resum diari"



- La pantalla "Resum diari" per defecte demana les dades i gràfiques del dia actual i les mostra,
- Opcionalment l'usuari pot seleccionar una data concreta mitjançant un camp de tipus calendari i consultar les dades del dia seleccionat,
- Al moment d'enviar el dia seleccionat, es fa un control sobre el dia enviat per determinar si existeixen dades emmagatzemades dins l' "Arxiu",
- Si hi ha dades emmagatzemades, la classe "Arxiu" torna les dades i es generen les gràfiques pel dia corresponent,
- Si resulta que no hi ha dades emmagatzemades, es torna a la pantalla "Resum diari" i es mostra un missatge d'avís indicant que no hi ha dades emmagatzemades pel dia seleccionat.



## Diagrama de seqüència “Resum anual”

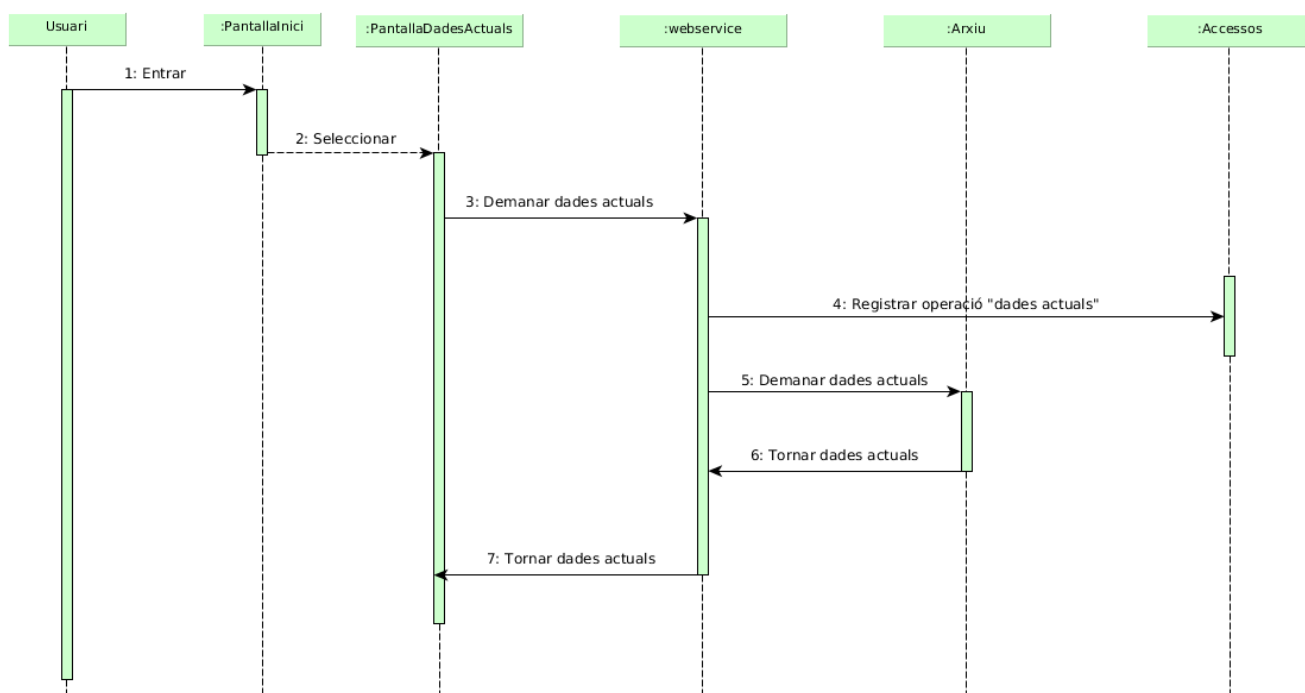


- La pantalla “Resum anual” per defecte demana les dades i gràfiques de l'any actual i les mostra,
- Opcionalment l'usuari pot seleccionar un any en concret mitjançant un camp de tipus desplegable i consultar les dades per l'any seleccionat,
- Al moment d'enviar l'any seleccionat, es fa un control sobre l'any enviat per determinar si existeixen dades emmagatzemades dins l' “Arxiu”,
- Si hi ha dades emmagatzemades, la classe “Arxiu” torna les dades i es generen les gràfiques per l'any corresponent,
- Si resulta que no hi ha dades emmagatzemades, es torna a la pantalla “Resum anual” i es mostra un missatge d'avís indicant que no hi ha dades emmagatzemades per l'any seleccionat.

## 4.5 - Diagrames de seqüència de l'aplicació Android

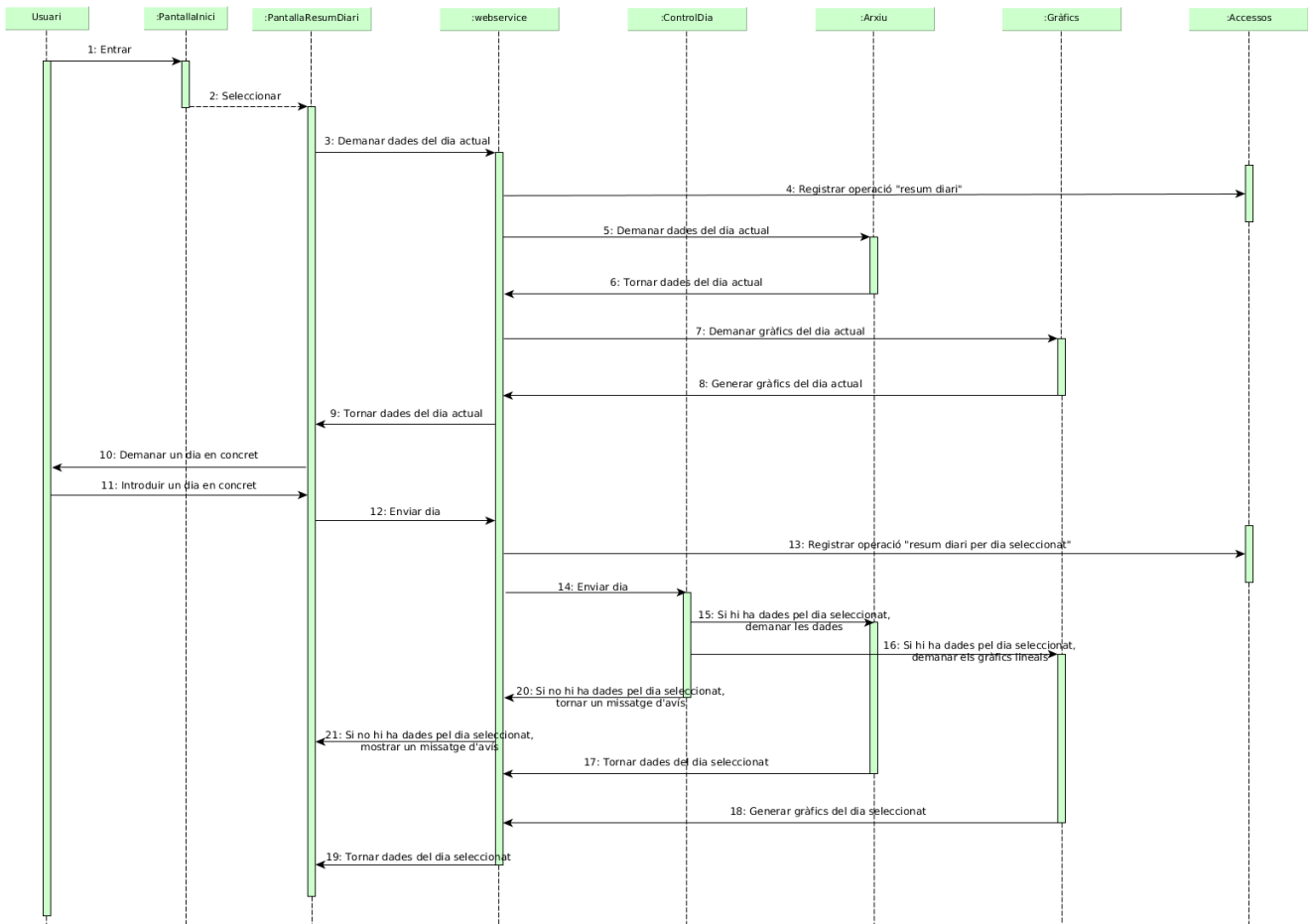
En aquest punt es mostren en detall els diagrames de seqüència de les operacions de l'aplicació Android, vistes en els apartats anteriors:

### Diagrama de seqüència "Dades actuals"



- La pantalla "Dades actuals" fa una crida al servei web demanant les darreres dades emmagatzemades per l'estació,
- El primer que fa el servei web és registrar la crida a la operació "dades actuals" mitjançant la classe "Accessos",
- El servei web demana a la classe "Arxiu" les darreres dades emmagatzemades,
- La classe "Arxiu" torna les darreres dades de temperatura, humitat i pressió al servei web,
- El servei web torna a la pantalla "Dades actuals" les darreres dades de temperatura, humitat i pressió.

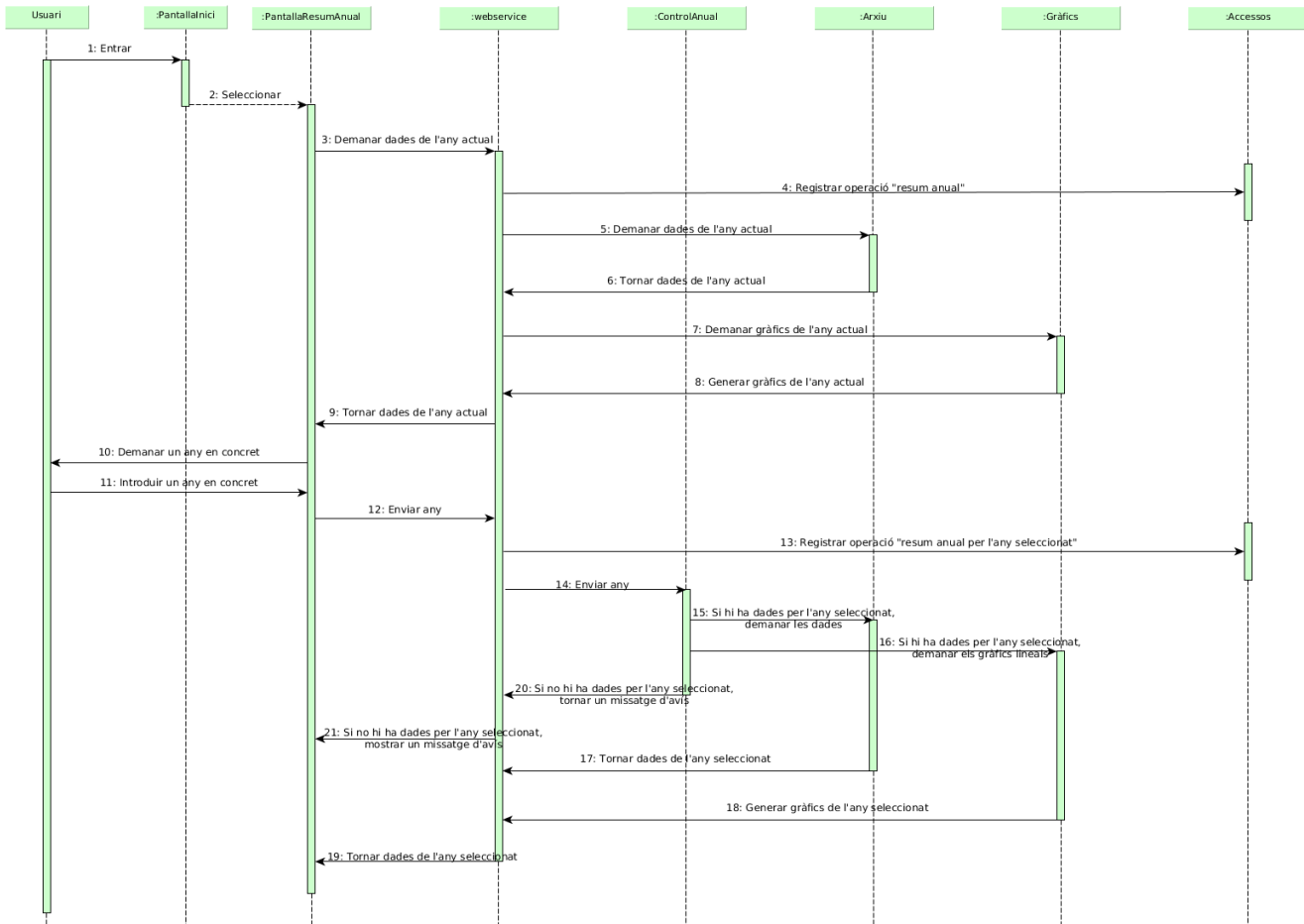
## Diagrama de seqüència "Resum diari"



- La pantalla "Resum diari" per defecte demana les dades i gràfiques del dia actual al servei web,
- Cada operació "Resum diari" que rebí el servei web serà registrada mitjançant la classe "Accessos",
- El servei web demana a la classe "Arxiu" les dades emmagatzemades i es generen les gràfiques pel dia actual,
- La classe "Arxiu" torna les darreres dades de temperatura, humitat i pressió al servei web,
- El servei web torna a la pantalla "Resum diari" les darreres dades de temperatura, humitat i pressió pel dia actual,
- Opcionalment l'usuari pot seleccionar una data concreta mitjançant un camp de tipus calendari i consultar les dades pel dia seleccionat,

- El servei web rep el dia seleccionat i fa un control per esbrinar si existeixen dades emmagatzemades dins l' "Arxiu",
- Si hi ha dades emmagatzemades, el servei web demana a la classe "Arxiu" les dades emmagatzemades i es generen les gràfiques pel dia seleccionat,
- La classe "Arxiu" torna les darreres dades de temperatura, humitat i pressió al servei web,
- El servei web torna a la pantalla "Resum diari" les dades de temperatura, humitat i pressió pel dia seleccionat,
- Si resulta que no hi ha dades emmagatzemades, es torna a la pantalla "Resum diari" i es mostra un missatge d'avís indicant que no hi ha dades emmagatzemades pel dia seleccionat.

### Diagrama de seqüència "Resum anual"



- La pantalla “Resum anual” per defecte demana les dades i gràfiques de l'any actual al servei web,
- Cada operació “Resum anual” que rebí el servei web serà registrada mitjançant la classe “Accessos”.
- El servei web demana a la classe “Arxiu” les dades emmagatzemades i es generen les gràfiques per l'any actual,
- La classe “Arxiu” torna les darreres dades de temperatura, humitat i pressió al servei web,
- El servei web torna a la pantalla “Resum anual” les darreres dades de temperatura, humitat i pressió per l'any actual,
- Opcionalment l'usuari pot seleccionar una any concret mitjançant un camp de tipus desplegable i consultar les dades per l'any seleccionat,
- El servei web rep l'any seleccionat i fa un control per esbrinar si existeixen dades emmagatzemades dins l' “Arxiu”,
- Si hi ha dades emmagatzemades, el servei web demana a la classe “Arxiu” les dades emmagatzemades i es generen les gràfiques per l'any seleccionat,
- La classe “Arxiu” torna les darreres dades de temperatura, humitat i pressió al servei web,
- El servei web torna a la pantalla “Resum diari” les dades de temperatura, humitat i pressió per l'any seleccionat,
- Si resulta que no hi ha dades emmagatzemades, es torna a la pantalla “Resum anual” i es mostra un missatge d'avís indicant que no hi ha dades emmagatzemades per l'any seleccionat.

## 5 - Implementació

En aquest capítol de la memòria s'exposa com he implementat cadascuna de les aplicacions desenvolupades, explico en quins llenguatges s'han programat i descriu les llibreries que he utilitzat per completar tots els requisits del projecte.

També es manifesta altres aspectes que considero rellevants pel que fa a les implementacions portades a terme durant aquest treball de fi de carrera.

### 5.1 - Publicació les dades meteorològiques a la xarxa social Twitter

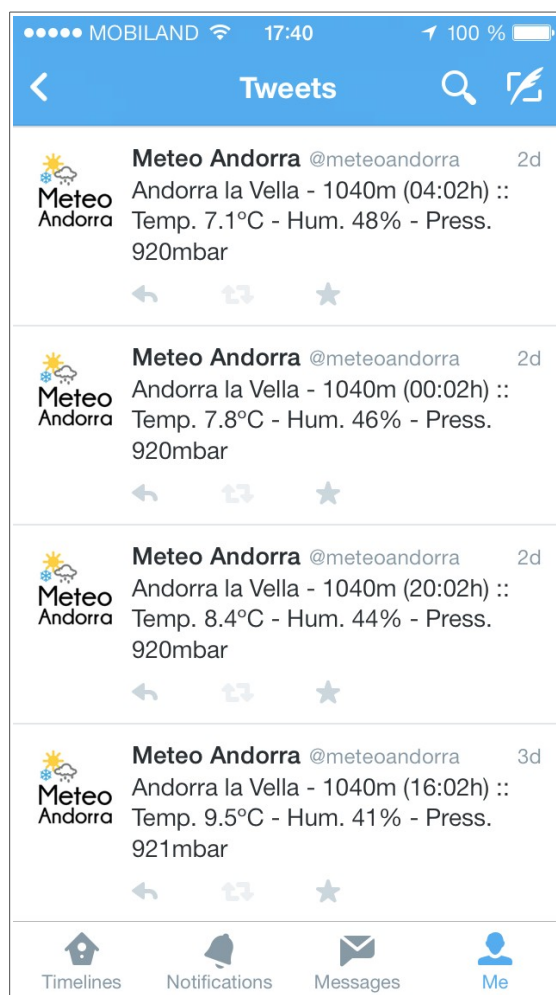
D'entre les diferents aplicacions desenvolupades durant el projecte, l'encarregada de publicar les dades meteorològiques a la xarxa social Twitter ha estat la més senzilla d'implementar.

L'aplicació consisteix en un petit script programat en llenguatge Python que es connecta a la base de dades del servidor local on es troba l'estació meteorològica, i recupera la pressió, humitat i temperatura de la darrera lectura enregistrada per finalment, publicar un tweet amb les dades recuperades a la xarxa social.

El llenguatge Python és un llenguatge lliure, senzill i molt potent al mateix temps. Té el gran avantatge que disposa de moltes biblioteques ja desenvolupades que permeten reduir molt temps i esforços a l'hora de trobar una solució per a un problema concret. A més, el codi resultant sempre acaba sent molt més net i simple si es compara amb d'altres llenguatges de programació (com podria ser, per exemple, el llenguatge Java), fet que permet mantenir de manera menys complicada els programes.

És per aquests motius exposats que he considerat el llenguatge Python com el candidat idoni per implementar aquesta part del projecte.

D'altra banda, he utilitzat la llibreria [twitter 1.9.4](#) del mateix llenguatge, la qual consisteix en una API minimalista i un conjunt d'eines que permeten interactuar amb la xarxa Twitter, ja sigui des de la línia de comandes o des de programes d'aquest mateix llenguatge. En aquest cas en concret m'ha servit per simplificar el procés d'autenticació amb la xarxa social (el qual es fa amb el protocol d'autenticació OAuth), i per publicar els tweets.



Finalment, s'ha programat l'script dins l'administrador de processos *cron* del servidor local per tal que aquest s'executi de forma periòdica cada quatre hores.

He considerat que executar-lo cada quatre hores és el més idoni. De fer-ho més sovint podria saturar per excés de tweets als seguidors de *@meteoandorra*, i pel contrari, crec que fer-ho amb menys freqüència no permetria seguir l'evolució diària de les dades meteorològiques.

## **5.2 - Desenvolupament de l'aplicació web**

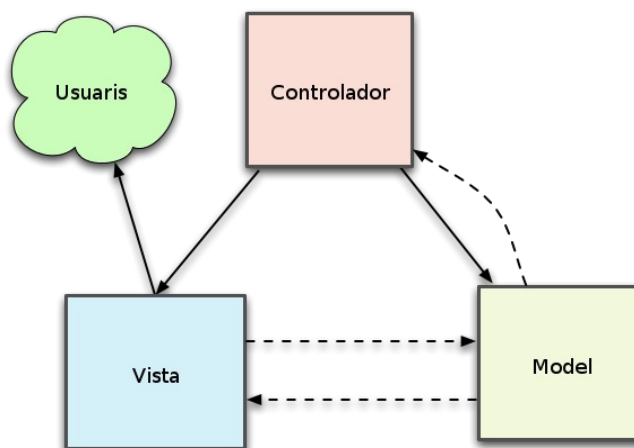
S'ha decidit programar l'aplicació web utilitzant el llenguatge PHP per ser un llenguatge molt madur i estable a l'hora de programar pàgines web dinàmiques. A l'igual que el llenguatge Python també és un llenguatge amb una llicència de tipus lliure, que destaca sobretot per la seva versatilitat, velocitat, seguretat, simplicitat i per ser multi-plataforma.

Tanmateix, s'ha de dir que per tal de facilitar la feina del desenvolupament de l'aplicació web s'ha utilitzat l'eina de treball per a PHP Zend Framework, la qual, una vegada superada la corba d'aprenentatge, esdevé un eina de gran utilitat.

Aquest *framework* de PHP està completament orientat a objectes, el qual permet als programadors ampliar tots els seus components sense haver de modificar el codi base. El seu disseny consisteix en agrupacions de classes d'una manera molt simple, convertint-lo en una eina fortament modular. D'aquesta manera, permet construir projectes utilitzant els components de forma aïllada reduint així el desacoblament general del projecte.

Segurament un dels seus punts més forts és que per sí mateix el seu esquema de treball utilitza el patró de disseny *Model View Controller* (d'ara endavant, MVC), permetent estructurar el codi del projecte d'una manera ordenada i neta, i separant els diferents nivells de responsabilitats. D'aquesta forma s'aconsegueix un desacoblament molt alt entre les diferents capes, s'augmenta la reutilització i l'adaptabilitat, i se'n facilita el manteniment.

El patró de disseny MVC estructura les aplicacions en tres capes ben diferenciades: la capa model, la capa vista i la capa controlador.



### 1) La capa model

Inclou la implementació de les classes que permeten interactuar i manipular les dades que s'emmagatzemen en les bases de dades.

### 2) La capa vista

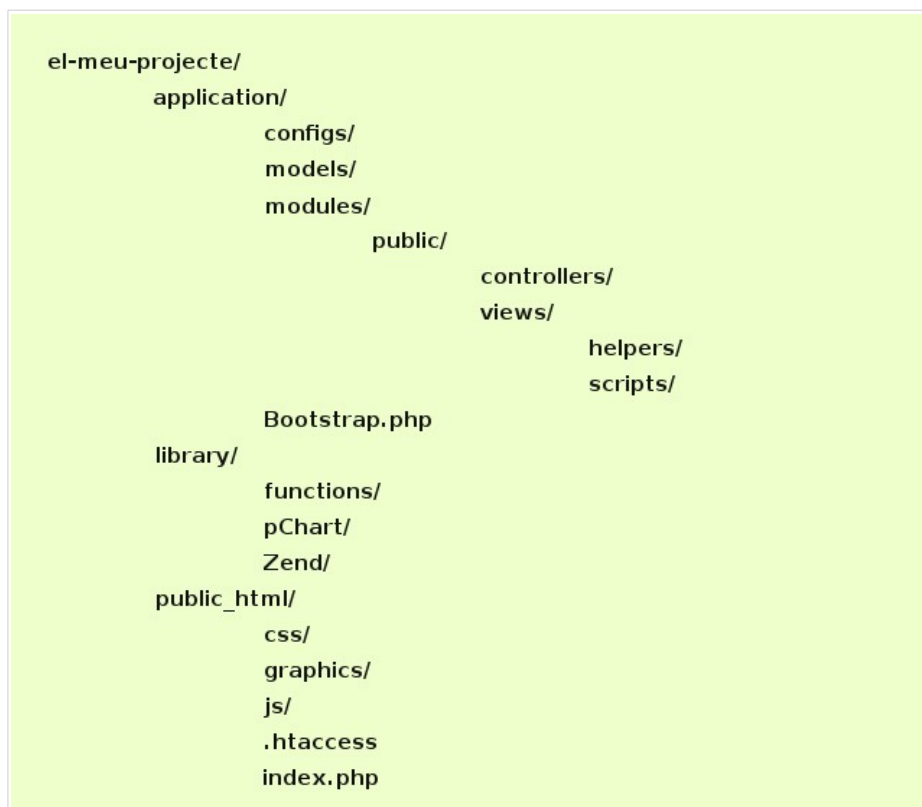
Inclou els dissenys d'interfícies que permeten presentar les dades als usuaris, i els components que permeten també interactuar l'usuari amb l'aplicació.

### 3) La capa controlador

És la capa responsable de la gestió dels esdeveniments de l'aplicació i de coordinar la interacció entre les capes model i vista.

A banda d'implementar el patró de disseny MVC, el Zend Framework també està dissenyat seguint un patró de disseny *Front Controller* (d'ara endavant, FC), que és l'encarregat de rebre i controlar totes les sol·licituds que li arriben des d'una URL, i que en funció dels paràmetres obtinguts cridarà a un controlador o a un altre. El que significa que ha d'existir un únic script PHP per atendre totes les peticions, tal i com s'explica a continuació.

Tot plegat fa que un projecte desenvolupat amb l'eina de treball Zend Framework acabi tenint una estructura més o menys predeterminada, que en el present projecte ha estat la següent:





- application: és el directori que conté l'aplicació i on s'emmagatzemen el sistema MVC, les configuracions i l'arxiu Bootstrap.php,
- configs: és el directori on s'emmagatzemen els fitxers de configuració de l'aplicació,
- models: és el directori que conté les classes de la capa model,
- modules: és un directori que permet agrupar un conjunt de controladors per aconseguir tenir el codi de l'aplicació més organitzat,
- controllers: és el directori que conté les classes de la capa controlador,
- views: és el directori que conté les classes de la capa vista,
- helpers: aquest directori conté ajudants de vista (petits scripts que permeten reduir i simplificar codi repetitiu),
- scripts: aquest directori conté per cada controlador, la seva respectiva vista (plantilles de pàgines .html),
- Bootstrap.php: la finalitat d'aquest arxiu és la d'arrancar l'aplicació i d'inicialitzar els components per fer que aquests estiguin disponibles per a l'aplicació,
- library: és on es guarden les llibreries comunes de l'aplicació, incloses la del Zend Framework en sí,
- public\_html: aquest directori conté tots els arxius públics de l'aplicació,
- .htaccess: la finalitat d'aquest arxiu és la de modificar algunes directrius del servidor web per tal de redirigir totes les peticions que vinguin des d'una URL a un únic arxiu index.php,
- index.php: la finalitat d'aquest arxiu és la d'atendre totes les peticions de l'aplicació (gràcies a les directrius declarades al fitxer .htaccess), aconseguint d'aquesta manera cridar al patró de disseny FC i executar l'aplicació a través d'una crida al fitxer Bootstrap.php.

Aquest factor de combinar els models MVC i FC permet un desenvolupament molt àgil i ordenat a l'hora de programar pàgines webs dinàmiques, fent d'aquest *framework* una eina molt potent i versàtil.

Respecte al disseny inicial que s'ha explicat en el capítol 4 d'aquest document, cal comentar que en el moment d'implementar-lo s'ha decidit variar-lo una mica per tal de poder millorar l'aspecte gràfic resultant i fer més atractiva l'aplicació web.

D'haver seguit fil per randa el plantejament inicial del capítol 4, hauria hagut d'implementar un total de 5 controladors i 5 vistes per a cadascun dels casos d'ús, és a dir, s'hauria hagut d'implementar un controlador i una vista pels casos: "Consultar dades actuals", "Consultar resum diari", "Consultar resum anual", "Avis legal" i "Enviar missatge". Al final no he cregut necessari realitzar un controlador i una vista específic pel cas d'ús "Consultar dades actuals", ja que he considerat que seria més pràctic incloure la informació de les dades actuals en cadascuna de les vistes. És per aquest motiu que al final s'ha decidit implementar el cas d'ús "Consultar dades actuals" mitjançant un *helper*.

En la següent captura de pantalla es pot veure com la vista "Resum diari" inclou el *helper* "Dades actuals" en la columna lateral de la dreta.

The screenshot shows the 'Meteo Andorra' website interface. At the top, there are navigation links for 'Català', 'English', 'Español', and 'Français'. The main header includes the logo and the text '{ Dades meteorològiques del Principat d'Andorra }'. Below this, there are tabs for 'Resum diari' (selected) and 'Resum anual'. The main content area is divided into two columns. The left column contains a summary of the station (Andorra la Vella, 1040m) and sensors (Temperature, Humidity, Pressure). It features a 'Temperatura' line graph showing data for 'Ahir' (yesterday) and 'Avui' (today). Below the graph, it lists the minimum and maximum temperatures for both days. A 'Humitat' line graph is also present. The right column features a 'Dades actuals Andorra la Vella' section with the latest update time and current temperature, humidity, and pressure. Below this is a 'Previsió del temps' section showing forecasts for 'Avui', 'Demà', and 'Dilluns'. At the bottom right, there is a 'Tweets més recents @meteoandorra' section displaying recent tweets with weather data for various times of the day.

**Meteo Andorra**  
{ Dades meteorològiques del Principat d'Andorra }

Resum diari | Resum anual

Estació: Andorra la Vella  
Alçada: 1040m  
Sensors: Temperatura, Humitat, Pressió

Selecció d'una data:

**Temperatura**

Ahí: 9.7 °C (Màxima a les 15:45h)  
Avui: 8.5 °C (Màxima a les 15:45h)

**Humitat**

**Dades actuals Andorra la Vella**  
Última actualització: 21-12-2013 a les 17:55h

- Temperatura: 7.5 °C
- Humitat: 32 %
- Pressió: 925 mbar

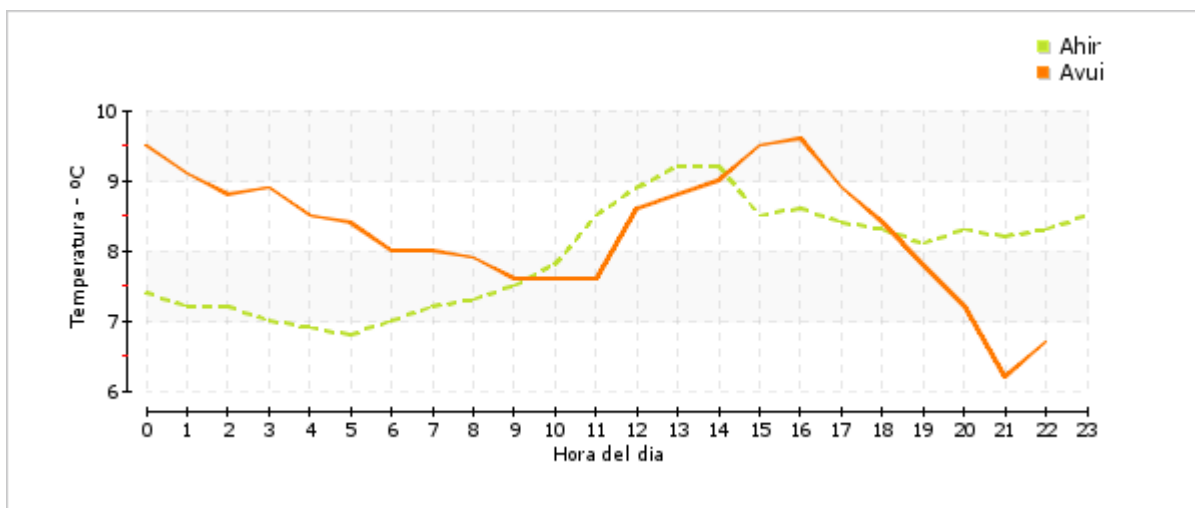
**Previsió del temps**

Avui	Demà	Dilluns
Cel cobert	Sol	Sol
-1 °C - 10 °C	0 °C - 10 °C	-1 °C - 10 °C
Vent: dèbil	Vent: dèbil	Vent: dèbil

**Tweets més recents @meteoandorra**

- Andorra la Vella - 1040m (16:02h) :: Temp. 8.5°C - Hum. 29% - Press. 924mbar
- Andorra la Vella - 1040m (12:02h) :: Temp. 5.0°C - Hum. 33% - Press. 927mbar
- Previsió per avui dissabte :: Cel cobert :: Temp. Min. 0°C - Temp. Màx. 9°C - Vent dèbil
- Andorra la Vella - 1040m (08:02h) :: Temp. 3.8°C - Hum. 30% - Press. 926mbar
- Andorra la Vella - 1040m (04:02h) :: Temp. 4.1°C - Hum. 21% - Press. 926mbar

Finalment, per tal d'acabar d'explicar la implementació de l'aplicació web, considero necessari detallar que per generar els gràfics lineals de les vistes dels casos d'ús "Consultar resum diari" i "Consultar resum anual", he utilitzat la llibreria [pChart 2.1.3](#), una classe PHP que permet crear imatges i gràfics lineals amb anti-aliàs i un gran nivell de detall, tal i com es pot observar en la figura següent:



### **5.3 - Desenvolupament del servei web**

Els serveis web (en anglès, *web services*) són un conjunt de protocols oberts i interoperables que permeten la invocació remota d'operacions entre sistemes heterogenis. Utilitzen el protocol HTTP com a canal de transport i el llenguatge de marques XML com a llenguatge de codificació d'informació.

Un d'aquests conjunts de protocols que darrerament ha tingut una gran acceptació i creixement ha estat l'especificació del protocol SOAP (Simple Object Access Protocol), desenvolupat inicialment per les companyies Userland i Microsoft al 1998. Aquesta especificació ha passat a ser un estàndard del W3C (World Wide Web Consortium) liderada per l'XML Protocol Working Group.

Els principals avantatges que es poden destacar del protocol SOAP són:

#### **1) Interoperabilitat**

Com utilitza estàndards oberts afavoreix la comunicació entre sistemes heterogenis.

#### **2) Seguretat**

Com utilitza el protocol HTTP com a canal de transport garanteix una comunicació a través de tallafocs.

### 3) Desacoblament

Els clients i servidors no s'han de conèixer entre ells a priori per comunicar-se.

### 4) Disponibilitat

Un servei web està sempre disponible a Internet i pot ser invocat per diferents clients en moments diferents.

### 5) Eficiència

Com utilitza el protocol HTTP, aquest no manté una connexió permanent entre client i servidor, la comunicació es produeix només quan és necessària.

No obstant, per tal que els diferents sistemes que interactuaran amb l'aplicació desenvolupada puguin invocar els mètodes del servei web, serà necessari crear un contracte que defineixi el servei ofert, els seus mètodes, els tipus dels paràmetres que necessitaran els mètodes i el resultat d'aquests.

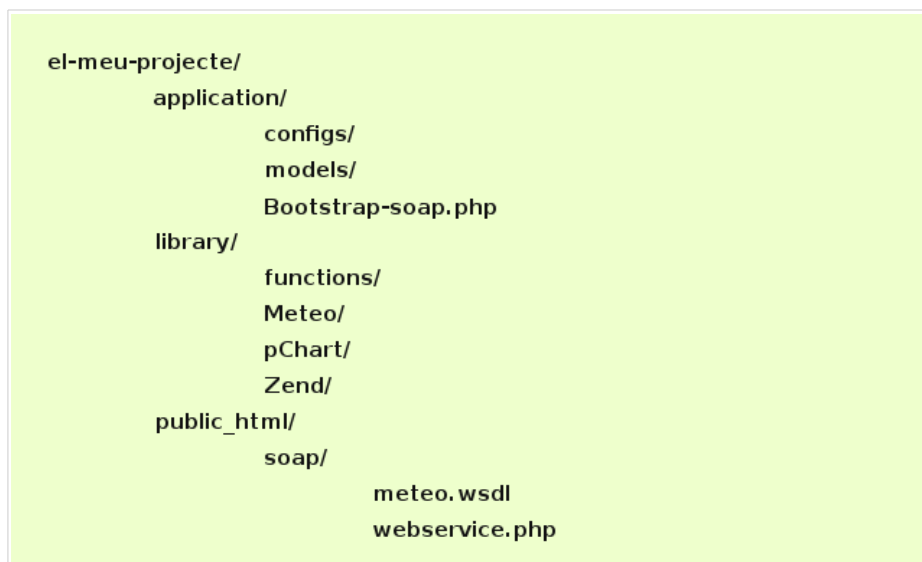
WSDL és un llenguatge XML estàndard del W3C que defineix els mètodes oferts pel servei, els missatges de sol·licitud i resposta intercanviats, incloent-hi el tipus de paràmetres dels mètodes, i dels protocols de comunicació. Per poder invocar els serveis oferts remotament des d'un client, només cal que el client localitzi el contracte WSDL mitjançant una crida a l'URL del mateix com a localitzador directe.

Tal i com plantejava en el capítol 2 - Descripció i objectius, aquest projecte ha de permetre la interacció entre diferents sistemes heterogenis (d'entrada he plantejat la interacció amb sistemes Android, però qui sap si més endavant interessarà la interacció amb dispositius que utilitzin el sistema iOS), és per aquest motiu i pels avantatges esmentats anteriorment, que s'ha decidit implementar la interacció amb el sistema explicat mitjançant un servei web amb l'especificació SOAP, i amb WSDL com a contracte per definir els mètodes que oferirà el sistema descrit.

A l'igual que en el desenvolupament de l'aplicació web, també s'ha optat per desenvolupar el servei web en llenguatge PHP juntament amb l'eina de treball Zend Framework pels motius que tot seguit s'exposen:

- 1) L'eina de treball Zend Framework disposa del component SoapServer que es capaç de gestionar tot tipus de peticions fetes a través del protocol SOAP,
- 2) L'eina de treball Zend Framework disposa del component AutoDiscover que permet crear de forma automàtica el contracte WSDL del servei SOAP ofert,
- 3) S'ha pogut aprofitar gran part del codi generat en l'aplicació web (les classes de la capa model, les funcions per generar els gràfics lineals, etc.), estalviant així molta feina de desenvolupament.

En el cas del servei web, l'estructura resultant de directoris ha estat la següent:



- application: és el directori que conté l'aplicació i a on s'emmagatzemen les classes de la capa model, les configuracions del servei SOAP i l'arxiu Bootstrap-soap.php,
- Bootstrap-soap.php: la finalitat d'aquest arxiu és la d'inicialitzar els components per fer que aquests estiguin disponibles pel servei web,
- library: és on es guarden les llibreries comunes de l'aplicació i la classe Meteo que conté els mètodes oferts pel servei web,
- public\_html: aquest directori conté tots els arxius públics de l'aplicació,
- meteo.wsdl: és el contracte WSDL on es definiran els mètodes oferts pel servei,
- webservice.php: la finalitat d'aquest arxiu és doble:
  - 1a) crear de forma automàtica el contracte WSDL mitjançant el component AutoDiscover,
  - 2a) gestionar totes les peticions fetes a través del protocol SOAP mitjançant el component SoapServer.

D'aquesta manera el fitxer WSDL es pot localitzar de forma directa mitjançant una crida a la URL del mateix fitxer.

(per exemple, en el servidor de producció el fitxer WSDL s'accedeix a través de la URL: <http://www.meteoandorra.com/soap/webservice.php?wsdl>)

```
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="http://www.meteoandorra.com/soap/webservice.php" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap-enc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/" name="Manager" targetNamespace="http://www.meteoandorra.com/soap/webservice.php">
  <types>...</types>
  <portType name="ManagerPort">
    <operation name="hello_web_service">
      <documentation>Return 'Hello web service!</documentation>
      <input message="tns:hello_web_serviceIn"/>
      <output message="tns:hello_web_serviceOut"/>
    </operation>
    <operation name="sum">
      <documentation>Return the sum of two variables</documentation>
      <input message="tns:sumIn"/>
      <output message="tns:sumOut"/>
    </operation>
    <operation name="recent_data">
      <documentation>Return recent data recieved form Weather station</documentation>
      <input message="tns:recent_dataIn"/>
      <output message="tns:recent_dataOut"/>
    </operation>
    <operation name="weather_forecast">
      <documentation>Return weather forecast</documentation>
      <input message="tns:weather_forecastIn"/>
      <output message="tns:weather_forecastOut"/>
    </operation>
    <operation name="daily_summary">
      <documentation>Return weather daily summary</documentation>
      <input message="tns:daily_summaryIn"/>
      <output message="tns:daily_summaryOut"/>
    </operation>
    <operation name="annual_summary">
      <documentation>Return weather annual summary</documentation>
      <input message="tns:annual_summaryIn"/>
      <output message="tns:annual_summaryOut"/>
    </operation>
  </portType>
  <binding name="ManagerBinding" type="tns:ManagerPort">...</binding>
  <service name="ManagerService">...</service>
  <message name="hello_web_serviceIn"/>
  <message name="hello_web_serviceOut">...</message>
  <message name="sumIn">...</message>
  <message name="sumOut">...</message>
  <message name="recent_dataIn">...</message>
  <message name="recent_dataOut">...</message>
  <message name="weather_forecastIn">...</message>
  <message name="weather_forecastOut">...</message>
  <message name="daily_summaryIn">...</message>
  <message name="daily_summaryOut">...</message>
  <message name="annual_summaryIn">...</message>
  <message name="annual_summaryOut">...</message>
</definitions>
```

### Fitxer WSDL del servidor de producció (entorn real)

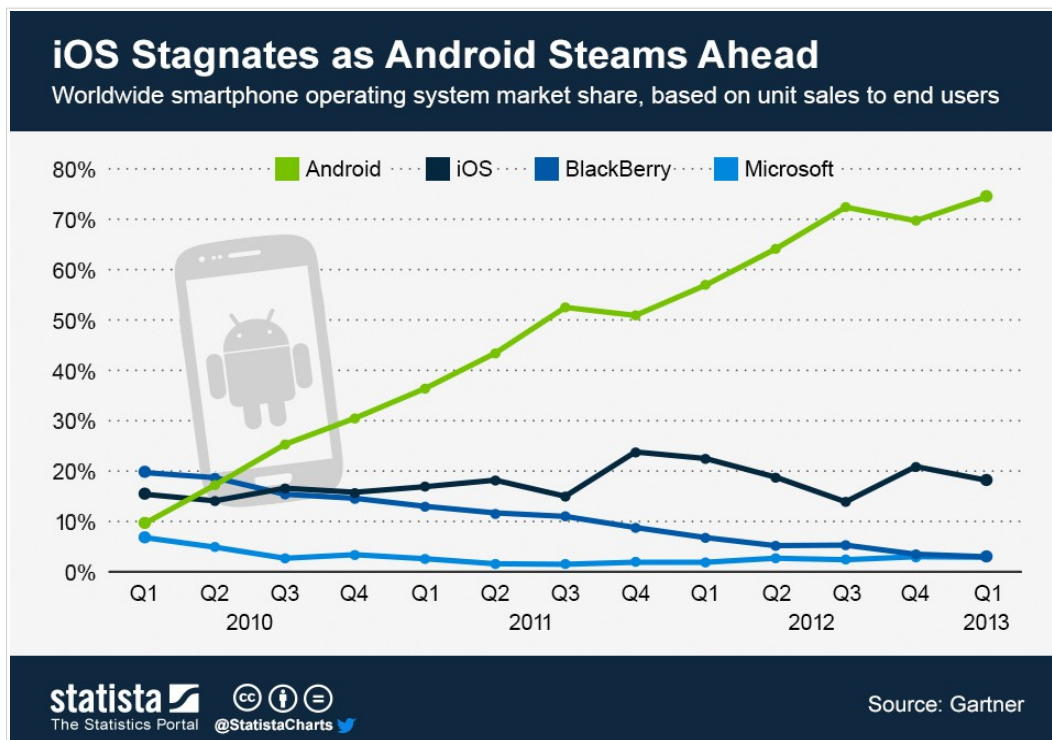
Per acabar, vull puntualitzar que s'ha programat el servei web per tal que retorni els resultats en format JSON.

JSON (JavaScript Object Notation), és un format lleuger que permet intercanviar dades sense la necessitat d'utilitzar l'estàndard XML, reduint d'aquesta manera el flux de dades entre els sistemes.

Com cada vegada hi ha més suport d'aquest format per a diferents llenguatges (C, C++, C#, Java, JavaScript, Objective-C, PHP, Python, etc.), i com que el servei web no ha de tornar estructures complexes de dades, he considerat convenient que torni els resultats en aquest format per tal d'augmentar la simplicitat i la velocitat de comunicació entre el servei SOAP i els sistemes client.

#### **5.4 - Desenvolupament de l'aplicació Android**

S'ha decidit desenvolupar aquesta aplicació per a sistemes Android perquè avui en dia, i segons fonts consultades, aquest sistema per a *smartphones* té gairebé una quota de mercat del 80%, tal i com es pot veure en la figura següent:



**Quota de mercat de sistemes per a *smartphones*  
(font consultada: <http://www.statista.com/>)**

Les característiques principals d'aquest sistema són:

- suport per a diferents tipus de tecnologies de connectivitat (GSM/EDGE, UMTS, Bluetooth, Wi-Fi, WiMAX, ...),
- s'adapta a pantalles de diferents resolucions,
- suport per a pantalles tàctils i altre tipus de hardware addicional (GPS, acceleròmetres, giroscopis, sensors de proximitat, de llum, de pressió, ...),
- disposa de llibreries gràfiques 2D, i de llibreries gràfiques 3D basades en l'especificació OpenGL ES 2.0,
- té suport per a diferents formats multimèdia (H.263, H.264, MPEG-4, AAC, HE-AAC, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, ...),
- té la capacitat de dur a terme simultàniament diverses activitats (és un sistema multitasca).

Hi ha una gran quantitat de llenguatges que permeten programar per a Android, des de VisualBasic (amb Basic4Android), C# (utilitzant l'entorn de programació Xamarin.Android), HTML5, etc. Tanmateix, s'ha optat per utilitzar el llenguatge de programació oficial (Java), perquè crec que és el llenguatge més estès per a aquesta plataforma i del que es troba més documentació.



L'entorn de desenvolupament o *Software Development Kit* (d'ara endavant, SDK) d'Android pot descarregar-se de la pàgina <http://developer.android.com/sdk/index.html>

Proporciona les llibreries i les eines de desenvolupament necessàries per tal de crear, provar i depurar aplicacions Android. A més, inclou els components essencials del SDK, una versió de l'IDE Eclipse amb funcions d'ADT (*Android Developer Tools*), i les imatges més recents del sistema Android per depurar les aplicacions mitjançant un emulador. En definitiva, tot un conjunt d'eines per agilitzar al màxim el desenvolupament d'aplicacions Android.

No obstant, en el present projecte no s'ha utilitzat l'emulació del sistema durant el desenvolupament a conseqüència del seu mal funcionament. Personalment ha estat molt més pràctic i àgil desenvolupar directament amb el dispositiu Nexus 7 connectat per USB a l'ordinador, activant des de les preferències del sistema les "Opcions de desenvolupament" i la "Depuració USB".



Android proporciona els seus propis patrons de disseny per utilitzar-los en funció de la finalitat de l'aplicació que es vol implementar. El millor de tot és que des de la seva pàgina web també es poden descarregar exemples força complets per tal de no començar a desenvolupar una aplicació des de zero.

He considerat adient utilitzar el patró *Navigation Drawer* perquè crec que és el patró que millor s'escau per l'aplicació que s'ha volgut desenvolupar.

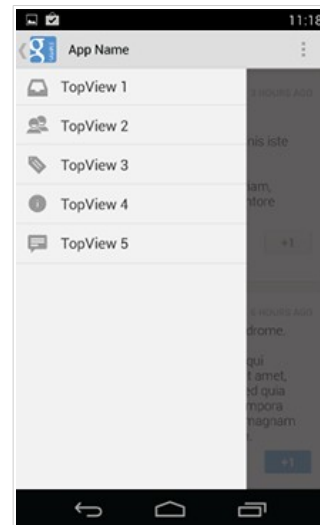
Aquest patró de disseny parteix d'un panell de navegació que s'amaga i es mostra des del marge esquerra de la pantalla, ensenyant les principals opcions de navegació de l'aplicació.

L'usuari pot ocultar i fer aparèixer aquest panell de navegació lliscant la pantalla des de la vora esquerra de la pantalla o tocant la icona de l'aplicació situada en la barra d'acció de l'aplicació.

Quan el panell de navegació s'expandeix, aquest se superposa sobre el contingut de l'aplicació, però no sobre la barra d'acció.

Seguint el disseny inicial que s'ha explicat en el capítol 4 de la memòria, he optat per afegir en el panell de navegació una opció per a cadascun dels casos d'ús plantejats. Així, el panell de navegació de la present aplicació ha quedat amb les següents opcions:

- Opció "Dades actuals"
- Opció "Resum diari"
- Opció "Resum anual"



Sense entrar massa en detall, l'estructura de directoris del projecte ha quedat de la següent manera:

```
el-meu-projecte/  
  src/  
    MainActivity.java  
    Webservice.java  
  libs/  
    android-support-v4.jar  
    gson-2.2.4.jar  
    ksoap2-android-assembly-3.0.0.jar  
  res/  
    AndroidManifest.xml
```

- src: és el directori que conté tot el codi font de l'aplicació, classes, classes auxiliars, etc.
- MainActivity.java: classe que permet gestionar l'activitat principal de l'aplicació, així com els seus comportaments,
- WebService.java: classe que permet gestionar les crides al servei web SOAP,
- libs: aquest directori conté les llibreries auxiliars de l'aplicació (normalment en format .jar),
- android-support-v4.jar: llibreria que permet mantenir la compatibilitat entre diferents versions d'Android,
- gson-2.2.4.jar: llibreria Java que permet tractar cadenes en representació JSON,
- ksoap2-android-assembly-3.0.0.jar: llibreria Java que permet gestionar comunicacions SOAP de forma lleugera i eficient per a clients de la plataforma Android,
- res: conté altres fitxers necessaris del projecte: els fitxers amb les descripcions de les vistes, les imatges, logotips, traduccions, etc.
- AndroidManifest.xml: aquest fitxer conté les definicions dels aspectes principals de l'aplicació: nom de l'aplicació, versió, pantalles, llibreries addicionals que utilitza, els permisos que necessita per executar-se, entre d'altres.

En el punt on es detallava la implementació del servei web, s'ha explicat que al final s'havia decidit implementar la interacció utilitzant l'especificació SOAP i els contractes WSDL. De la mateixa manera també s'ha explicat que les dades es transmetran utilitzant el format JSON. Resulta que Android no proporciona en el seu SDK llibreries per gestionar comunicacions amb especificació SOAP i dades en format JSON, és per aquest motiu que s'han utilitzat llibreries addicionals (gson-2.2.4.jar i ksoap2-android-assembly-3.0.0.jar) per tal de facilitar la feina a l'hora d'establir el lligam entre el servei web i l'aplicació Android.

Abans de finalitzar aquest capítol crec convenient explicar com funciona la comunicació entre el servei web i l'aplicació Android, per tal de detallar els tipus de dades que s'intercanvien.

Tal i com s'acaba de veure l'aplicació Android té tres opcions, les quals permeten consultar les dades actuals, el resum diari i el resum anual. Això implica que el servei web ha de contenir uns mètodes que permetin a l'aplicació Android obtenir les dades d'aquestes opcions.

Per tal de simplificar-ho s'ha decidit implementar en el servei web un mètode per a cada opció de consulta.

D'aquesta manera hi ha un total de tres mètodes, que he anomenat:

- *recent\_data ()* per obtenir les dades que es mostraran en l'opció "Dades actuals",
- *daily\_summary ()* per obtenir les dades que es mostraran a l'opció "Resum diari",
- *annual\_summary()* per obtenir les dades que es mostraran a l'opció "Resum diari".

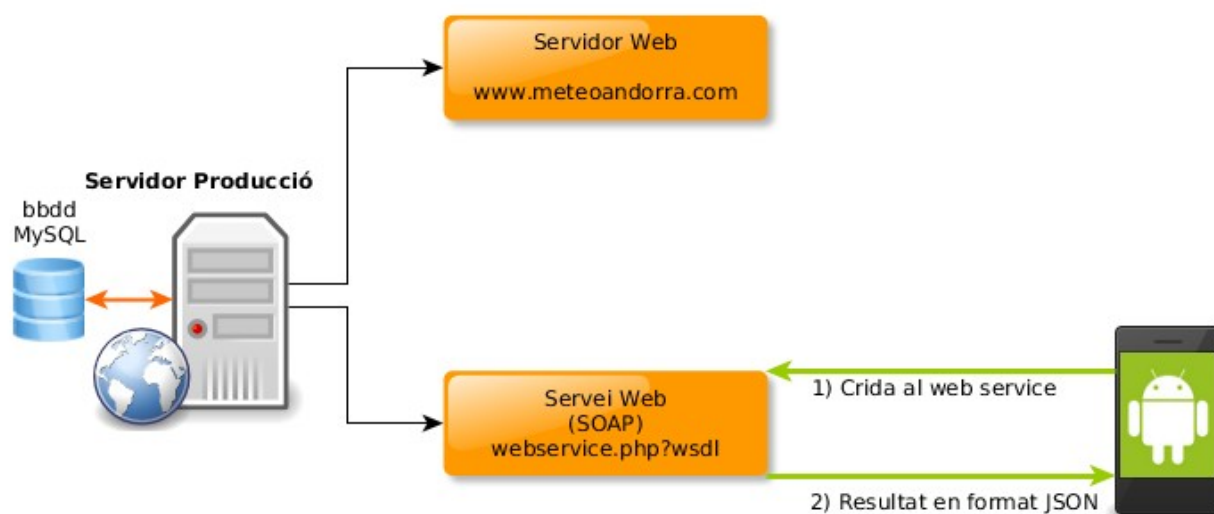
**Observació:** si ens hi fixem bé, els noms dels mètodes del servei web es corresponen amb els mètodes definits en el fitxer WSDL de la pàgina 38.

Sigui quina sigui l'opció, l'esquema de funcionament és el mateix: l'aplicació Android crida a un dels mètodes del servei web, aquest executa les operacions necessàries per tornar els resultats (consultar la base de dades, construir el resultat, ...) i torna les dades a l'aplicació Android en format JSON.

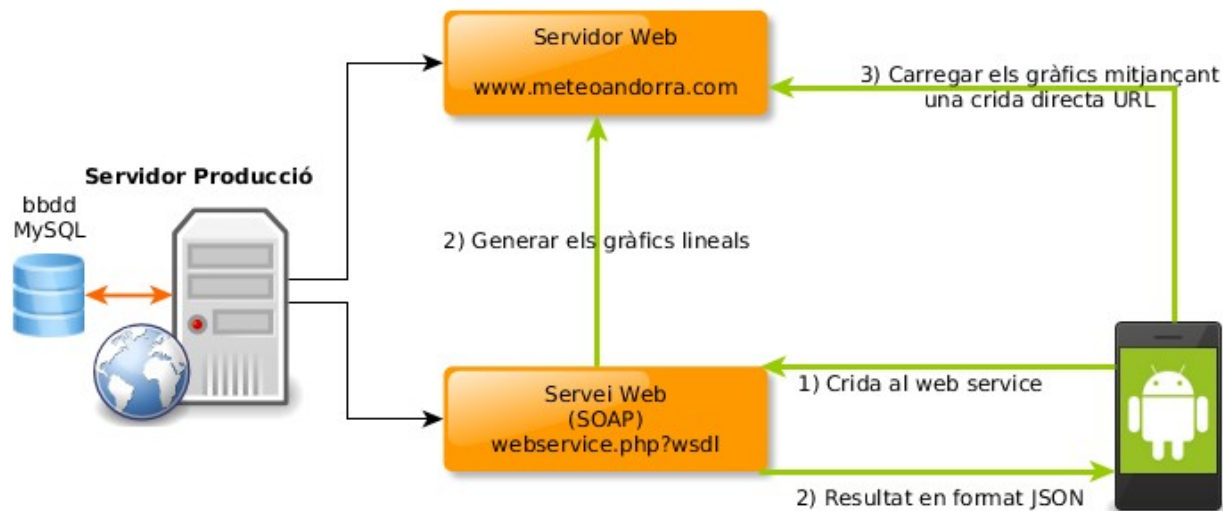
No obstant, els mètodes *daily\_summary ()* i *annual\_summary()* tenen un comportament diferent, ja que, a banda de tornar els seus resultats a l'aplicació Android, també han de generar els gràfics lineals amb les progressions diàries i anuals. Aquests gràfics no s'envien juntament amb el resultat en format JSON (tal i com s'ha explicat anteriorment), sinó que es guarden en una carpeta pública de l'aplicació web, i correspon a l'aplicació Android carregar els gràfics mitjançant una crida directa a la URL de les imatges.

Les crides a les operacions del servei web poden resumir-se amb els següents esquemes:

#### Crida al mètode *recent\_data ()*



Crida als mètodes *daily\_summary ()* i *annual\_summary()*



## **6 - Conclusions i valoracions**

Personalment ha estat molt satisfactori desenvolupar aquest projecte i no puc més que remarcar que l'experiència ha estat altament positiva.

La veritat és que genera una certa incredulitat, a la vegada que molta il·lusió, veure com un petit projecte que va començar publicant tweets en mode de proves a la xarxa social Twitter, ha anat creixent fins a convertir-se en un projecte molt complet on actualment hi interactuen diferents sistemes de vàries tecnologies.

Aquest projecte m'ha donat la possibilitat d'estudiar i desenvolupar aplicacions en diferents llenguatges de programació (Python, PHP, Java), conèixer nous protocols i solucions que permeten la interacció entre sistemes heterogenis (mitjançant els serveis web i el protocol SOAP), o aprendre a programar per a una nova plataforma de dispositius mòbils com és el cas del sistema operatiu Android. D'aquesta manera he pogut posar en pràctica molts coneixements adquirits mentre cursava algunes de les assignatures d'aquest segon cicle, com ara bases de dades, enginyeria del programari, xarxes, sistemes distribuïts, metodologia i gestió de projectes informàtics, etc.

Tanmateix, no només he aprofundit en coneixements sobre tecnologies relacionades amb els camps de l'enginyeria informàtica, sinó que també m'ha servit per introduir-me a l'apassionant món de la meteorologia, les estacions meteorològiques i les xarxes de sensors, les quals estic convençut que seran de gran importància en un futur pròxim.

Un dels punts claus del projecte ha estat lligar-lo amb la xarxa social Twitter. Tot i ser la part menys complicada del projecte, és on realment s'ha anat veient diàriament si el projecte interessava o no. A més, ha brindat la possibilitat que molts usuaris hi poguessin interactuar de forma directa, aportant idees, comentaris, observacions, etc. Queda patent que avui en dia la meteorologia interessa molt, i les xarxes socials són una part molt important de les aplicacions informàtiques.

L'ús de l'eina de treball Zend Framework també ha estat un element clau d'aquest projecte ja que m'ha permès desenvolupar a la vegada l'aplicació web i el servei web, reutilitzant d'aquesta manera codi entre les dues aplicacions i reduint així el temps de desenvolupament.

Respecte el sistema Android he de confessar que al principi em va costar una mica entendre com s'estructuren els projectes d'aquest sistema, comprendre com es defineixen les interfícies, o aprendre els seus tipus de patrons de disseny. No obstant, el web oficial orientat al seu desenvolupament està molt ben explicat i conté un gran nombre de petits projectes que m'han estat de gran utilitat.

Com a apunt final només em queda afegir que valoro molt favorablement haver fet aquest projecte final. Crec que, juntament amb l'assignatura "Metodologia de gestió de projectes informàtics", ha estat la millor manera d'acabar aquests estudis.

Si miro enrere, puc afirmar sense cap mena de dubte que cursar aquest segon cicle ha estat una decisió molt encertada i l'esforç ha valgut la pena, ja que m'ha permès créixer tant en l'àmbit professional com en el personal.

## **6.1 - Aspectes a millorar**

Tant l'esquema de funcionament general, com algun element del maquinara seleccionat, o cadascuna de les aplicacions desenvolupades, poden ser millorats considerablement:

### Sistema general

- Actualment el servidor local i l'estació meteorològica estan connectats de forma normal a la xarxa elèctrica. Això vil dir que si es produeix un tall de corrent, el sistema queda inoperant i es deixa de recollir les dades de l'estació meteorològica. És per aquest motiu que seria important connectar el servidor local i l'estació a un SAI, per tal de no perdre les dades de l'estació durant la interrupció del servei.

### Estació meteorològica

- Encara que el fabricant Oregon Scientific pugui semblar un referent d'estacions meteorològiques, m'he adonat que això no és així. Durant tot el període de funcionament (més d'un any) he pogut veure que el sensor de temperatura dona errors de lectures quan més alta és la temperatura. D'altra banda, pel que he pogut llegir en diferents fòrums especialitzats el sensor d'humitat tampoc ofereix lectures gaire correctes. Si a tot això li sumem que en un any s'ha hagut de substituir l'estació perquè es començava a penjar cada 10 dies, és evident que aquesta marca no ofereix el mínim de garanties pel tipus de servei que es vol oferir.
- Un altre aspecte a millorar seria la ubicació de l'estació ja que cal complir un mínim de requisits per aconseguir lectures correctes dels sensors (l'estació ha d'estar a un mínim d'alçada del sòl, no pot estar molt pròxima a edificis, etc.).

Escollir un altre fabricant d'estació meteorològica i canviar la ubicació d'aquesta ajudaria a oferir un servei més fiable i professional.

### Script de publicació de les dades meteorològiques a la xarxa social Twitter

- Considero que la formula de publicar un tweet cada quatre hores és molt encertada ja que permet seguir l'evolució de les dades durant el dia, i a la vegada no satura els seguidors amb excés d'informació. Tanmateix, es podria complementar amb la publicació de captures d'imatges fetes des d'una webcam per tal de fer el compte de *@meteoandorra* més atractiu.

### Aplicació web

- El disseny actual de l'aplicació web està pensat per a una resolució i mida de pantalla concret. És per aquest motiu que estaria bé replantejar el disseny i fer-lo de tipus *responsive*, d'aquesta manera l'aplicació es podria consultar correctament des de dispositius mòbils, *tablets*, etc.

### Servei web

- El servei web s'ha construït de tal manera que qualsevol persona pugui accedir als seus mètodes i pugui obtenir les dades meteorològiques. Seria convenient introduir un mecanisme d'autenticació i fer que els usuaris que hi vulguin accedir s'hagin de registrar. Així es podria controlar l'accés al servei web i obtenir més dades d'utilització del servei.

### Aplicació Android

Pel que fa a l'aplicació Android crec que seria interessant incloure:

- El *timeline* de la xarxa social Twitter (els tweets publicats) dins una nova opció de l'aplicació. Així hi hauria la possibilitat de consultar un altre canal d'informació dins la mateixa aplicació.
- Un sistema d'alertes per poder avisar als usuaris amb missatges sobre algun tipus d'incidència. Com per exemple quan la temperatura registrada és inferior als zero graus, quan hi ha risc d'allaus, o quan s'ha activat un tipus de pla d'emergència i/o protecció civil.

## **6.2 - Autoavaluació**

Aquest treball m'ha servit per aprendre noves tecnologies i per posar en pràctica múltiples coneixements adquirits al llarg d'un bon nombre de les assignatures cursades.

Crec haver complert amb tots els objectius marcats a l'inici, malgrat que en algun punt m'he desviat de la planificació inicial per tal de millorar el present treball. És per aquest motiu que estic molt satisfet amb el desenvolupament i execució del projecte realitzat.

## **6 - Bibliografia i pàgines web consultades**

- **Zend Framework et PHP - Programmation par composants -**

Capítol 10: Services Web

Autor: Christophe HARO  
Editorial: Éditions ENI - Décembre 2008

- **Zend Framework - Bien développer en PHP -**

Capítol 12: Interopérabilité et services web

Autors: Julien Pauli i Guillaume Ponçon  
Editorial: Éditions EYROLLES - Novembre 2008

- **Enginyeria del programari orientat a l'objecte**

Apunts UOC  
Autors: Jordi Fernández González , Jordi Pradel i Miquel , José Antonio Raya Martos

- **Arquitectura de sistemes distribuïts**

Apunts UOC  
Autors: Leandro Navarro Moldes , Joan Manuel Marquès i Puig

- **Metodologia de gestió de projectes informàtics**

Apunts UOC  
Autors: José Ramón Rodríguez , Pere Mariné Jové

- **<http://www.python.org/>**

Pàgina oficial del llenguatge de programació Python

- **<https://pypi.python.org/pypi/twitter>**

Pàgina oficial de la llibreria Python Twitter Tools

- **<http://www.php.net/>**

Pàgina oficial del llenguatge de programació PHP

- **<http://framework.zend.com/>**

Pàgina oficial de l'eina de treball per a PHP Zend Framework



- <http://www.pchart.net/>  
Pàgina oficial de la classe pChart per a PHP
- <http://www.android.com/>  
Pàgina oficial del sistema operatiu Android
- <http://developer.android.com/sdk/index.html>  
Pàgina des d'on descarregar l'SDK oficial d'Android
- <http://developer.android.com/develop/index.html>  
Pàgina oficial de programació d'Android
- <https://code.google.com/p/google-gson/>  
Llibreria Java pel tractament de dades en format JSON
- <https://code.google.com/p/ksoap2-android/>  
Llibreria SOAP pel sistema operatiu Android
- <http://stackoverflow.com/>  
Portal de programació on trobar preguntes i respostes
- <http://stackexchange.com/>  
Portal de programació on trobar preguntes i respostes
- <http://www.oregonscientific.com/>  
Pàgina oficial del fabricant d'estacions meteorològiques Oregon Scientific
- <https://www.google.com/nexus/7/>  
Pàgina oficial del dispositiu Nexus 7 de Google
- <http://www.meteoclimatic.com/>  
Portal col·laboratiu que pretén facilitar l'accés a informació meteorològica espanyola
- <http://www.wviewweather.com/>  
Pàgina oficial del programari per a estacions meteorològiques wview
- <http://www.wikipedia.org/>

## **7 - Glossari**

- ADT - Android Development Tools: Eines de desenvolupament d'Android.
- Android: Conjunt de programari per a telèfons mòbils. Inclou el sistema operatiu, programari intermediari i aplicacions.
- cron: Programa informàtic que realitza tasques a intervals regulars de temps.
- Framework: Eina o marc de treball per a la organització del desenvolupament de software.
- Front Controller (FC): Patró arquitectònic per al desenvolupament de software.
- GNU/Linux: Sistema operatiu complet amb conjunt d'utilitats de GNU, entorns d'escriptori i altres aplicacions. El nucli Linux fou desenvolupat inicialment per Linus Torvalds.
- HTTP: Protocol per a l'intercanvi de documents d'hipertext i multimèdia al web.
- IDE: Entorn integrat de desenvolupament.
- Java: Llenguatge de programació orientat a objectes dissenyat el 1990 per James Gosling.
- JSON: Estàndard obert basat en text, dissenyat per a l'intercanvi de dades.
- Model View Controller (MVC): Patró arquitectònic per al desenvolupament de software.
- Patró: Solució general a un problema comú i recurrent en el disseny de programari.
- Python: Llenguatge de programació d'alt nivell creat per Guido van Rossum l'any 1991.
- PHP: Llenguatge de programació interpretat creat l'any 1994 per Rasmus Lerdof.
- script: Guió o conjunt d'instruccions informàtiques.
- SAI: Sistema d'alimentació ininterrompuda.
- SDK: Kit de desenvolupament de software.
- Servei web: Col·lecció de protocols i estàndards que serveix per intercanviar dades entre aplicacions i sistemes informàtics heterogenis.
- SOAP: Protocol de comunicació dissenyat per intercanviar missatges en format XML.
- WSDL: Llenguatge en format XML que serveix per descriure serveis web.
- wview: Programari per a sistemes UNIX, que permet la recuperació de dades de diferents models d'estacions meteorològiques.

## Annex 1: Aplicació web

A continuació es mostren les vistes per a cadascun dels casos d'ús de l'aplicació web.

### Vista Consultar resum diari

The screenshot displays the 'Meteo Andorra' website interface. At the top, there is a navigation bar with language options: Català, English, Español, and Français. The main header features the 'Meteo Andorra' logo and the tagline '{ Dades meteorològiques del Principat d'Andorra }'. Below the header, there are two tabs: 'Resum diari' (selected) and 'Resum anual'. The main content area is divided into several sections:

- Estació:** Andorra la Vella
- Alçada:** 1040m
- Sensors:** Temperatura, Humitat, Pressió
- Selecció de data:** A calendar interface showing 'Gen.' and '2014'. A date picker is open, showing a grid of days from 1 to 31. The current date selected is the 4th of January.
- Temperatura:** A line graph showing temperature in degrees Celsius over a 24-hour period. The y-axis ranges from 6 to 12. The x-axis is labeled 'Hora del dia' (0 to 23). The graph shows a peak of 10.2°C at 12:25h and a minimum of 6.8°C at 09:15h.
- Dades actuals Andorra la Vella:** A box showing current weather data: Última actualització: 04-01-2014 a les 19:55h. Data: Temperatura: 8.1 °C, Humitat: 72 %, Pressió: 901 mbar.
- Previsió del temps:** A section for the next three days: Avui (Pluja, 5°C - 8°C, Vent: moderat), Demà (Mig ennuvolat, 1°C - 9°C, Vent: moderat), and Dilluns (Sol, 0°C - 14°C, Vent: moderat).
- Tweets més recents @meteoandorra:** A list of recent tweets from the official account, providing real-time weather updates.

Captura de pantalla on es pot veure els diferents elements de la vista Consultar resum diari: les dades de l'estació, el gràfic de temperatura pel dia actual, les dades de temperatura mínima i màxima diàries, un camp per poder seleccionar una data, entre d'altres.

També es pot veure el *helper* "Dades actuals" que s'explicava en la pàgina 34.

## Vista Consultar resum anual

**Meteo Andorra**  
{ Dades meteorològiques del Principat d'Andorra }

Català English Español Français

Resum diari **Resum anual**

• Estació: Andorra la Vella  
• Alçada: 1040m  
• Sensors:  
• Temperatura  
• Humitat  
• Pressió

Seleccionar un any  
2013 Buscar

**Temperatura mitjana mensual**

Temperatura - °C

■ 2013

Mes de l'any	Temperatura mitjana (°C)
Gen.	~5
Febr.	~4
Març	~8
Abr.	~11
Maig	~13
Juny	~18
Jul.	~21
Ag.	~21
Set.	~18
Oct.	~14
Nov.	~8
Des.	~7

**Gener**

- Mínima: 1.3 °C (22-01-2013 a les 10:10h)
- Màxima: 18.7 °C (06-01-2013 a les 15:20h)

**Febrer**

- Mínima: -5.9 °C (26-02-2013 a les 07:50h)
- Màxima: 17.3 °C (21-02-2013 a les 14:40h)

**Dades actuals Andorra la Vella**

Última actualització: 04-01-2014 a les 19:55h

- Temperatura: 8.1 °C
- Humitat: 72 %
- Pressió: 901 mbar

**Previsió del temps**

Avui	Demà	Dilluns
Pluja	Mig ennuvolat	Sol
5 °C - 8 °C	1 °C - 9 °C	0 °C - 14 °C
Vent: moderat	Vent: moderat	Vent: moderat

**Tweets més recents @meteoandorra**

Andorra la Vella - 1040m (20:02h) :: Temp. 8.1°C - Hum. 71% - Press. 900mbar

Andorra la Vella - 1040m (16:02h) :: Temp. 8.7°C - Hum. 70% - Press. 902mbar

Andorra la Vella - 1040m (12:02h) :: Temp.

Captura de pantalla on es pot veure els diferents elements de la vista Consultar resum anual: les dades de l'estació, el gràfic de temperatures mitjanes per l'any seleccionat, les temperatures mínimes i màximes mensuals, un camp desplegable per poder canviar d'any, etc.

## Vista Enviar missatge

The screenshot shows the 'Enviar missatge' (Send message) form on the Meteo Andorra website. The form is titled 'Contactar' and includes the following fields:

- El teu nom:
- El teu e-mail:
- Assumpte:
- Missatge:

Below the message field is a CAPTCHA section with the text: 'Repte CAPTCHA: El següent repte es fa per comprovar que sou un visitant humà i prevenir els enviaments automàtics de correu no desitjat.' The CAPTCHA image shows the text 'm187k9r' with a grid of dots. To the right of the CAPTCHA is a 'Codi:' field with an input box and an asterisk.

At the bottom of the form is an 'Enviar E-Mail' button.

The website header includes the Meteo Andorra logo and the tagline '{ Dades meteorològiques del Principat d'Andorra }'. The navigation menu shows 'Resum diari' and 'Resum anual'. The top right corner has language options: 'Català', 'English', 'Español', and 'Français'.

On the right side of the page, there are three widgets:

- Dades actuals Andorra la Vella**: Última actualització: 04-01-2014 a les 19:55h. Data: Temperatura: 8.1 °C, Humitat: 72 %, Pressió: 901 mbar.
- Previsió del temps**: A table showing weather for 'Avui', 'Demà', and 'Dilluns'.
- Tweets més recents @meteoandorra**: A list of recent tweets from the account.

Avui	Demà	Dilluns
Pluja	Mig ennuvolat	Sol
5 °C - 8 °C	1 °C - 9 °C	0 °C - 14 °C
Vent: moderat	Vent: moderat	Vent: moderat

Tweets més recents @meteoandorra
Andorra la Vella - 1040m (20:02h) :: Temp. 8.1°C - Hum. 71% - Press. 900mbar
Andorra la Vella - 1040m (16:02h) :: Temp. 8.7°C - Hum. 70% - Press. 902mbar
Andorra la Vella - 1040m (12:02h) :: Temp.

Captura de pantalla on es pot veure els diferents elements de la vista Enviar missatge: els diferents camps que s'enviaran per correu electrònic i un repte CAPTCHA per evitar l'enviament de correu no desitjat.

## Vista Avís legal

The screenshot shows the 'Avis legal' page on the website 'Meteo Andorra'. The page features a header with the site logo and navigation links for 'Resum diari' and 'Resum anual'. The main content area is titled 'Avis legal' and contains several sections: 'Titularitat', 'Continguts del web', and 'Crèdits'. To the right, there are sidebars for 'Dades actuals Andorra la Vella' and 'Tweets més recents @meteoandorra'. The background of the page is a scenic view of a town in a valley.

**Meteo Andorra**  
{ Dades meteorològiques del Principat d'Andorra }

Català English Español Français

Resum diari Resum anual

### Avis legal

**Titularitat**

Sogas Industries Inc. és propietària i gestora d'aquest lloc web.

Sogas Industries Inc. està inscrita al registre de Societats del M.I. Govern Platanian, amb el número de registre 12345 i data de constitució 10/01/2014.

El número de registre tributari de Sogas Industries Inc. és Z-750-KAWA.

**Continguts del web**

Sogas Industries Inc. es reserva el dret d'afegir, modificar o eliminar en qualsevol moment la informació exposada en aquest lloc web, així com suspendre'n temporalment l'accés per introduir-hi millores o per dur a terme el manteniment, sense necessitat de comunicació prèvia.

Sogas Industries Inc. no s'assumeix cap responsabilitat derivada de les dades mostrades en aquest lloc web, ni de l'ús que es pugui arribar a fer.

Sogas Industries Inc. no pot garantir que les dades mostrades en aquest lloc web siguin exemptes d'error, però sí vetllarà perquè no n'hi hagi i per esmenar-los tan aviat com sigui possible.

**Crèdits**

El logotip de Meteo Andorra ha estat creat per [Montse Mayol Mas](#).

Les icones de les previsions meteorològiques han estat creades per [xiao4](#) i obtingudes de [DeviantArt.com](#).

### Dades actuals Andorra la Vella

Última actualització: 04-01-2014 a les 19:55h

- Temperatura: 8.1 °C
- Humitat: 72 %
- Pressió: 901 mbar

### Previsió del temps

### Tweets més recents @meteoandorra

Andorra la Vella - 1040m (16:02h) :: Temp. 12.2°C - Hum. 45% - Press. 915mbar

Andorra la Vella - 1040m (12:02h) :: Temp. 9.7°C - Hum. 47% - Press. 917mbar

Previsió per avui dijous :: Sol :: Temp. Mín. -3°C - Temp. Màx. 13°C - Vent moderat

Andorra la Vella - 1040m (08:02h) :: Temp. 7.8°C - Hum. 47% - Press. 917mbar

Andorra la Vella - 1040m (04:02h) :: Temp. 8.2°C - Hum. 47% - Press. 918mbar

Captura de pantalla on es pot veure els diferents elements de la vista Avis legal.

## **Annex 2: Aplicació Android**

A continuació es mostren les vistes per a cadascun dels casos d'ús de l'aplicació Android.

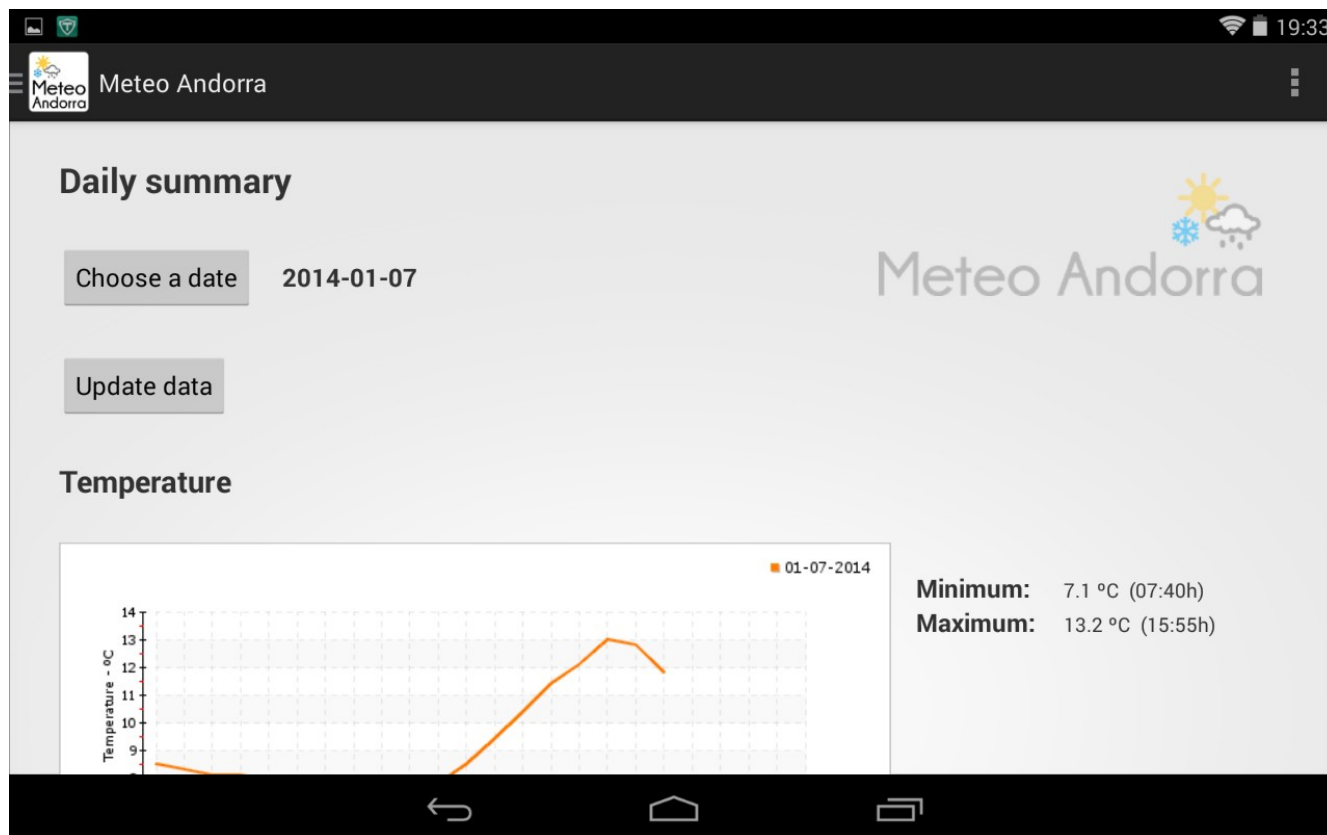
### **Vista Dades actuals**



Captura de pantalla on es pot veure els diferents elements de la vista Dades actuals.

Fent clic al botó 'Actualitzar dades' permet recuperar les dades actuals de temperatura, humitat i pressió mitjançant una crida al servei web.

## Vista Resum diari



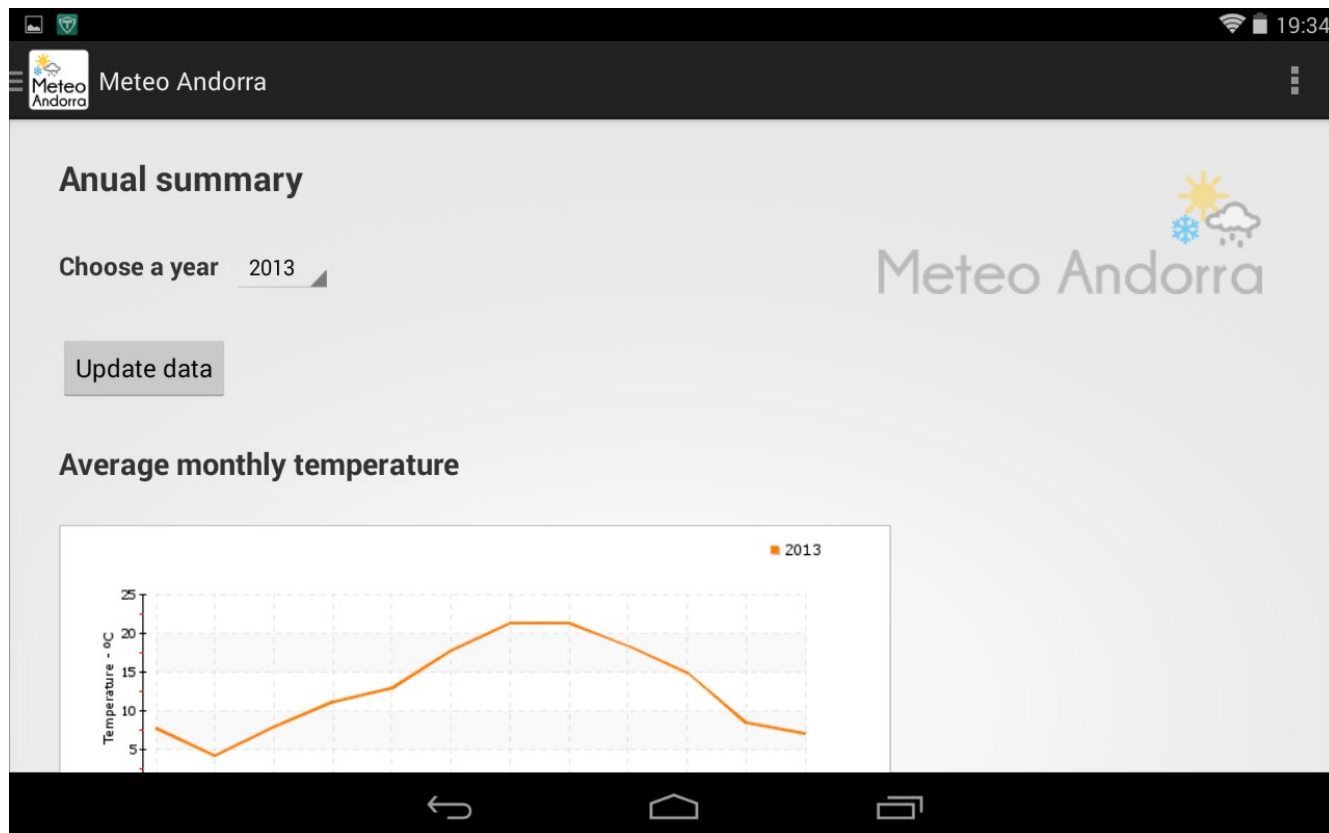
Captura de pantalla on es pot veure els diferents elements de la vista Resum diari.

El botó 'Seleccionar una data' obre un diàleg per poder seleccionar una data.

El botó 'Actualitzar dades' permet recuperar els gràfics i les dades diàries pel dia seleccionat, mitjançant una crida al servei web.



## Vista Resum anual

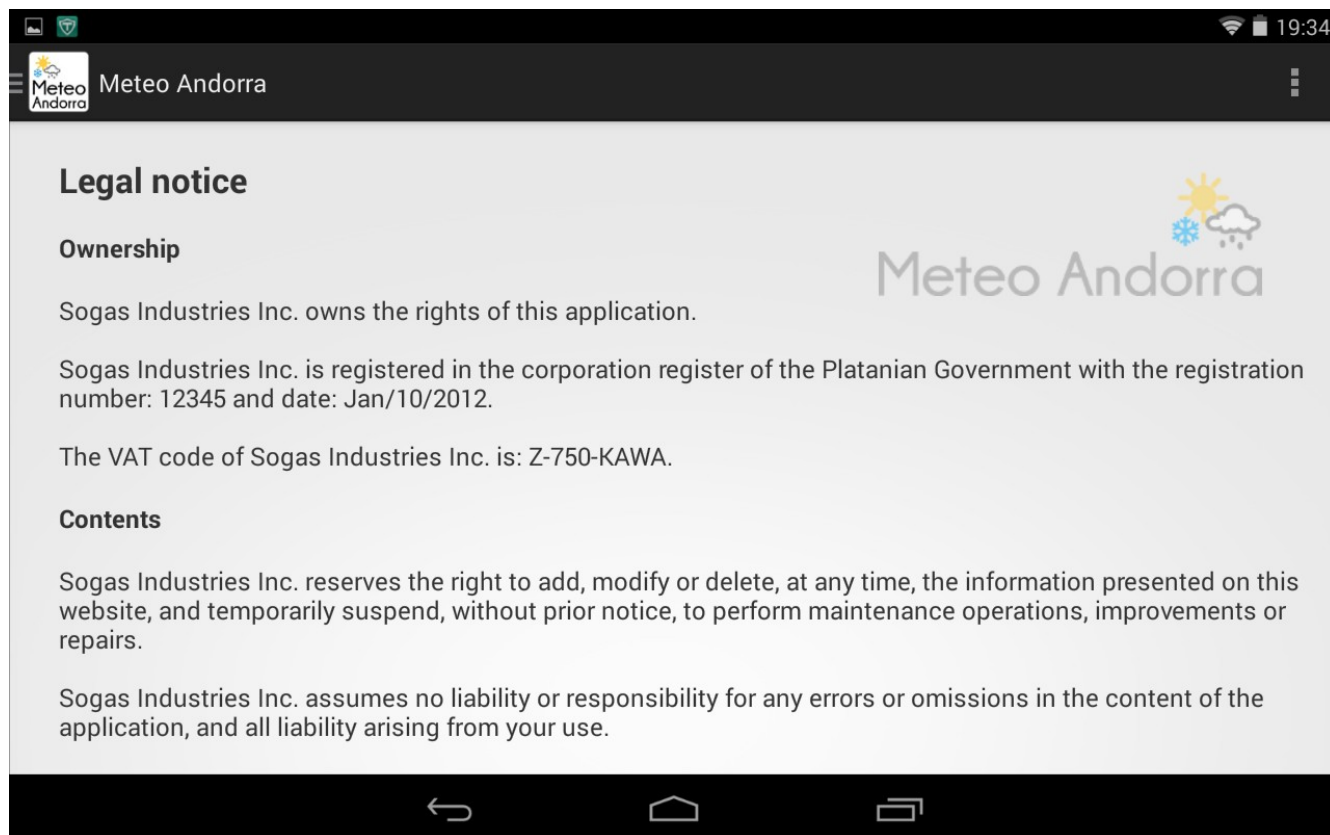


Captura de pantalla on es pot veure els diferents elements de la vista Resum anual.

Inclou un camp desplegable per poder seleccionar l'any pel quin poder consultar el resum.

El botó 'Actualitzar dades' permet recuperar els gràfics i les dades anuals per l'any seleccionat, mitjançant una crida al servei web.

## **Vista Avís legal**



No estava inclòs en la definició inicial del projecte, però al final també s'ha decidit incloure una vista Avís legal en l'aplicació Android.

## **Annex 3: Instal·lació d'un entorn de desenvolupament LAMP**

En aquest apartat es descriu, sense entrar massa en detall, el procés que he seguit per muntar un servidor web de desenvolupament, i poder, d'aquesta forma, programar l'aplicació web i el servei web d'una manera molt més àgil i ràpida. A més, el fet de tenir un servidor de desenvolupament amb característiques similars a les d'un servidor de producció, permet preveure problemes quan arriba el moment de publicar el web en un entorn real.

Per tal de fer-ho més entenedor se senyalitzarà d'ara endavant amb les següents imatges:



les comandes executades des de la consola,



els continguts dels fitxers editats.

LAMP és l'acrònim utilitzat per descriure els servidors amb les següents característiques:

- Sistema operatiu GNU/Linux,
- Servidor de pàgines web Apache,
- Gestor de bases de dades MySQL,
- Perl, PHP o Python com a llenguatge per programar pàgines web dinàmiques.

No obstant, i per tenir un servidor encara més semblant a un real, s'ha decidit instal·lar els següents programes addicionals:

- Eina phpMyAdmin per gestionar les bases de dades,
- Programa SSMTP per lliurar correus electrònics per SMTP des d'un servidor local.

### **Observacions:**

1) L'eina SSMTP s'ha utilitzat per comprovar el correcte funcionament de la vista Enviar missatge (la qual envia missatges per correu electrònic), sense tenir la necessitat d'instal·lar un servidor de correu propi.

2) El sistema operatiu utilitzat és la distribució GNU/Linux Mint 14 64-bit i totes les comandes s'han executat des d'un usuari amb perfil administrador (*root*).

## Instal·lació del servidor HTTP Apache i del mòdul PHP5

```
~ $> aptitude install apache2 php5 libapache2-mod-php5  
~ $>
```

Per comprovar si s'ha instal·lat correctament es pot obrir un navegador i anar a l'adreça <http://localhost/> per veure si el servei està funcionant.

## Instal·lació del gestor de bases de dades MySQL i de l'eina phpMyAdmin

```
~ $> aptitude install mysql-server libapache2-mod-auth-mysql  
...  
~ $> aptitude install php5-mysql phpmyadmin  
~ $>
```

Per comprovar si s'ha instal·lat correctament es pot obrir un navegador i anar a l'adreça <http://localhost/phpmyadmin>.

Per defecte el servidor Apache allotja les pàgines web dins el directori `/var/www`. Per poder treballar d'una forma més còmoda des de la *home* de l'usuari, i no estar pendent de nivells d'accessos i permisos, cal fer les modificacions següents:

- 1) Editar el fitxer `/etc/apache2/mods-available/userdir.conf` i substituir la línia `AllowOverride FileInfo AuthConfig Limit Indexes`

per:

`AllowOverride All`

```
<IfModule mod_userdir.c>  
    UserDir public_html  
    UserDir disabled root  
  
    <Directory /home/*/public_html>  
        # AllowOverride FileInfo AuthConfig Limit Indexes  
        AllowOverride All  
        Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec  
        <Limit GET POST OPTIONS>  
            Order allow,deny  
            Allow from all  
        </Limit>  
        <LimitExcept GET POST OPTIONS>  
            Order deny,allow  
            Deny from all  
        </LimitExcept>  
    </Directory>  
</IfModule>
```

**Fitxer userdir.conf**

2) Editar el fitxer `/etc/apache2/mods-available/php5.conf` i comentar tota la part que carrega el mòdul `mod_userdir.c` del final:

```
<IfModule mod_php5.c>
  <FilesMatch ".+\.(ph(p[345]?|t|tml)$">
    SetHandler application/x-httpd-php
  </FilesMatch>
  <FilesMatch ".+\.(phps)$">
    SetHandler application/x-httpd-php-source
    # Deny access to raw php sources by default
    # To re-enable it's recommended to enable access to the files
    # only in specific virtual host or directory
    Order Deny,Allow
    Deny from all
  </FilesMatch>
  # Deny access to files without filename (e.g. '.php')
  <FilesMatch "^\.ph(p[345]?|t|tml|ps)$">
    Order Deny,Allow
    Deny from all
  </FilesMatch>

  # Running PHP scripts in user directories is disabled by default
  #
  # To re-enable PHP in user directories comment the following lines
  # (from <IfModule ...> to </IfModule>.) Do NOT set it to On as it
  # prevents .htaccess files from disabling it.
  #<IfModule mod_userdir.c>
  #   <Directory /home/*/public_html>
  #     php_admin_value engine Off
  #   </Directory>
  #</IfModule>
</IfModule>
```

#### Fitxer `php5.conf`

3) Editar el fitxer `/etc/apache2/envvars` i substituir l'usuari i grup `www-data` pel nostre:

```
...
# Since there is no sane way to get the parsed apache2 config in scripts, some
# settings are defined via environment variables and then used in apache2ctl,
# /etc/init.d/apache2, /etc/logrotate.d/apache2, etc.

export APACHE_RUN_USER=usuari
export APACHE_RUN_GROUP=usuari
...
```

#### Fitxer `envvars`

### Instal·lació del programa SSMTP

```
~ $> aptitude install ssmtp mailutils
~ $>
```

Una vegada instal·lat s'ha d'editar el fitxer `/etc/ssmtp/ssmtp.conf` per indicar-li al programa una configuració SMTP d'un compte de correu vàlid (com per exemple, un compte de Gmail):

```
#
# Config file for sSMTP sendmail
#
# The person who gets all mail for userids < 1000
# Make this empty to disable rewriting.
root=el-meu-compte@gmail.com

# The place where the mail goes. The actual machine name is required no
# MX records are consulted. Commonly mailhosts are named mail.domain.com
mailhub=smtp.gmail.com:587

# Where will the mail seem to come from?
rewriteDomain=gmail.com

# The full hostname
hostname=el-meu-compte@gmail.com

# Use SSL/TLS before starting negotiation
UseTLS=Yes
UseSTARTTLS=Yes

# Username/Password
AuthUser=el-meu-compte@gmail.com
AuthPass=la-meva-contrasenya

# Are users allowed to set their own From: address?
# YES - Allow the user to specify their own From: address
# NO - Use the system generated From: address
FromLineOverride=YES
```

#### **Fitxer ssmtp.conf configurat amb un compte de Gmail**

Acte seguit s'ha d'editar el fitxer `/etc/php5/apache2/php.ini` per indicar que els enviaments de correus electrònics es facin mitjançant el programa SSMTP:

```
...
; For Unix only. You may supply arguments as well (default: "sendmail -t -i").
; http://php.net/sendmail-path
sendmail_path = /usr/sbin/ssmtp -t -i
...
```

#### **Fitxer php.ini**

Arribat a aquest punt el servidor de desenvolupament ja té tots els serveis necessaris. Ara només cal donar d'alta un nom de domini local i indicar quina serà la carpeta que contindrà l'aplicació web:

1) Crear la carpeta del projecte (per exemple, 'meteo'):

```
~ $> mkdir /home/usuari/meteo  
~ $>
```

2) Crear l'arxiu *virtualhost* del projecte:

```
~ $> cd /etc/apache2/sites-available  
~ $> cp default meteo.local  
~ $>
```


3) Editar el fitxer creat /etc/apache2/sites-available/meteo.local per indicar la configuració del nom de domini local:

```
<VirtualHost *:80>  
    ServerAdmin webmaster@localhost  
    ServerName meteo.local  
    ServerAlias www.meteo.local  
    DocumentRoot /home/usuari/meteo  
    <Directory />  
        Options FollowSymLinks  
        # AllowOverride None  
        AllowOverride All  
    </Directory>  
    <Directory /home/usuari/meteo>  
        Options Indexes FollowSymLinks MultiViews  
        AllowOverride None  
        Order allow,deny  
        allow from all  
    </Directory>  
  
    ErrorLog ${APACHE_LOG_DIR}/meteo-error.log  
  
    # Possible values include: debug, info, notice, warn, error, crit,  
    # alert, emerg.  
    LogLevel warn  
  
    CustomLog ${APACHE_LOG_DIR}/access.log combined  
</VirtualHost>
```

**Fitxer virtualhost meteo.local**

4) Habilitar el nou *virtualhost* i reiniciar el servidor Apache per activar els canvis:

```
~ $> a2ensite meteo.local  
~ $> service apache2 reload  
~ $>
```



En el cas d'haver completat tots els passos correctament, es podrà consultar des d'un navegador el projecte emmagatzemat en la carpeta `/home/usuari/meteo`, des de l'adreça local <http://meteo.local/>.



## **Annex 4: Programari wview**

Fins ara només s'ha parlat de les aplicacions que s'han desenvolupat, els llenguatges de programació que s'han fet servir, les llibreries utilitzades, els patrons de disseny, l'entorn de desenvolupament que s'ha instal·lat, etc. Però en cap moment s'ha mencionat quin és el funcionament i de com es recuperen les dades de l'estació meteorològica. És per aquest motiu que considero important dedicar un apartat en aquesta memòria, per explicar quin és el sistema que s'ha utilitzat per recuperar les dades meteorològiques de l'estació, ja que és el pilar central del present projecte.



<http://wviewweather.com/>

Les estacions meteorològiques d'Oregon Scientific que es connecten per USB als ordinadors, utilitzen una combinació del protocol USB-HID (USB - Human Interface Device), i d'un altre protocol propietat del mateix fabricant per transmetre les dades entre l'estació i l'ordinador. Això fa que només es pugui consultar i recuperar les dades de l'estació utilitzant un programari privatiu d'Oregon Scientific.

No obstant, i pel que he pogut esbrinar consultant diversos fòrums i pàgines web especialitzades, gràcies a l'enginyeria inversa i a l'esforç d'alguns usuaris molt voluntariosos, s'ha pogut desxifrar el protocol d'algunes d'aquestes estacions i deduir com es transfereixen les dades cap a l'ordinador. Això ha fet possible l'aparició de diversos *data loggers* i altres programaris alternatius al del fabricant, els quals permeten recuperar les dades de les estacions. D'entre aquests programes alternatius el que he decidit utilitzar és el programari wview, el qual està disponible sota una llicència lliure GPL versió 2.

El programari per a sistemes UNIX wview és una col·lecció de programes i dimonis (un dimoni és un programa informàtic que treballa en segon terme i que normalment s'inicia com a procés), que permeten recuperar les dades i condicions meteorològiques actuals d'una estació meteorològica, així com el seu històric de dades en el cas en que l'estació també arxivi les dades.

Suporta diversos models d'estacions de diferents fabricants, com per exemple estacions de Texas Weather Instruments, Davis, La Crosse i Oregon Scientific.

Com a característiques principals, es pot destacar:

- Emmagatzematge de les de l'estació en una base de dades relacional SQLite3,
- Generació d'informes, resums i gràfics en format HTML,
- Exportació remota de les dades cap a bases de dades MySQL o PostgreSQL,
- Configuració accessible mitjançant la línia de comandes o una interfície web,
- Generació d'alarmes i avisos mitjançant connexions TCP amb *sockets*.

En distribucions basades en el sistema Ubuntu la seva instal·lació és molt senzilla. Només cal afegir els següents repositoris (editant, per exemple, el fitxer `/etc/apt/sources.list`):

```
...  
deb http://www.wviewweather.com/apt/precise precise main  
deb-src http://www.wviewweather.com/apt/precise precise main  
...
```



#### Fitxer `sources.list`

i instal·lar el paquet `debian` mitjançant la següent comanda:

```
~ $> aptitude update  
...  
~ $> aptitude install wview  
~ $>
```



Si res ha sortit malament es podrà accedir a la configuració del programari utilitzant la interfície web des de l'adreça <http://localhost/wviewmgmt/login.php>.

Apareixerà llavors una diàleg demanant la contrasenya d'administració (que per defecte és "wview"), i una vegada introduïda la contrasenya es podrà accedir a la gestió del programa.

**Please enter the admin password:**

D'entre les seves múltiples opcions, les utilitzades per a l'execució d'aquest projecte han estat:

- 1) Pàgina per consultar l'estat del sistema (per comprovar si els dimonis s'estan executant correctament, per determinar si hi ha algun tipus d'error, etc.),
- 2) Pàgina on indicar els detalls de la connexió de la base de dades del servidor de producció, per així poder sincronitzar les dades meteorològiques del servidor local (SQLite3) amb les de l'entorn real (MySQL),
- 3) Pàgina on indicar el model d'estació meteorològica que està connectada per USB.

Service	Status	Message	Stat1	Value	Stat2	Value	Stat3	Value	Stat4	Value
wview	Running	Normal operation	LOOP packets received	79108	Archive packets generated	15820				
html	Running	Normal operation	Images defined	52	Templates defined	14	Images generated	1572032	Templates generated	1107400
alarms	Not Started									
cwop	Not Started									
http	Not Started									
ftp	Not Started									
ssh	Not Started									
pmon	Running	Normal operation	Processes restarted	0						

Pàgina de l'estat del sistema amb l'estat dels diferents dimonis

Pàgina on especificar les dades per a l'exportació remota

The simulator has no station interface. The WMRUSBWH1080 /TE923 weather stations only support a native USB interface. No interface configuration is required.

Configure storm criteria.

Preset accumulated values if data was missed

### Pàgina on especificar el model d'estació connectada per USB

Però per a que el procés de sincronització pugui completar-se amb èxit cal fer encara un parell de passos més:

#### 1) Autoritzar connexions remotes al gestor MySQL de l'entorn de producció

Per motius de seguretat els allotjaments web no accepten connexions remotes al gestor MySQL (només accepten connexions de forma local). És per aquesta raó que cal autoritzar les connexions remotes al MySQL, de tal forma que el servidor local es pugui connectar al servidor real i actualitzar així la base de dades de l'entorn de producció.

Això no obstant, com que autoritzar connexions remotes pot generar problemes de seguretat, també és important revisar i actualitzar les regles del tallafocs per evitar que qualsevol adreça es pugui connectar al servidor real i comprometre les dades d'alguna manera.

## 2) Sincronitzar les bases de dades de forma periòdica

Finalment cal programar un automatisme per sincronitzar l'exportació de les dades de la base de dades SQLite3 cap a la base de dades del MySQL. D'aquesta manera les dades meteorològiques de l'entorn de producció s'actualitzaran amb una certa freqüència.

Per sort el programari wview incorpora un altre programa que s'encarrega de fer aquesta sincronització: el programa wview-mysql-export. Aquest recupera les dades de connexió remota introduïdes en la pàgina on s'especifiquen les dades d'exportació, i cada vegada que s'executi se sincronitzaran les dues bases de dades.

L'únic que queda per fer és afegir l'execució del programa a l'administrador de processos *cron* per a que s'executi cada cert temps.

Per fer-ho cal introduir la següent comanda des de la consola del servidor local on està connectada l'estació:

```
~ $> crontab -e  
~ $>
```

i afegir la corresponent entrada per tal que s'executi el programa wview-mysql-export:

```
...  
# wview - exportació de dades  
0 */1 * * * /usr/bin/wview-mysql-export  
  
# wview - fer tweet amb les dades meteorològiques actuals  
02 0,4,8,12,16,20 * * * /usr/bin/python /home/usuari/meteo/twitter.py  
...
```

### **Entrades de l'administrador de processos *cron***

#### **Observacions:**

1) En la primera entrada del *cron* es pot veure com el programa wview-mysql-export està programat per a que s'executi una vegada cada hora en punt.

2) La segona entrada del *cron* correspon a l'execució de l'script en python que s'encarrega de fer els tweets de les dades actuals a la xarxa social Twitter. Aquest s'executa cada quatre hores tal i com s'ha comentat anteriorment (a les 0:02h, a les 4:02h, a les 8:02h a les 12:02h, a les 16:02h i a les 20:02h).

## **Annex 5: Comprovació del sistema implementat**

Durant el transcurs d'aquest treball s'ha desenvolupat un conjunt de quatre aplicacions diferents, de les quals s'ha explicat des del disseny inicial fins a la seva implementació, amb més o menys detall. Tanmateix, crec que seria interessant acabar d'explicar quin és l'estat de cadascuna de les aplicacions, així com descriure com accedir a les que estan funcionant en l'actualitat de forma real.

### **1 - Script de publicació de les dades actuals a la xarxa social Twitter**

Tal i com s'ha comentat aquesta aplicació és un petit script programat en llenguatge Python que, mitjançant una entrada a l'administrador de processos *cron*, s'executa cada quatre hores per publicar les dades actuals de temperatura, humitat i pressió a la xarxa social Twitter.

Està instal·lat en el servidor local on està connectada l'estació i funciona 24/7 (24 hores al dia, 7 dies a la setmana).

El nom d'usuari de la xarxa Twitter on es publiquen els tweets és *@meteoandorra* i el seu *timeline* es pot consultar anant a l'adreça <http://twitter.com/meteoandorra>.

### **2 - Aplicació web**

Aquesta aplicació permet consultar les dades actuals de l'estació, les dades meteorològiques per un dia en concret i, les mitjanes, mínims i màxims dels mesos d'un any determinat. Està programada en llenguatge PHP juntament amb l'eina de treball Zend Framework, i les dades estan emmagatzemades en una base de dades MySQL.

L'aplicació està instal·lada en un servidor web real virtualitzat de tipus LAMP (Linux, Apache, MySQL i PHP).

L'adreça web de l'aplicació desenvolupada des d'on es pot consultar l'arxiu històric de les dades meteorològiques és <http://www.meteoandorra.com/>.

### **3 - Servei web**

El servei web desenvolupat utilitza l'especificació SOAP com a protocol per enviar missatges cap als clients. També s'ha programat en llenguatge PHP i l'eina de treball Zend Framework.

El servei web està instal·lat en el mateix servidor real on està instal·lada l'aplicació web.

Per tal que els clients puguin localitzar els mètodes del servei web s'ha creat un contracte WSDL, el qual és accessible des de l'URL: [www.meteoandorra.com/soap/webservice.php?wsdl](http://www.meteoandorra.com/soap/webservice.php?wsdl).

Es pot comprovar el seu funcionament utilitzant un client molt lleuger i senzill realitzat en llenguatge Python (la funcionalitat d'aquest client senzill s'explicava en la pàgina 11). Per fer-ho cal tenir instal·lat l'interpret de llenguatge Python i les llibreries python-soappy i python-suds.

La següent comanda mostra l'execució i el resultat de dit script:

```
~ $> python /home/usuari/meteo/soap.py
Dades Actuals
=====
Resultat (cadena JSON):
{"dateTime":"1389524100","barometer":917,"outTemp":8.1,"outHumidity":54}
Temperatura: 8.1 °C
Humitat: 54 %
Pressió : 917 mbar

Desum Diari
=====
Resultat (cadena JSON):
{"tempMin":6.8,"tempMinTimestamp":"09:35h","tempMax":10,"tempMaxTimestamp"
:"01:05h","humMin":54,"humMinTimestamp":"11:50h","humMax":58,"humMaxTimest
amp":"02:05h","pressMin":916,"pressMinTimestamp":"01:05h","pressMax":918,"
pressMaxTimestamp":"09:35h"}
...
```

**Observació:** Com s'ha comentat, el resultat del servei web és una cadena en format JSON. Aquesta s'ha de mapejar posteriorment i convertir-la en una estructura de dades per poder mostrar el resultat correctament.

#### **4 - Aplicació Android**

Malauradament l'aplicació per a sistemes Android no està publicada en el *Play Store* de Google, ja que pel moment les empreses i persones ubicades al Principat d'Andorra no tenen autorització per fer-ho (hi ha una llista de països autoritzats i Andorra no hi està inclòs).

Teòricament l'aplicació es pot provar mitjançant un emulador que ve inclòs amb l'Android SDK, no obstant, no s'ha aconseguit fer-lo funcionar correctament. És per aquest motiu que el procés que s'explicarà a continuació detalla com executar l'aplicació en un dispositiu físic real.

**Observació:** Cal tenir en compte que el desenvolupament s'ha realitzat en un dispositiu amb una pantalla de 7 polzades. L'execució d'aquesta aplicació en dispositius amb una pantalla inferior pot suposar una incorrecta visualització.

## 1) Descarregar i executar l'Android SDK

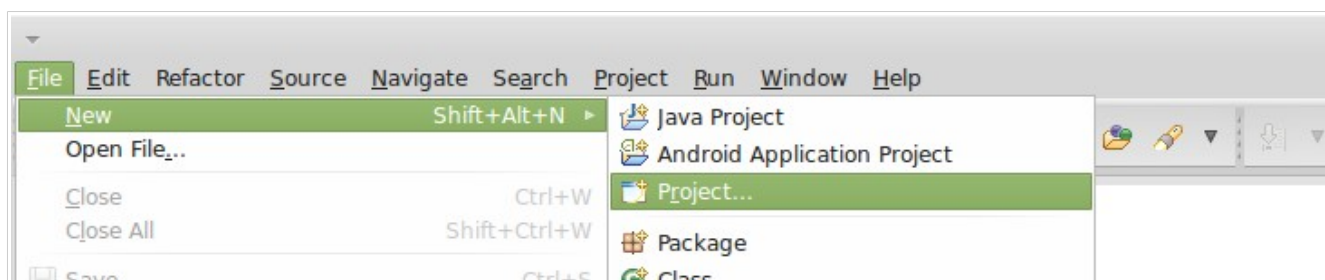
- El primer que s'ha de fer és anar al web oficial des d'on es pot descarregar l'Android SDK: <http://developer.android.com/sdk/index.html>.

- Una vegada descarregat, es descomprimeix l'arxiu i es fa clic sobre el llançador de l'Eclipse que es troba dins la carpeta `eclipse` de l'SDK. Això obrirà l'Eclipse - Android Developer Tools.

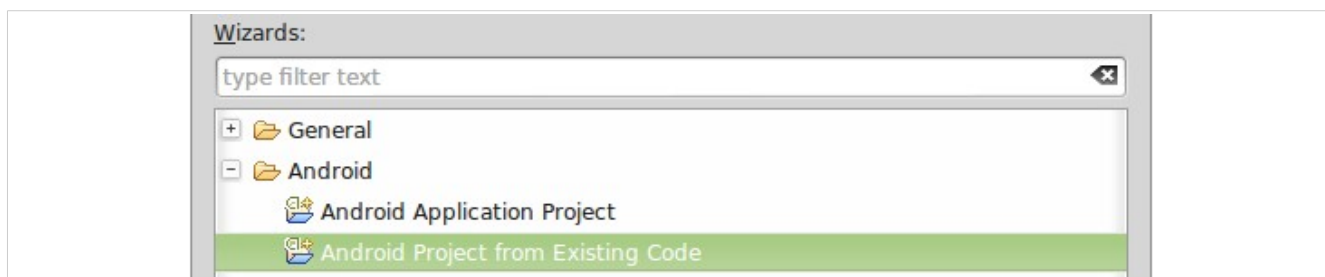
- Apareixerà un diàleg on es demana escollir el *workspace* (directori de treball) amb el que es voldrà treballar. S'escull el directori corresponent i ja es té l'Eclipse - ADT a punt per començar a treballar.

## 2) Crear el projecte a partir del codi font

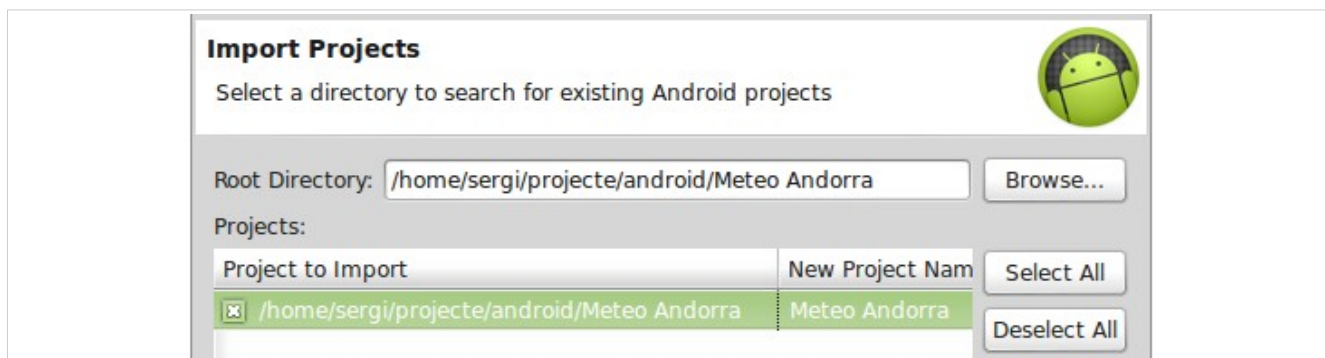
- Des de l'Eclipse crear el projecte anant a **File** » **New** » **Project**,



- Seleccionar l'opció **Android** » **Android Project from Existing Code** i fer clic a **Next**,



- Seleccionar la carpeta **Meteo Andorra** del projecte Android i fer clic a **Finish**,





### 3) Canviar la configuració del dispositiu Android

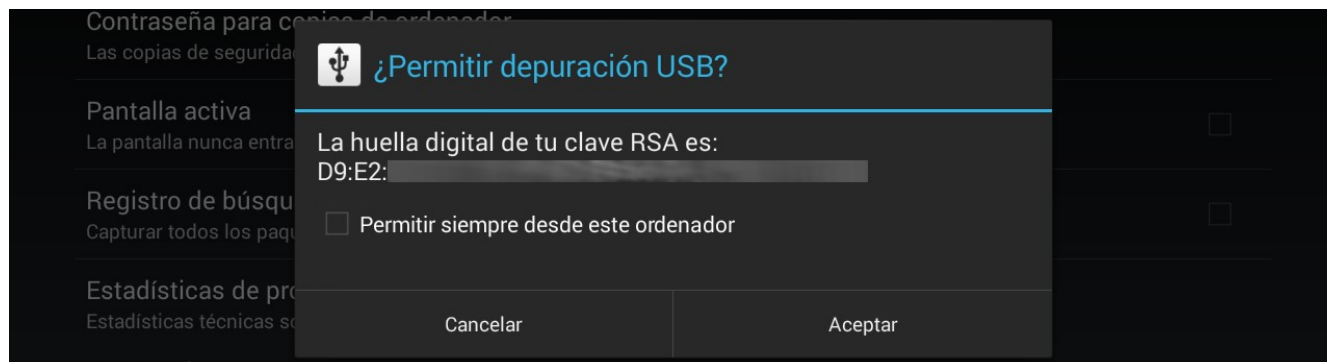
- Connectar per USB el dispositiu Android a l'ordinador,
- Des del dispositiu Android anar a Ajustos i activar les Opcions de desenvolupament,



- Dins les Opcions de desenvolupament activar també la Depuració USB,

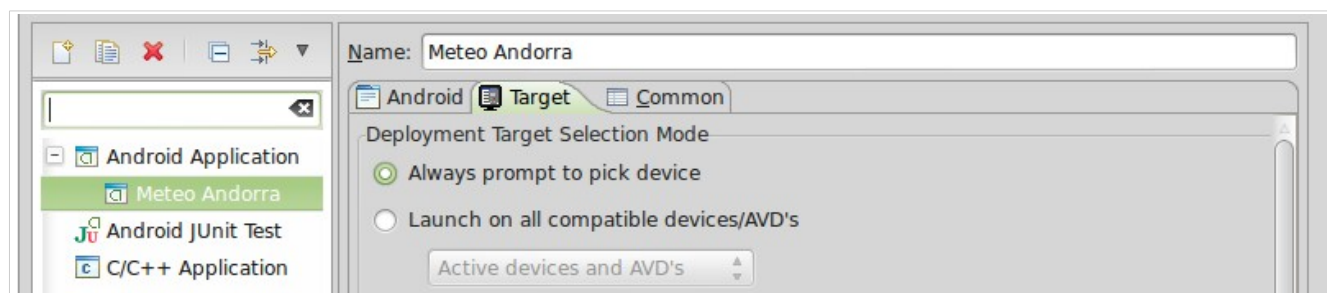


- Autoritzar la connexió des de l'ordinador connectat,

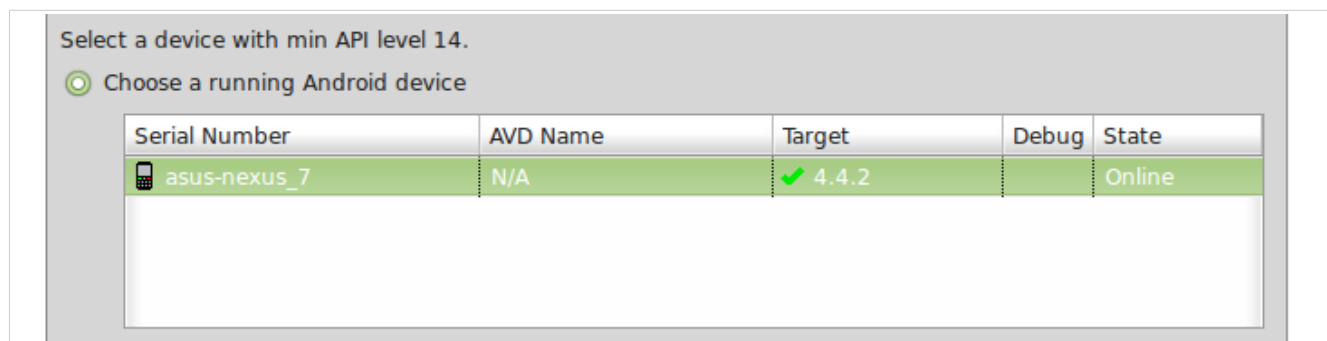


#### 4) Fer que l'aplicació s'instal·li al dispositiu Android

- Des de l'Eclipse anar a Run » Run configurations, i dins el Target seleccionar l'opció Always prompt to pick device,



- Executar l'aplicació anant a Run » Run, i clicar a Ok.



Si s'han realitzat tots els passos correctament l'aplicació s'instal·larà al dispositiu i podrà ser executada. Si no, es pot instal·lar manualment des dels següents enllaços (caldrà dir-li al dispositiu que permeti la instal·lació d'aplicacions que no siguin del *Play Store* de Google):

- Per a Android 4.3: <http://www.meteoandorra.com/android/MeteoAndorra-43.apk>
- Per a Android 4.4: <http://www.meteoandorra.com/android/MeteoAndorra-44.apk>

## **Annex 6: Valoració econòmica**

Per finalitzar aquest treball de fi de carrera es realitzarà una valoració econòmica, per tal d'avaluar la inversió inicial que ha calgut fer, així com les despeses que suposaria tenir el sistema implementat en funcionament.

### **1 - Inversió inicial realitzada**

<b>Descripció</b>	<b>Preu</b>
Estació meteorològica Oregon Scientific WMR88	95 €
Tablet ASUS Nexus 7 - 16GB	189.90 €
Registre domini meteoandorra.com	13,95 €
Servidor web virtualitzat 10GB	199 €
	<b>TOTAL: 497,85 €</b>

### **2 - Cost anual**

<b>Descripció</b>	<b>Preu</b>
Renovació domini meteoandorra.com	13,95 €/any
Servidor web virtualitzat 10GB	199 €/any
	<b>TOTAL: 212,95 €/any</b>

En la inversió inicial no s'ha inclòs l'equip de desenvolupament (el portàtil Apple MacBook) ni el servidor local, perquè aquests equips van ser adquirits prèviament per a d'altres finalitats i el seu cost estava amortitzat des de feia temps.

Tampoc s'ha tingut en compte les hores de programació ni les dedicades a la implantació del sistema.

Com es pot observar ni el cost inicial, ni el cost anual (que surt a poc més de 17€ al mes) és excessivament elevat. També s'ha de dir que el web [www.meteoandorra.com](http://www.meteoandorra.com) no fa gaire que està funcionant, i per tant, encara és aviat per decidir si el seu cost és assumible o no pel nombre de visites que rep. Estaria bé seguir l'evolució durant almenys un any per determinar si compensa tenir el web en funcionament, o si pel contrari és millor tancar-lo i centrar-se exclusivament a les xarxes socials.