

TREBALL DE RECERCA

# Aplicació per a la captura i visualització de rutes

2on Batxillerat  
TECNOLOGIA

Institut l'Alt Berguedà



## AGRAÏMENTS

Al finalitzar un projecte tan dur i ple de dificultats com el desenvolupat en aquest treball de recerca, és inevitable sentir-se orgullós del mèrit personal. Tot i això, aquest projecte no hagués estat possible sense la participació de persones que han facilitat les coses per tal de que el treball arribés a un final exitós. Per aquest motiu, em plau utilitzar aquest espai per ser justa i conseqüent, expressant a aquestes persones els meus més sincers agraïments.

En primer lloc, vull agrair de manera especial a l'enginyer de telecomunicacions Ferran Candela Farré per ajudar a estructurar les comunicacions i interaccions internes del projecte i també per ajudar a solucionar els problemes que ha anat presentat l'aplicació durant tota la seva implementació. Pel seu suport i confiança, ja que sense la seva ajuda el treball no hauria estat mai possible.

Vull també expressar un sincer agraïment a la meva família i amics, pels seus ànims constants, i en especial a la meva estimada germana Noelia, enginyera de telecomunicacions, que gràcies a la seva experiència, ha ajudat a escollir un treball creatiu i alhora a la mida d'un estudiant de Batxillerat, per la seva supervisió i suport a l'hora d'entendre molts conceptes que no haguessin estat possibles sense el recolzament d'una persona especialitzada. Li agraeixo també que hagi aportat idees que han enriquit el treball en gran mesura.

Per últim, donar les gràcies al tutor del treball, pel seu suport i constància, i per la seva ajuda sempre que ha sigut necessària.

# ÍNDEX

1.- INTRODUCCIÓ.....	3
2.- OBJECTIUS .....	5
3.- PLANIFICACIÓ .....	6
3.1.- Definició de les tasques.....	6
3.2.- Diagrama de Gantt .....	7
4.- CONCEPTES TEÒRICS .....	8
4.1.- Llenguatges de programació.....	8
4.1.1.- Java.....	8
4.1.2.- HTML i CSS .....	10
4.1.3.- JavaScript.....	11
4.1.4.- Android .....	12
4.2.- Conceptes bàsics de programació .....	13
4.2.1.- Tipus de dades .....	13
4.2.2.- Tipus d'estructures .....	14
4.2.3.- Operadors .....	15
4.2.4.- Modificadors d'accés .....	16
4.2.5.- Objecte .....	16
4.2.6.- Classe.....	16
4.2.7.- Atribut .....	16
4.2.8.- Mètode .....	16
4.2.9.- Debbuger.....	17
4.3.- Conceptes relacionats amb el projecte.....	17
4.4.- Les coordenades GPS.....	21
5.- PROJECTE.....	22
5.1.- Disseny.....	22
5.1.1.- Descripció del comportament .....	22
5.1.2.- Esbós Pàgina Web .....	24
5.1.3.- Esbós programa Android.....	25
5.2.- Implementació.....	27
5.2.1.- Aplicació Web.....	27
5.2.1.4.- Hibernate i Base de dades.....	39
5.2.1.5.- Interacció amb Google Maps .....	40
5.2.2.- Aplicació Android.....	43

5.2.2.1.- Definició del funcionament.....	43
5.2.2.2.- Format del fitxer.....	44
5.3.- Verificació.....	45
5.3.1.- Ruta de prova - Fitxer de dades manual.....	46
5.3.2.- Ruta real - La primera ruta.....	47
6.- CONCLUSIONS.....	48
6.1.- Reflexió del projecte .....	48
6.2.- Línies de continuïtat.....	48
7.- BIBLIOGRAFIA .....	50
8.- ANNEXES	

## 1.- INTRODUCCIÓ

### *Introducció àmbit de treball*

Avui dia, es viu submergit en un món que demana solucions constants a qualsevol problema, dificultat o facilitat que pugui presentar-se amb la finalitat de facilitar a les persones, una millor qualitat de vida. Per aconseguir aquest propòsit doncs, durant molts anys, la tecnologia ha anat i segueix creant millores tals com aparells electrònics, controls automàtics o bé programes informàtics que amb una ràpida assimilació social, han passat a formar part habitual del dia a dia de les persones.

Aquest treball es centrarà en un dels àmbits compresos dins el món de la tecnologia; la programació i comunicació. Quan en tecnologia es parla del concepte *programació*, es fa referència a un procés l'objectiu del qual és crear un programa que mostri un determinat comportament que el mateix programador ha escollit. Per a crear aquest programa i que al mateix temps l'aparell l'interpreti i n'executi les instruccions que se li indiquen, necessàriament ha d'haver estat programat amb un determinat *llenguatge de programació*, idioma informàtic artificial dissenyat amb la única finalitat de ser l'intermediari entre el llenguatge humà i el de l'aparell a programar (ordinador, mòbil o qualsevol màquina).

Aquesta tècnica doncs, s'ha dut a terme aplicada a dos aparells pioners en el món de la tecnologia; l'ordinador i el mòbil, els quals permetran a l'usuari gaudir d'aquest treball que el programador ha realitzat

### *En què consisteix?*

El treball següent, per tant, consisteix en la creació d'una pàgina web que permetrà visualitzar rutes sobre un mapa, tot utilitzant les prestacions que ofereix *Google Maps*, i que, a més, proporcionarà dades de cada un dels punts de la ruta escollida. L'aparell que permetrà capturar les dades de la ruta és el mòbil, en concret un mòbil amb Sistema Operatiu *Android*.

En conclusió, es crearà un programa compost; el mòbil agafarà les dades de la ruta, les emmagatzemarà, i al ser descarregades a l'ordinador, podran ser visualitzades mitjançant la Pàgina Web.

### *Motivacions*

Referent a l'elaboració i a les motivacions que van impulsar a la realització d'un treball sobre aquest tema, pot dir-se que el major al·licient va ser el gran interès en la programació ja que com és sabut, tothom utilitza aparells programats sense ser conscients del gran esforç que ha suposat per a

la persona que ha fet el treball primitiu. Un esforç que s'ha pogut comprovar en primera persona i que ha portat a una satisfacció personal molt positiva. La inserció en un tema totalment nou i del qual no se'n tenen coneixements previs ha sigut també una motivació i un repte personal.

### *Problemes i dificultats*

Abans d'arribar a la meta proposada però, s'han trobat algunes dificultats que no han deixat de sorgir durant tota la realització del treball. Algunes d'aquestes dificultats han estat degudes a que la simple manca d'un sol caràcter durant l'escriptura o de la confusió d'una lletra majúscula per minúscula durant la creació del codi, porta directament al mal funcionament de tot el conjunt del programa. També han sorgit moments de dificultat durant els quals no es trobaven maneres per fer entendre al programa el desig del programador, però que s'han pogut solucionar gràcies a que hi ha diversitat de camins per arribar al mateix objectiu. Altres problemes han sorgit per la insuficiència de coneixements en aquest ampli món, però que gràcies a l'extensa informació que es pot trobar sobre el tema i a l'ajuda de persones especialitzades s'han pogut solucionar amb èxit.

### *Estructura del treball*

Seguidament, es farà un breu resum del que es troba dins aquesta memòria per tal de fer més entenedora la continuïtat duta a terme.

En primer lloc, s'exposen els objectius i la planificació del projecte que serviran per situar al lector. Darrerament, hi ha diversos conceptes teòrics els quals són molt necessaris per entendre la part pràctica que s'ha realitzat. D'aquests conceptes se n'explicarà la part que té més relació amb el projecte i no la seva totalitat, sent aquesta tant àmplia que sortiria del marc del treball.

Un cop finalitzada aquesta part, es troba la secció més important del treball; l'apartat d'explicació del projecte. En aquest, hi ha tota la informació de la part pràctica i del projecte pròpiament dit, la qual està explicada detalladament.

Per acabar, es troben les conclusions, juntament amb una reflexió sobre les millores que es poden afegir al treball, sent aquestes, línies de continuïtat per a projeccions futures.

A part de la memòria, hi ha adjuntades als annexes cada una de les línies de codi que han estat necessàries per al funcionament complet del programa. Aquestes línies són el nucli del projecte i han estat comentades de manera que fan més entenedor el que s'està duent a terme.

## **2.- OBJECTIUS**

En aquest treball es troben dos objectius molt clars; la creació d'una pàgina Web i la d'una aplicació per Android. Aquests elements però, tenen un punt en comú que farà possible el funcionament complet de l'aplicació.

Un objectiu doncs, consisteix aconseguir crear una Pàgina Web que, com s'ha citat, a partir de la lectura d'un fitxer, sigui capaç d'interpretar les dades proporcionades i plasmar-les de manera entenedora per a l'usuari. En aquest cas, el fitxer contindrà informació sobre el posicionament GPS de rutes, i la pàgina Web haurà de plasmar-les sobre un mapa per tal que l'usuari les pugui entendre amb facilitat.

L'altre objectiu establert, és doncs la creació d'una aplicació per Android capaç d'emmagatzemar les posicions d'una ruta qualsevol en un fitxer i que alhora aquest fitxer compleixi els requeriments de la Pàgina Web, és a dir, un fitxer compatible amb el que la pàgina Web és capaç de llegir.

Aquests objectius s'intentaran aconseguir a través d'un programa funcional bàsic, que realitzi, en un principi, les funcions necessàries per al funcionament del programa. Si el temps ho permet, s'aplicaran millores tals com informació extra de la ruta o un disseny més elaborat.



### **3.- PLANIFICACIÓ**

En aquest apartat s'explicarà com s'ha estructurat el temps del projecte i quines són les tasques principals en les que s'ha dividit. A més es proporcionarà un calendari que detalla el temps emprat per a cada tasca.

#### **3.1.- Definició de les tasques**

##### 1. Estudi previ

Ha sigut necessari aprendre a programar en:

- Java
- JavaScript
- Android
- HTML+CSS

Per assolir aquests coneixements s'han utilitzat principalment llibres i tutorials. Aquests es poden consultar a l'apartat de Bibliografia.

##### 2. Disseny del contingut

Abans de realitzar el disseny definitiu de cada part dels programes, (Pàgina Web, Aplicació Android) es van dur a terme tres dissenys previs:

- Primer esbós – “fet a mà” (DIN-A4)
- Segon esbós – “Balsamiq Mockups” (programa de disseny)
- Tercer esbós – “Programmer's Notepad” (lector de text)

##### 3. Desenvolupament

- Pàgina Web
- Aplicació Android

##### 4. Execució del resultat

##### 5. Resultat i conclusions

##### 6. Redacció de la memòria

### 3.2.- Diagrama de Gantt

A continuació es presenta una planificació orientativa del projecte. S'ha dividit el treball en diferents tasques especificades a l'apartat anterior i que queden esquematitzades a continuació:

	Activity Name	Start	Finish
1	○ Planificació TR	5/2/2012	12/14/2012
2	☐ Estudi previ	5/2/2012	7/13/2012
3	· Aprenentatge Java	5/2/2012	6/1/2012
4	· Aprenentatge JavaScript	6/4/2012	6/29/2012
5	· Aprenentatge Android	7/2/2012	7/13/2012
6	☐ Disseny del contingut	7/16/2012	8/3/2012
7	· Aplicació Web	7/16/2012	7/25/2012
8	· Aplicació Android	7/26/2012	8/3/2012
9	☐ Desenvolupament	8/6/2012	10/26/2012
10	· Aplicació Web	8/6/2012	9/14/2012
11	· Aplicació Android	9/17/2012	10/26/2012
12	☐ Execució del resultat	10/29/2012	11/16/2012
13	☐ Resultats i conclusions	11/19/2012	11/30/2012
14	☐ Redacció memoria	12/3/2012	12/14/2012

FIGURA 1.1: Tasques Diagrama de Gantt

D'altra banda, s'adjunta un diagrama de Gantt de la mateixa planificació, on queden reflectides les tasques en funció del temps emprat:

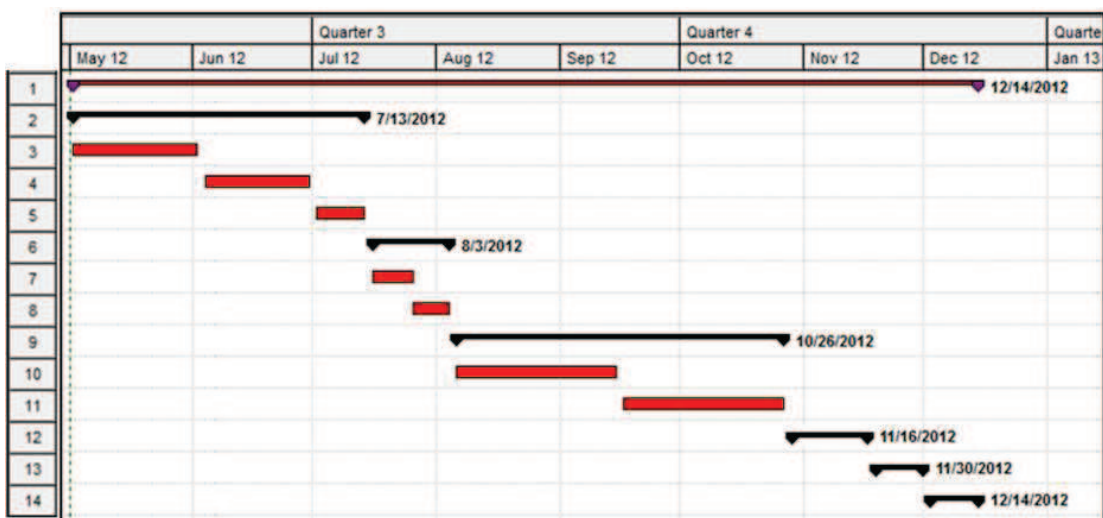


FIGURA 1.2: Diagrama de Gantt

## 4.- CONCEPTES TEÒRICS

Per entendre alguns dels passos de realització del projecte, és necessari tenir unes nocions bàsiques sobre els conceptes que s'utilitzaran durant la resta del treball.

En aquest apartat doncs, en primer lloc es farà una breu definició de cadascun dels llenguatges de programació que s'han fet servir i s'explicaran altres conceptes com els de les entitats que han intervingut en el traspàs de dades. D'altra banda, i ja entrant en coneixements pròpiament de programació, es dedicarà un apartat sencer a l'explicació de les eines bàsiques que s'han hagut d'estudiar per tal de ser capaços de dur a terme una implementació tal i com la que es presenta en aquest document.

### 4.1.- Llenguatges de programació

Tenint en compte que amb un sol llenguatge de programació no hauria sigut possible realitzar l'aplicació, a continuació s'explicarà quins són els llenguatges que s'han emprat i quina és la funció de cada un en aquest projecte.

#### 4.1.1.- Java

Java, és un dels llenguatges de programació més complets, segurs i utilitzats del moment. La seva característica principal és que tot programa en Java ha de ser *compilat*<sup>1</sup> i el codi que es genera *bytecodes*<sup>2</sup> s'interpreta per una màquina virtual. La Màquina Virtual Java (JVM) és l'entorn en el que s'executen els programes Java, i es pot definir essencialment com un ordinador abstracte, el qual està aïllat del *hardware*<sup>3</sup> de l'ordinador base.

Utilitza un tipus de programació orientada a objectes que intenta assemblar-se a la manera de pensar de l'home i no de la màquina. La utilització d'objectes doncs, permet que el codi sigui més entenedor per les persones.

A més a més, està millor organitzat que altres llenguatges de programació (C, C++...), i és fàcil d'estructurar, ja que per cada objecte d'una classe hi ha tot un seguit de característiques que el componen. Si no s'utilitzessin objectes, cada característica es trobaria dins del codi però no organitzada d'una manera concreta, és a dir, estaria desordenada. Així doncs, aquesta peculiaritat permet controlar el codi de manera més fàcil i fer-lo més entenedor.

- ❖ **Compilar**<sup>1</sup>: Procés de traducció d'un llenguatge de programació a un altre on el segon normalment és llenguatge màquina de manera que la màquina és capaç d'interpretar. Aquest procés el realitza el compilador, que és un programa informàtic dedicat a l'acció anteriorment descrita.
- ❖ **Bytecode**<sup>2</sup>: Codi intermedi entre el llenguatge de programació i el llenguatge màquina. Normalment tractat com a fitxer binari.
- ❖ **Hardware**<sup>3</sup>: Totes les parts tangibles que conformen un sistema informàtic. Components elèctrics, electrònics, mecànics i electromecànics.

Així doncs, l'objecte és la representació d'un concepte per a un programa, i conté tota la informació necessària per abastir el mateix concepte: les dades que descriuen el seu estat, les operacions que poden modificar aquest estat, etc.

En aquest projecte, Java ha sigut el llenguatge més utilitzat, ja que ha permès els intercanvis d'informació amb la base de dades, programar les funcions internes de la Pàgina Web, connexions entre els HTML existents, etc.

A continuació es presenta un exemple d'una part d'una classe escrita en Java. La resta del codi es pot consultar a l'Annex IX.

```
//Nova classe anomenada Ruta. Caracterització d'una ruta
public class Ruta {

    //Atributs: - Propietats/paràmetres que conté la classe Ruta
    private int _idRuta; //Nombre encarregat d'identificar la ruta amb la que
es treballa
    private String _NomRuta; // Paràmetre que contindrà el nom de la ruta
escollit per l'usuari
    private double _Longitud; // Paràmetre que contindrà la longitud total de
la ruta
    private Date _IniciTemps; // Paràmetre que contindrà la data de l'instant
en que va començar la ruta
    private Date _FinalTemps; // Paràmetre que contindrà el data de l'instant
en que va acabar la ruta
    private String _Comentaris; // Paràmetre que contindrà un comentari de la
ruta escollit per l'usuari
    private String _TempsTotal; // Paràmetre que contindrà el temps total en
el que s'ha efectuat la ruta
    private List<PosicioRuta> _Posicions; // Artibut format per una llista
d'objectes de la classe PosicioRuta

    //Mètodes: - Gets i Sets:
    public int get_idRuta() {
        return _idRuta; // Funció que permet obtenir el paràmetre
d'identificació de la ruta
    }
    public void set_idRuta(int _idRuta) {
        this._idRuta = _idRuta; // Funció que permet guardar el paràmetre
d'identificació de la ruta
    }
    public String get_NomRuta() {
        return _NomRuta; // Funció que permet obtenir el nom de la ruta
    }
    public void set_NomRuta(String _NomRuta) {
        this._NomRuta = _NomRuta; // Funció que permet guardar el nom de la
ruta
    }
    ...
}
```

FIGURA 2.1: Ruta.java

El codi que es presenta és un exemple parcial de la classe Ruta implementada en el projecte. Aquesta classe conté els seus paràmetres o atributs tals com `_idRuta`, `_NomRuta`... A més a més, conté totes les funcions que permeten manipular aquests paràmetres, com ara els gets i sets. Els gets permeten obtenir el valor del paràmetre al que fa referència, en canvi, el set permet guardar a la variable corresponent el valor que sigui necessari.

A la figura 2.1, es crea una nova classe anomenada Ruta, formada per vuit atributs (proprietats o paràmetres) que la caracteritzen. Un objecte d'aquesta classe contindria cada un d'aquests atributs com a característiques.

#### 4.1.2.- HTML i CSS

L'HTML (*HiperText Markup Language*) és un llenguatge que s'utilitza per establir l'estructura i contingut d'una pàgina web. Els arxius amb aquest perfil utilitzen una extensió `.htm` o `.html`.

Aquest ha estat emprat per donar estructura a la Pàgina Web, és a dir, cada un dels elements que poden ser visualitzats estan escrits en aquest llenguatge. Així doncs, cada una de les diferents pàgines creades té el seu propi codi HTML.

El codi HTML es troba en el format següent:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 //EN">
<html>
  <head>
    <title>Títol de la pàgina</title>
  </head>
  <body>
    <!-- Contingut principal -->
  </body>
</html>
```

FIGURA 2.2: Format bàsic HTML

Aquí es troba una pàgina Web amb un títol adjuntat anomenat "Títol de la pàgina" però sense cap altre element ja que el contingut principal està buit.

El contingut principal s'escriuria dins de *body* – Subtítols, botons, fotografies, taules ... –.

El CSS (*Cascading Style Sheets*), a diferència de l'HTML, és un llenguatge utilitzat per definir la presentació d'un document escrit en HTML o XML<sup>4</sup>. L'objectiu bàsic d'aquest és separar l'estructura d'un document de la seva presentació.

❖ **XML<sup>4</sup>**: De la terminologia anglesa "eXtensible Markup Language". Llenguatge format per etiquetes que dona suport per exemple a Bases de Dades. Permet definir la gramàtica de llenguatges específics.

Un CSS no requereix una estructura igualitària, és a dir, tot el conjunt de codi CSS no està emmarcat dins de cap delimitació, sinó que dóna format a cada un dels elements que li són adjuntats. En aquest exemple, es mostra el disseny general de la pàgina Web:

```
body {  
  padding-left: 11em;  
  font-family: "Arial", Georgia,  
    Times, serif;  
  color: #000000;  
  background-color: #aec469;  
  width: 1100px;  
}
```

FIGURA 2.3: Disseny general de la Web en CSS

Si bé un HTML permet introduir elements dins la Pàgina Web, el CSS permet modificar-los per tal d'aconseguir que aquests tinguin una millor presentació. En el requadre anterior s'observa com el CSS permet escollir el tipus de lletra desitjat, el seu color, el color de fons i la mida.

#### 4.1.3.- JavaScript

JavaScript és un llenguatge interpretat (no es compila ni hi ha generacions intermediàries de codificacions de cap tipus). Aquest llenguatge permet inserir diverses Pàgines Web en una de sola a través de *marcs* (el marc utilitzat en aquest cas és el mapa de Google Maps). Aquests marcs s'executen a l'ordinador del visitant de la pàgina Web, i no al servidor d'aquesta.

El llenguatge JavaScript s'insereix a documents HTML, de forma que el seu codi queda reflectit a la pròpia pàgina i no és cridat de cap font externa. Un apartat de codi escrit en aquest llenguatge s'anomena *Script*.

Així doncs, tot el codi de JavaScript s'ha de citar a la pàgina HTML de la següent manera:

```
<SCRIPT Language="JavaScript">  
<!--Aquí anirà el codi -->  
</SCRIPT>
```

FIGURA 2.4: Estructura format JavaScript

Aquest llenguatge apareix a l'HTML de les pàgines que utilitzen prestacions de Google Maps dins el projecte. La seva funció és, a través d'Internet, demanar la informació de cada posició i les longituds entre punts a les prestacions per més tard, poder-la utilitzar en benefici propi. D'altra banda també s'utilitza per fer ús d'una taula dinàmica ja preparada coneguda amb el nom de *dataTable*.

#### 4.1.4.- Android

Android, no és un llenguatge de programació pròpiament dit, sinó que es un Sistema Operatiu creat especialment per *Smartphones*<sup>5</sup>. Tot i això, qualsevol persona pot crear aplicacions per introduir-les dins del sistema.

Les aplicacions d'Android es desenvolupen habitualment en llenguatge Java amb Android Development Kit o Android SDK (conjunt complet d'eines de desenvolupament, que inclou *llibries*<sup>6</sup> especialitzades diferents de les de Java), tot i que hi ha disponibles altres eines de desenvolupament.

El desenvolupament de les aplicacions Android no requereix aprendre llenguatges complexos de programació. Tot el que es necessita és un coneixement acceptable de Java i estar en possessió del kit de desenvolupament del software "*SDK*", el qual es pot descarregar gratuïtament des de qualsevol cercador com ara Google.

Totes les aplicacions estan comprimides en un format *APK*<sup>7</sup>, que permet instal·lar-les sense dificultat des de qualsevol explorador d'arxius en la majoria de dispositius.

- ❖ *Smartphone*<sup>5</sup>: Telèfon intel·ligent. És aquell telèfon mòbil que porta incorporat un sistema operatiu específic i que ofereix més capacitat de computació i connectivitat.
- ❖ *Llibreria*<sup>6</sup>: Conjunt de subprogrames utilitzats per desenvolupar software.
- ❖ *APK*<sup>7</sup>: És un tipus de format d'arxius comprimits amb diferents extensions.

## 4.2.- Conceptes bàsics de programació

Els conceptes explicats tot seguit fan referència a la eina bàsica de la programació, és a dir, el llenguatge, que no és més que un seguit de paraules clau que tenen assignat un funcionament o acció concreta, les quals formen el complet volum d'instruccions dels programes.

La major part dels llenguatges de programació utilitzen una sèrie de paraules i sentències que tenen moltes similituds. No obstant, cadascun d'aquests llenguatges té matisos diferents a l'hora d'escriure la mateixa instrucció. D'aquesta manera, les definicions explicades a continuació estan particularitzades al llenguatge Java ja que ha estat aquest llenguatge el que s'ha utilitzat per realitzar el desenvolupament de l'aplicació.

### 4.2.1.- Tipus de dades

En programació, les variables són espais reservats de la memòria que, com el seu nom indica, poden canviar el seu contingut durant la execució d'un programa.

En Java, s'ha d'associar un tipus de dada específic a totes les variables que s'utilitzen, el que en programació es coneix com a inicialització d'una variable. Abans d'utilitzar la variable doncs, és necessari inicialitzar-la per tal d'evitar errors.

Cada tipus de dada defineix tot el possible rang de valors que una variable pot prendre al llarg de la seva vida al programa.

A continuació es mostra els tipus de dades que s'han utilitzat al projecte:

VARIABLE	TIPUS	CAPACITAT
int	Nombre enter	4bytes (32bits)
long	Nombre enter	8bytes (64bits)
double	Nombre fraccionari	8bytes (64bits)
String	Cadena de textos	Varia segons els caràcters de l'String

*FIGURA 2.5:Tipus de dades*

Per decidir quina variable del mateix tipus però de diferent capacitat escollir, s'ha de contemplar la variable real amb la que es treballa. Per exemple, si la variable que s'utilitza és un nombre enter i se sap que aquesta pot prendre valors molt grans (majors de 4bytes), és millor declarar-la amb un *long* per tal d'evitar problemes amb el programa. Si la variable no ha de prendre valors molt grans, (sempre prenent com a mesura la seva capacitat) és millor declarar la variable com a *int* per tal



d'estalviar espai (no obstant el valor que prengui la variable, l'espai que ocupa sempre és l'establert per la taula de capacitats).

A diferència del llenguatge anteriorment citat, quan es programa amb JavaScript no és necessària aquesta especificació ja que el mateix llenguatge reconeix el tipus de variable que se li introdueix. Així doncs, totes les variables es declaren amb la mateixa paraula; **var**.

#### 4.2.2.- Tipus d'estructures

- *Estructures condicionals* - En programació, hi ha dues estructures condicionals; *if-else* i *switch*. Aquesta segona no s'explicarà ja que no ha sigut necessària la seva utilització. En el cas de la primera aquí s'exposa un exemple i una explicació:

```
if (condició) {
    instrucció1();
    instrucció2();
    // etc
}
else {
    instrucció1();
    instrucció2();
    // etc
}
```

*FIGURA 2.6: Estructura condicional if-else*

El funcionament de la construcció *if-else* és molt simple. En primer lloc, s'avalua la primera condició, si la expressió és certa, s'executa la instrucció o grup d'instruccions. En cas contrari l'execució del programa continua per la següent sentència en ordre seqüencial.

Aquesta construcció pot tenir tantes condicions com es desitgi, o pot tenir-ne una de sola, és a dir, es podria construir un *if-else-if* o un *if*.

- *Estructures de repetició* - Dins dels recursos de Java, es poden trobar diferents estructures de repetició o bucles (*while*, *do while*, *for*) entre els quals només s'ha fet servir la construcció iterativa *for*, que presenta la sintaxi següent:

```
for (sentència_inicial ; condició ;
increment/decrement)
{
    grup_de_sentències
}
```

*FIGURA 2.7: Estructura de repetició del bucle for*

La primera part de la construcció és on s'inicialitza la variable que controla el nombre de vegades que s'ha d'executar el cos del bucle. Aquesta sentència s'executa només una vegada abans d'entrar per primer cop dins el bucle.

La segona part correspon a la condició que indica quan ha de finalitzar el bucle. En aquest cas la condició s'avalua abans d'executar el cos del bucle.

La tercera part correspon a una sentència d'increment o decrement de la variable de control del bucle, que en aquest cas s'executa sempre després de la execució del cos del bucle.

### 4.2.3.- Operadors

Tots els llenguatges de programació permeten realitzar operacions sobre una, dues o tres dades de tipus primitiu, retornant un valor determinat també de tipus primitiu. Els mecanismes que permeten dur a terme aquestes operacions són els operadors.

A continuació, es mostren alguns dels tipus d'operadors utilitzats en aquest projecte i la seva funcionalitat:

OPERADOR	DESCRIPCIÓ
++,--	Increment i decrement
+, -	Suma i resta
!	<b>Complement booleà</b>
*,/,%	Multiplicació, divisió, residu
+	Concatenació
<, <=, >, >=	Comparació
==, !=	Igualtat i desigualtat
&&	<b>AND condicional</b>
=	Assignació

FIGURA 2.8: Operadors

#### 4.2.4.- Modificadors d'accés

Els modificadors d'accés permeten determinar qui pot accedir a les dades i mètodes que pertanyen a una classe, és a dir, n'indiquen la visibilitat.

Hi ha tres tipus de modificadors d'accés:

- *Private*: Només s'hi té accés des de la classe.
- *Protected*: Només s'hi té accés des de la mateixa classe i les classes que hereden d'ella, és a dir, aquelles classes que tot i tenir els seus propis mètodes i atributs, conté també els procedents de la classe pare.
- *Public*: Només s'hi té accés des de qualsevol classe definida al mateix *package*<sup>8</sup> (contenedor de classes que permet agrupar diferents parts d'un programa amb elements comuns) o bé des de qualsevol classe definida.

#### 4.2.5.- Objecte

La programació orientada a objectes (POO) abstruï les entitats del món real com a objectes i les relacions entre elles com a pas de missatges. Els objectes són instàncies o exemplars d'una classe o plantilla, i posseeixen com a característiques atributs (variables - valors) i mètodes (funcions - accions).

#### 4.2.6.- Classe

Una Classe de Java, és una plantilla on es defineixen els atributs i els mètodes per als objectes que es creen a partir de dita classe, és a dir, defineixen la forma i el comportament d'un objecte.

#### 4.2.7.- Atribut

Les dades, s'encapsulen dins d'una classe en forma d'atributs. Aquest atributs, no són més que variables declarades dins del domini d'una classe.

#### 4.2.8.- Mètode

Els mètodes són funcions definides dins d'una classe. Poden fer referència directament a les variables de la classe (atributs).

Aquests s'invoquen sobre un objecte de la classe a la qual pertanyen i en executar-los sobre un objecte d'una altra classe determinada, les seves variables canvien de valor i prenen els valors que tenen al nou objecte.

❖ *Package*<sup>8</sup>: Bloc o conjunt de classes que tenen una relació entre elles.

### 4.2.9.- Debugger

El debugger o depurador, és una eina que permet la execució controlada d'un programa o codi, per seguir cada instrucció executada i localitzar així *bugs* o errors.

Aquesta eina, ha estat molt útil per trobar petits errors que desencadenaven en un mal funcionament de la execució. Però no només ha sigut útil en aquest sentit, sinó que també ha permès executar petites parts de codi de manera autònoma i veure detalladament què succeeix durant la part escollida.

### 4.3.- Conceptes relacionats amb el projecte

Per a la creació completa del projecte ha sigut necessari l'estudi i comprensió dels aspectes que es presenten tot seguit.

No tenen relació directa entre ells, ni tampoc amb els conceptes explicats anteriorment, tot i això, és necessari tenir-ne una noció bàsica per comprendre la part de memòria pràctica. A més a més, ja s'intenta explicar el paper que tenen en el projecte.

- **Base de dades** - Una base de dades és una entitat la qual permet emmagatzemar grans quantitats d'informació de forma organitzada per a poder-la localitzar i utilitzar fàcilment.

El que aquesta proporciona, és l'accés de l'usuari a les dades, podent així visualitzar-les, ingressar-les o actualitzar-les depenent dels drets que l'usuari disposi.

Una base de dades pot ser de dos tipus:

- *Local*: Utilitzada només per un usuari a un ordinador. Aquest es el tipus de base de dades que s'ha fet servir.
- *Distribuïda*: La informació s'emmagatzema en equips remots i es pot accedir a ella a través d'una xarxa.

La funció de la base de dades, és emmagatzemar les dades que l'usuari proporciona de la ruta a través de la Pàgina Web. Aquesta informació es col·lecciona en forma de taules (una de rutes i una de posicions), de manera que cada ruta i cada posició de la ruta posseeix una fila de la seva respectiva taula.

- **Servidor** - A Internet, un servidor és un ordinador remot que proporciona les dades sol·licitades per part dels navegadors d'altres ordinadors. És a dir, s'entén per servidor l'ordinador

físic en el que funciona un *software*<sup>9</sup> que realitza algunes feines en nom de l'usuari. Aquesta màquina té el propòsit de proveir dades de manera que altres màquines puguin utilitzar-les.

Els servidors emmagatzemen informació en forma de pàgines Web i a través dels protocols HTTP i HTTPS les entreguen a petició dels clients (navegador Web) en format HTML.

Un servidor pot ser:

- *Local* - Com el que s'ha fet servir en aquest treball. És aquell que s'instal·la, per mitja d'un programa, al propi ordinador per permetre provar i navegar la Pàgina Web que es vol crear des d'un equip personal.
- *Remot* - Els servidors remots, en canvi, són ordinadors molt més grans i que posseeixen majors característiques que els equips normals. Aquests són els que donen un espai per emmagatzemar un lloc Web, és a dir, que permeten tenir el que s'anomena un *Hosting*<sup>10</sup> (allotjament Web). Aquest necessita un nom per al lloc Web, és a dir, un Domini, per mitjà del qual qualsevol persona podrà accedir a la pàgina Web a través de la xarxa. Els servidors d'aquest tipus s'han de llogar si es vol gaudir dels seus serveis, la qual cosa suposa un cost.

- *Protocols HTTP i HTTPS* - Quan es parla de protocols de comunicació es fa referència al conjunt de regles normalitzades que representen, senyalitzen, autentifiquen i detecten els errors que es troben en una informació abans d'enviar-la a través d'un canal comunicatiu. Aquesta acció és necessària per assegurar el bon funcionament del sistema.

El propòsit del protocol HTTP (*HyperText Transfer Protocol*) és permetre la transferència d'arxius (format HTML) entre el navegador (usuari) i un servidor web, connectats a través d'uns caràcters anomenats direcció URL.

D'altra banda, el protocol HTTPS (*HyperText Transfer Protocol Secure*), és un protocol basat en l'HTTP però destinat a la transferència segura de dades, és a dir, utilitza un canal encriptat amb l'objectiu d'aconseguir una informació que no pugui ser utilitzada per un atacant que hagi aconseguit interceptar la transferència de dades. La única informació que obtindrà serà un seguit de dades encriptades que resultaran impossibles de desxifrar.

Aquests són utilitzats internament quan es fan les peticions Web de l'usuari al servidor i quan es dona resposta del servidor a l'usuari.

❖ *Software*<sup>9</sup>: Tot el conjunt de programes informàtic i procediments que realitzen alguna tasca dins l'ordinador

❖ *Hosting*<sup>10</sup>: Servei que proporciona als usuaris d'Internet un sistema per poder emmagatzemar tot tipus de dades.

- **Tipus de pàgines web** - Segons la tipologia de construcció, les pàgines Web es poden classificar en dos grans grups. El propòsit d'aquest apartat és veure que la dificultat de construcció de la pàgina Web d'aquest treball és més complexa del que ho són les que podem consultar normalment.

- *Pàgines Web estàtiques* - Les pàgines Web estàtiques són aquelles el contingut de les quals no canvia freqüentment i alguna persona especialitzada s'encarrega del seu manteniment manualment.

Són pàgines escrites en HTML que únicament mostren informació. S'anomenen estàtiques ja que el visitant no té la possibilitat d'interactuar amb el contingut de la pàgina, tot i que poden incloure vídeos o animacions.

Aquesta opció és suficient per a aquells llocs Webs que simplement ofereixen una descripció de la seva empresa, per exemple; qui són, on s'ubiquen, serveis que ofereixen, productes, etc. Alguns exemples de pàgines d'aquest tipus són la pàgines de diaris com La Vanguardia, Ara, la Web del Institut l'Alt Berguedà, etc.

- *Pàgines Web dinàmiques* - Aquest és el tipus de pàgina Web que s'ha creat en el projecte. S'anomena pàgina Web dinàmica a aquella el contingut de la qual es genera a partir del que un usuari introdueix en un formulari.

Així doncs, el contingut de la pàgina no està inclòs en un arxiu HTML - com en el cas de les estàtiques - ja que està en canvi constant. Així doncs, el contingut és inclòs utilitzant altres llenguatges de programació.

Alguns exemples de pàgines dinàmiques són les pàgines de compra on-line, e-mails, foros, pàgines d'atenció a l'usuari, etc.

- **URL** - (*Uniform Resource Locator*) És una seqüència de caràcters, d'acord amb un format modèlic i estàndard, que es localitza a la barra de direccions del navegador web i que fa d'única adreça identificadora d'una pàgina Web d'Internet.

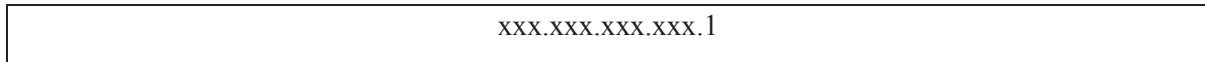
L'URL de la Pàgina Web creada és:

<a href="http://localhost:8080/VisualitzadorWebDeRutes/">http://localhost:8080/VisualitzadorWebDeRutes/</a>
---

*FIGURA 2.8: URL de la pàgina Web*

- **Direcció IP** - (Internet Protocol adress) És un seguit de números únics e irrepetibles amb els quals s'identifica un ordinador connectat a una xarxa que utilitzi aquest protocol.

Aquesta adreça està composta per quatre nombres enters entre el 0 i el 255 escrits segons el patró del format següent:



*FIGURA 2.9: Estructura d'una direcció IP*

Tot i que cada adreça és diferent, hi ha semblances entre alguns col·lectius. Per exemple, les IP dels particulars tenen una IP al *localhost* en comú, els grans col·lectius universitaris comencen per un número en concret, etc.

- **Localhost / loopback** - Per accedir a una pàgina Web, sempre s'hi accedeix mitjançant una direcció o URL. L'accés per navegar dins el propi equip es coneix com *localhost*. Aquest és un nom reservat que tenen tots els ordinadors, routers o dispositius que disposin d'una targeta de xarxa ethernet per referir-se a si mateixos. El nom *localhost* és traduït com la direcció IP *127.0.0.1*.

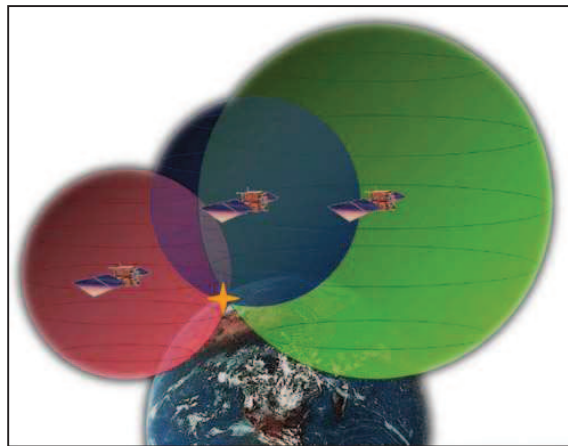
Així doncs, si es vol dur a terme una pàgina Web, es requereix la instal·lació d'un servidor local, que permeti accedir des del propi ordinador a la Pàgina Web creada amb la finalitat de fer proves sense contractar un *hosting* (servidor remot). La diferència és que només el propietari pot accedir a la seva pàgina web.

#### 4.4.- Les coordenades GPS

El GPS (*Global Position System*), és un sistema global de Navegació per satèl·lit (GNSS), que permet determinar a qualsevol part del món la posició d'un objecte o persona, amb una precisió molt elevada (pocs metres o fins i tot centímetres si s'utilitza un GPS diferencial).

Aquesta eina funciona gràcies a una xarxa de 27 satèl·lit terrestres (24 operatius i 3 de suport), ubicats a una òrbita al voltant del globus de 20.200 Km amb trajectòries sincronitzades per cobrir tota la superfície de la Terra.

Per determinar la posició, el receptor ubicat a la superfície terrestre, en aquest cas un Smartphone, localitza automàticament un mínim de tres satèl·lits de la xarxa, dels quals rep senyals indicant-ne la posició i l'hora a cada un d'ells. Finalment, per triangulació calcula la posició en que es troba.



*FIGURA 2.10: Càlcul de posicions per Triangulació*

Els receptors GPS estimen la distància als satèl·lits segons el temps que trigen les senyals en arribar al receptor i després, utilitzen aquesta informació per calcular les posicions.

El posicionament GPS doncs, proporciona la latitud i la longitud del punt en el que es troba la persona sobre la superfície terrestre. Per tant, la majoria de receptors proporcionen els valors d'aquestes coordenades en unitats de graus (°) i minuts ('). Tant la longitud com la latitud són angles, pel que es mesuren respecte a un angle 0° de referència ben definit; El meridià de Greenweech a Anglaterra que mesura latitud 0°, i el paral·lel de l'Equador la longitud 0°.



*FIGURA 2.11: Paral·lel d'Equador i meridià de Greenwich*



## 5.- PROJECTE

### 5.1.- Disseny

En l'elaboració de qualssevol projecte hi ha un seguit de passos que cal tenir en compte. A continuació es presenta el primer.

Inicialment s'ha de fer un plantejament dels objectius que volen aconseguir-se i, per tal d'arribar a aquesta fita, quin serà el procediment més oportú a seguir. Per aquest motiu una de les primeres tasques a l'hora de realitzar la implementació ha estat fer un esbós previ del projecte. Aquest esbós consta del plantejament bàsic i funcional del treball i alhora de la construcció d'un pre-disseny que mostri el que es realitzarà finalment.

Aquest treball es caracteritza per la utilització de dos aparells aïllats que interactuaran l'un amb l'altre i, per tant, és necessari un disseny per a cada un d'ells que permeti i faciliti el contacte recíproc entre l'usuari i el programa intern.

Abans de començar a crear el projecte és imprescindible pensar quina serà l'estructura tant visual com funcional que es vol aconseguir, i així plantejar una idea de totes les funcions que interessa que es duguin a terme.

#### 5.1.1.- Descripció del comportament

A continuació es descriu de manera seqüencial el comportament global del sistema:

- L'usuari inicia la mesura de les posicions GPS mitjançant el programa Android "Capturador de Rutes".
- Aquest programa recull les posicions periòdicament i les va emmagatzemant en un fitxer de text.
- Quan l'usuari decideix finalitzar la ruta, el fitxer complet s'emmagatzema a la memòria interna del telèfon mòbil.
- Seguidament, l'usuari ha de connectar el seu smartphone a l'ordinador mitjançant un cable USB o bé a través de connexions sense fil (WiFi, bluetooth..).
- El fitxer desitjat es guarda dins d'una carpeta creada especialment per a realitzar la funció d'emmagatzematge dels fitxers de rutes. (Aquest es podrà guardar a qualsevol lloc de l'ordinador o obrir la pàgina Web i enviar-lo directament des del mòbil).
- En accedir a la pàgina web, es pot escollir el nom de la ruta i afegir un comentari per a aquesta.

- En afegir aquests dos camps, s'ha de localitzar a l'ordinador el fitxer desitjat i seguidament es fa l'acció de desar-lo.
- Un cop desada la ruta a la Web, aquesta apareixerà dins d'una taula que en proporciona algunes dades, tals com el nom, l'instant de temps inicial i final, la longitud i el comentari. (Aquesta taula disposa d'un botó que permet a l'usuari eliminar en qualsevol moment les rutes que ja no desitgi visualitzar més).
- El nom de la ruta és un hipervincle que permet accedir a una altra pàgina on es troben totes les dades que el programa aconseguix i calcula.
- La nova pàgina, conté una taula amb dades de la ruta, les mateixes que a la taula anterior però amb millores com la incorporació del temps total, de l'altitud màxima i la mínima. A més a més, conté un mapa amb la ruta inscrita i amb marcadors a cada punt on l'aplicació del mòbil hagi capturat la informació corresponent.
- Es podran visualitzar les dades de cada posició; instant de temps, alçada, longitud i latitud.

Per poder dur a terme totes les accions i funcions citades, ha d'existir una interacció entre programa i usuari, però també interaccions internes entre aparells. A continuació s'exposa un diagrama de blocs del sistema que descriu la direcció del flux de dades i que clarifica quins són els sub-blocs que intervenen en el desenvolupament de totes les funcionalitats del treball i que seran tractats de manera individual en les pròximes pàgines.

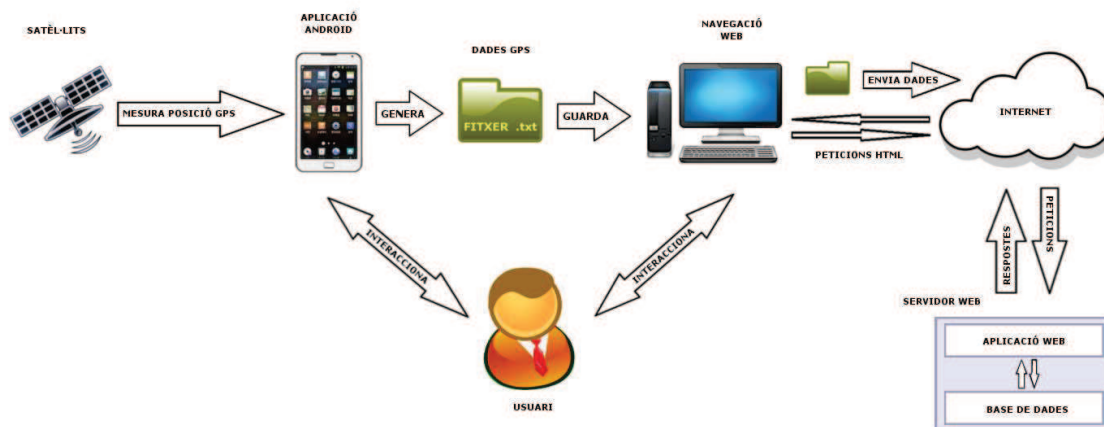


FIGURA 3.1: Diagrama de blocs del sistema complet

## 5.1.2.- Esbós Pàgina Web

### Esbós 1

El primer esbós que s'ha realitzat s'ha plasmat i dibuixat a mà per tal de poder retocar amb més facilitat els canvis a mesura que s'hi afegien totes les eines necessàries. D'altra banda, un segon esbós, ja més definitiu s'ha realitzat a l'ordinador amb un programa especialitzat anomenat *Balsamiq Mockups*, que ja permet una interacció entre pàgines a nivell d'usuari.

Aquest esbós doncs, està format per cinc pantalles:

- Pàgina inicial
  - Pàgina principal
  - Rutes emmagatzemades
  - Ruta 1
  - Dades específiques
- La pàgina inicial només conté una fotografia –link que redirecciona a la pàgina principal.

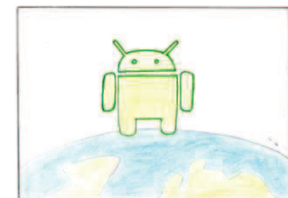


FIGURA 3.2: Pàgina Inicial

- La pàgina principal, està formada pel títol de l'aplicació, una fotografia i dos botons; un que dirigeix a la pàgina *Rutes emmagatzemades*, i l'altre a *Descarrega noves rutes*.

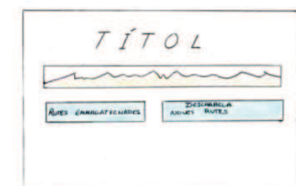


FIGURA 3.3: Pàgina principal

- A *Descarrega noves rutes* s'ha de cercar dins l'ordinador el fitxer que conté les dades, i en obrir-lo es guarda dins d'una taula amb totes les rutes que ja s'han descarregat a la pàgina que s'accedeix des del botó *Rutes emmagatzemades*.

- En aquesta pàgina, hi ha una taula amb el nom de totes les rutes, i cadascun dels noms és un hipervincle que permet accedir a la pàgina de visualització de la ruta.

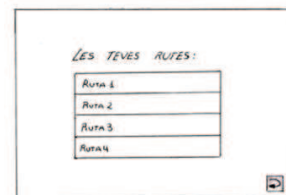


FIGURA 3.4: Rutes Emmagatzemades

- A la pàgina de visualització es troba el mapa amb la ruta plasmada i les dades del punt d'inici i final. A més incorpora un botó que permet accedir a *Dades específiques*.



FIGURA 3.5: Ruta 1

- La pàgina citada (*dades específiques*) conté dades de la ruta tals com el desnivell, l'alçada màxima i mínima, la distància total i el temps total.

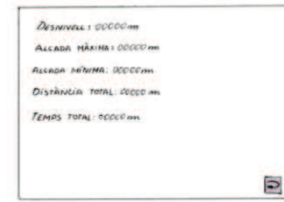


FIGURA 3.6: Dades específiques

Cada una de les pàgines, excepte la pàgina d'inici, tenen un botó que permet accedir a la pàgina anterior, és a dir, estan enllaçades les unes amb les altres de manera que es pugui navegar entre elles.

### **Esbós 2**

El segon esbós, realitzat amb el programa *Balsamiq Mockups*, ja permet observar l'estructura amb enllaços visibles. La seva organització és molt semblant a la explicada al primer esbós tot i que presenta petites diferències que modifiquen la manera estructural del disseny en alguna de les pàgines.

En aquest esbós, la pàgina que conté les rutes emmagatzemades, està en forma de menú, és a dir, a la part esquerra de la pantalla hi ha un menú amb el nom de totes les rutes, i en polsar sobre qualsevol nom, s'obre la pàgina de visualització tal qual es presenta en l'esbós principal, però sense desaparèixer el menú citat.

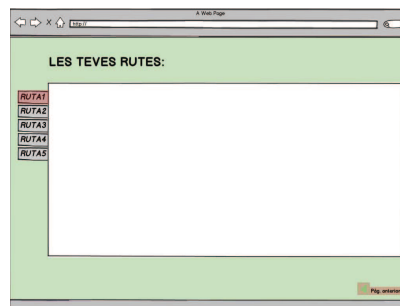


FIGURA 3.7: Visualització del menú amb Balsamiq Mockups. La resta de l'esbós es troba a l'Annex VIII

### **5.1.3.- Esbós programa Android**

L'esbós manual que s'ha fet per l'smartphone també conté cinc pantalles amb les que és possible interactuar:

- Pàgina inicial
- Pàgina principal
- Inici ruta
- Acaba
- Rutes emmagatzemades

- La primera pàgina és una imatge – link que redirecciona a la pàgina principal.



FIGURA 3.8: Pàgina inicial

- A la següent pàgina es troba un títol, una imatge i dos botons. El primer botó *Inici Ruta* obre una pàgina en què hi ha dos botons, *Comença* i *Acaba*.



FIGURA 3.9: Pàgina principal

- En el moment en que es posa comença, no es pot pausar el programa fins que es finalitzi la ruta amb el botó *Acaba*.
- Durant la ruta, es podrien observar dades instantànies de la ruta a la pantalla tals com temps, alçada actual i distància recorreguda.

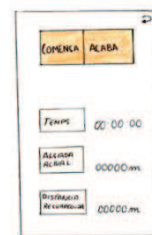


FIGURA 3.10: Inici ruta

- En pulsar el botó *Acaba*, el programa envia a l'usuari a una pàgina en la qual ha d'escriure el nom de la ruta.
- Un cop escrit el nom, s'utilitza el botó *Accepta* per redireccionar l'usuari a la pàgina Rutes emmagatzemades, a la qual també es pot accedir des de la pàgina principal.



FIGURA 3.11: Acaba

- En aquesta pantalla, es troba una taula amb totes les rutes realitzades.

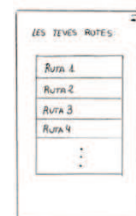


FIGURA 3.12: Rutes emmagatzemades

(Totes les pàgines, excepte la primera, incorporen un botó que permet anar a la pàgina anterior).

En passar el disseny a l'ordinador amb el programa ja citat *Balsamiq Mockups*, el funcionament és exactament el mateix amb una petita diferència; a la pàgina de rutes emmagatzemades (on es troba la taula amb els respectius noms de cada ruta) hi ha un botó que permet eliminar totes les rutes emmagatzemades al mòbil, per tal de no acumular-ne masses.



*FIGURA 3.13: Rutes emmagatzemades amb Balsamiq Mockups. La resta de l'esbós es troba a l'Annex IX*

## 5.2.- Implementació

### 5.2.1.- Aplicació Web

Rere l'esbós de la pàgina Web comença la implementació. Aquest procés consta de diferents etapes les quals s'expliquen en els propers apartats.

A mesura que es comença amb aquesta etapa d'implementació les funcions que es plantegen inicialment comencen a rebre modificacions degut a diferents factors: problemes que sorgeixen a l'hora de la programació del codi, buscar els camins més fàcils per al programador (ja que la intenció d'aquest projecte és la elaboració d'una aplicació funcional bàsica), però que a l'hora sigui pràctic per l'usuari, etc. Així doncs, durant els propers apartats es veurà l'evolució que pateix un projecte des de que es planteja el primer esbós fins a que es programa l'aplicació final.

#### 5.2.1.1.- Elaboració de la plantilla

Quan es parla de plantilla es fa referència a cadascuna de les pàgines que constituïran la web sencera, és a dir, les pàgines que es volien visualitzar i que serviran de mitjà de comunicació entre l'usuari i la màquina.

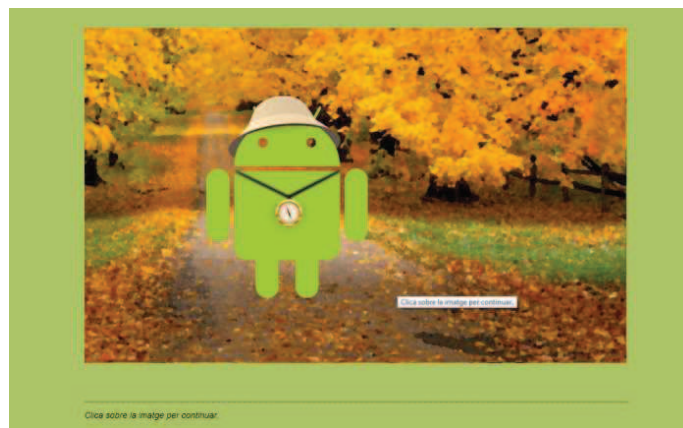
Aquesta etapa, doncs, consta de l'escriptura en HTML del contingut de cada pàgina. Aquest contingut és simplement a nivell d'entorn gràfic i d'usuari, sense entrar dins de la programació de funcionalitats. Així doncs, l'objectiu d'aquest procés d'elaboració de la plantilla és crear un contenidor que serveixi perquè posteriorment es puguin emmagatzemar, manipular i consultar les dades processades.

En aquest punt i a conseqüència dels factors que s'exposaran a continuació s'ha decidit dur a terme les modificacions pertinents i que queden reflectides en el disseny definitiu de la web.

## Pàgina 1 – BENVINGUDA

Un dels primers problemes que va provocar un canvi respecte de l'esbós inicial és el de les mides de la pàgina. En HTML, els patrons que s'insereixen al codi no s'adapten a la mida de cada pantalla d'ordinador per defecte, sinó que manualment el programador ha de posar la mida que vol per a cada instrument de la seva pàgina. Així doncs, la inserció d'una imatge que ha de realitzar la funció d'enllaç cap a la pàgina *Principal* ajustada a una pantalla panoràmica suposa un problema degut a que les pantalles de baixa resolució no poden veure la imatge sencera sense moure les barres de moviment horitzontals.

La solució al problema anterior és que per a la pàgina de *Benvinguda* s'ha buscat una mida estàndard (màxim 1000 píxels) que tot i no quedar centrada a totes les pantalles del mercat, permet a la gran majoria visualitzar-ho degudament. Aquesta pàgina per tant, no ocupa tota la pantalla, sinó que només una part. La resta conté un fons de color i un detall tal com una línia amb un comentari.



*FIGURA 3.14: Pàgina benvinguda.html*

A continuació es mostra el codi HTML d'aquesta primera pàgina amb els comentaris pertinents:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

  <head>

    <meta http-equiv="Content-Type" content="text/html; charset=ISO-
8859-1">

    <title>Benvinguda</title>

    <link rel="stylesheet" href="css/Style.css"> //Referència al
full d'estil de la pàgina en CSS
```

```
</head>

<body>

    <a title="Clica sobre la imatge per continuar." //Informació
visible en passar el cursor per damunt la imatge
    href="principal.html">

        <div id="imaginicial"></div> //Imatge

    </a>

    <address>Clica sobre la imatge per continuar.</address> //Peu de
pàgina

</body>

</html>
```

FIGURA 3.15: Codi HTML de la pàgina benvinguda.html

## Pàgina 2 – PRINCIPAL

En primer lloc, inconvenient en el format; tal i com es presenta el disseny, no hi ha cap entrada en la qual l'usuari pugui inserir el nom que desitja per la ruta descarregada i, per tant, els noms de les rutes son escollits pel mòbil i no es poden modificar.

La solució implementada a aquest inconvenient ha estat fer un formulari a la pàgina *Principal.html* que permeti a l'usuari introduir el nom de la ruta i com a extra, un comentari que la pugui identificar amb més facilitat. La incorporació d'aquest formulari suposa un gran avantatge ja que pot comportar un estalvi de programació de funcions a l'aplicació del mòbil tals com la pàgina en què es posa el nom de la ruta i la pàgina on es veuen els noms de les rutes emmagatzemades. Per tant, l'arxiu mòbil es guardarà amb un nom per defecte que es podria modificar únicament des de la Pàgina Web.





FIGURA 3.16: Pàgina principal.html

D'altra banda, la idea del menú plantejat al segon esbós, tot i ser molt pràctica presenta alguns inconvenients que desfavoreixen la seva implementació final, ja que el fet que el menú no hagi de ser fixa (les rutes augmenten i disminueixen segons la tria de l'usuari) en dificulta la seva programació.

La substitució del menú per una taula dinàmica que conté els noms de les rutes a *principal.html* proporciona als usuaris visualitzar una pàgina més completa i agradable. Aquesta taula, a part de contenir el nom de la ruta, conté també algunes dades que permeten reconèixer-la amb més facilitat (longitud, instant de temps inicial i final i comentari). A més s'ha afegit una opció que permet esborrar la ruta si l'usuari no desitja visualitzar-la més.

id	Nom de la ruta	Instant inicial	Instant final	Longitud (km)	Comentari	Esborrar
1	<a href="#">Vinya Vella</a>	2012/12/01 14:00:00	2012/12/01 14:22:00	1.45	Vaig trobar bolets a mig camí.	<input type="button" value="X"/>

Showing 1 to 1 of 1 entries

First Previous 1 Next Last

FIGURA 3.17: DataTable continguda a principal.html

Aquesta taula com és evident, no és estàtica, sinó que va canviant en funció de les peticions de l'usuari. El fet que sigui dinàmica doncs, ha portat a que aquesta sigui un tipus de taula especial anomenada *DataTable*, que facilita el moviment i alhora, la feina del programador.

En conclusió, finalment la pàgina *Principal*, a part d'una fotografia ja prevista i d'alguns comentaris fixes que fan la pàgina més divertida, conté un formulari per introduir dades de la ruta i una taula dinàmica que varia en funció de la voluntat de l'usuari.

A continuació es presenta una mostra del codi que ha sigut necessari per implementar la *DataTable* utilitzada en aquesta pàgina:

```
<script type="text/javascript">
    $(document).ready(
        function() {
$.getJSON("rutesEmmagatzemades.html",function(data) {
if (data) {
    $('#rutesEmmagatzemades').html('<div class="dataTables_wrapper"
id="example_wrapper"> <table cellpadding="0" cellspacing="0" border="0"
class="display" id="example"></table></div>');
$('#example').dataTable({
    "iDisplayLength": 7,
    "bJQueryUI": true,
    "bLengthChange": false,
    "bFilter": true,
    "bInfo": true,
    "sPaginationType" : "full_numbers",
    "aaData": data,
    "aoColumns": [{"sTitle" : "id"},
{"sTitle" : "Nom de la ruta"}, //Columna que conté el nom de la ruta
{"sTitle" : "Instant inicial"},//Columna que conté l'instant de temps inicial
{"sTitle" : "Instant final"},//Columna que conté l'instant de temps final
{"sTitle" : "Longitud (km)"}, //Columna que conté la longitud
{"sTitle" : "Comentari"},// Columna que conté el comentari
{"sTitle" : "Esborrar"}] // Columna que conté un botó per esborrar la ruta
    });
    } else {
        $('#rutesEmmagatzemades').html("");
    }
    });
});
</script>
```

FIGURA 3.18: Codi empleat per utilitzar la *dataTable*

Aquest codi, s'encarrega de fer la petició al servidor de les dades necessàries per omplir la taula. S'executa automàticament en el moment en que el client ja ha descarregat la pàgina (gràcies a `$(document).ready()`) i envia una petició a la url `<rutesEmmagatzemades.html>`. La resposta del servidor són les dades que han estat generades adequadament a les característiques de la taula. Com es pot veure, es fa també la configuració de certs paràmetres de la taula, entre ells el nombre

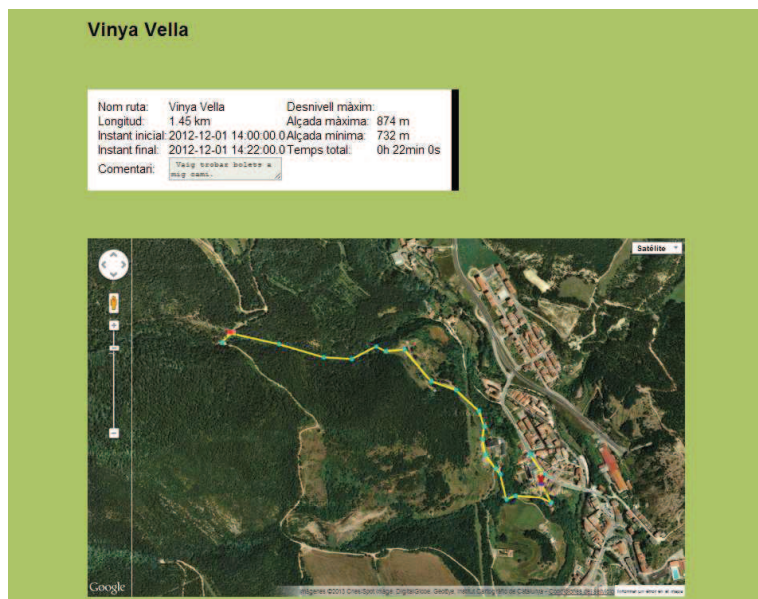
màxim de files que pot tenir <"iDisplayLenght": 7 > i el nom de cadascuna de les columnes <"aoColumns": [{"sTitle": "id"}, {"sTitle": "Nom de la ruta"}] ... >

### Pàgina 3 – DETALL RUTA

Rere l'agrupació d'aquests elements en una sola pàgina, queda la visualització de la ruta. A aquesta s'hi podrà accedir polsant al nom de la ruta desitjada un cop ja descarregada dins la taula.

Un altre cop, posar les dades específiques dins d'una pàgina aïllada seria una pèrdua d'espai ja que es podrien posar totes les dades dins d'una mateixa taula a la pàgina en què també permet visualitzar el mapa i la ruta plasmada en ell.

Així doncs, dins d'aquesta nova pàgina anomenada *DetallRuta.html*, es troba el mapa amb la ruta i sobre d'aquest una taula amb totes les dades que s'han desitjat donar d'una ruta.



*FIGURA 3.19: detallRuta.html*

#### 5.2.1.2.- Definició del funcionament

En aquest apartat es presenten diferents diagrames que permeten visualitzar el funcionament de l'aplicació.

El primer que es mostra, és un diagrama el qual permet veure les connexions entre pàgines Web. En aquest cas, només hi ha tres pàgines programades en HTML.

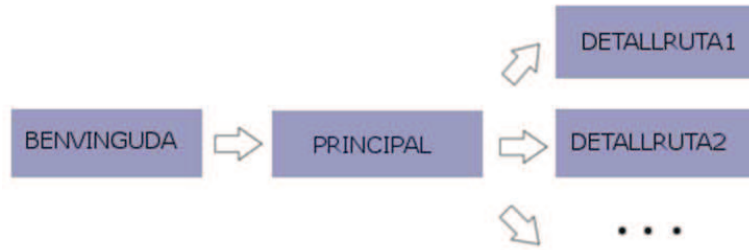


FIGURA 3.20: Diagrama de flux

Quan l'usuari realitza una acció amb la Web, el que fa és realitzar una petició al servidor. Aquest pot retornar un fitxer HTML (nova pàgina) o bé dades demanades per l'usuari (dades d'una taula). El següent esquema representa l'intercanvi per a aquesta aplicació:

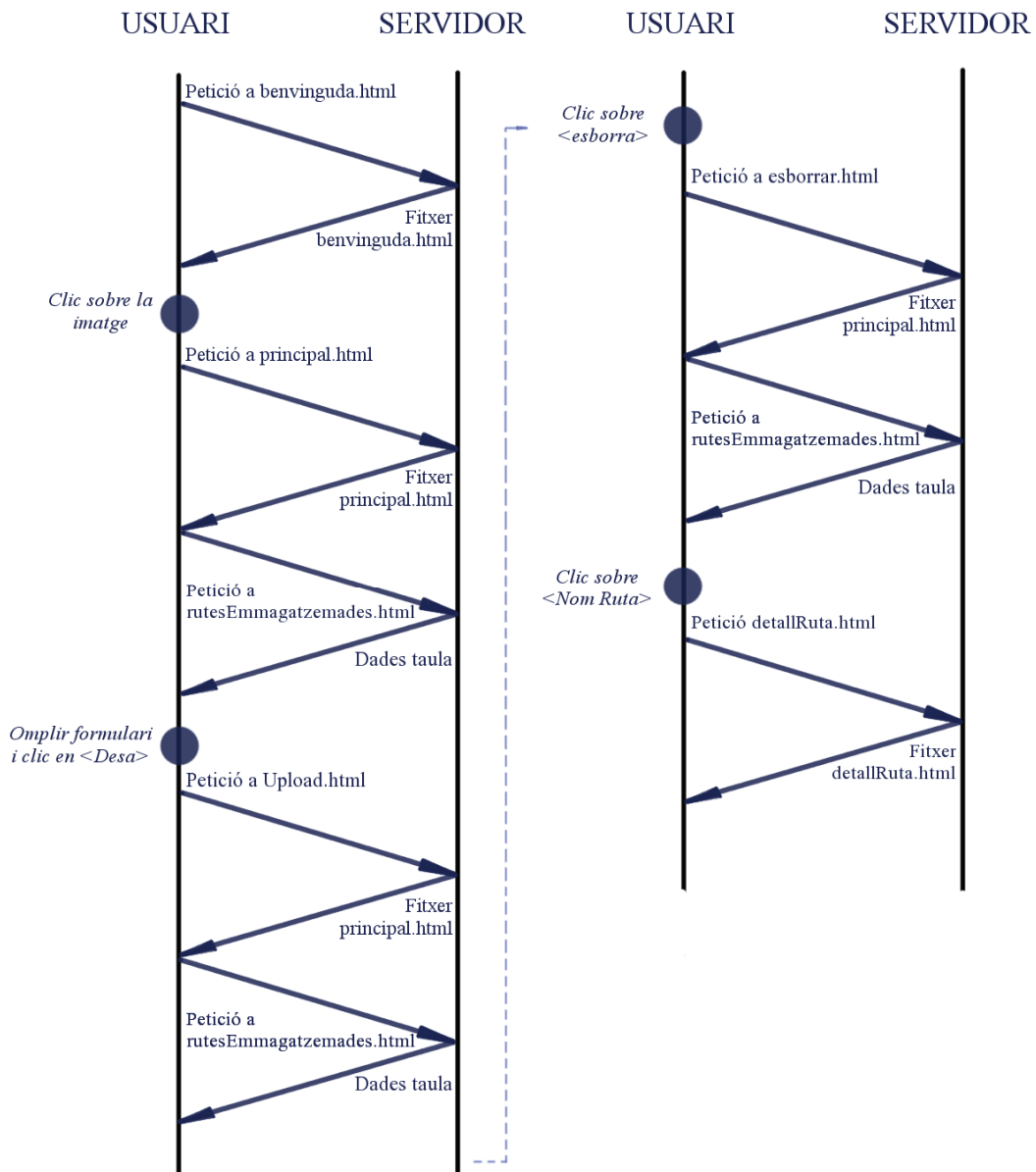


FIGURA 3.21: Diagrama de peticions

Interpretant el diagrama anterior, s'obté la següent seqüència lògica;

1. L'usuari fa una petició a `benvinguda.html` i el servidor respon enviant-li el fitxer `benvinguda.html`.
2. Si l'usuari clica sobre la imatge, envia una petició a `principal.html` i el servidor respon enviant-li el fitxer `principal.html`.
3. Quan l'usuari ha rebut `principal.html`, s'envia automàticament mitjançant Javascript, una petició a `rutesEmmagatzemades.html`. El servidor, en rebre aquesta petició, sap que l'usuari està sol·licitant les dades per omplir la taula de les rutes, per tant, consulta la base de dades i obté totes les rutes guardades. Transforma aquestes dades en una matriu que envia a l'usuari com a resposta.
4. Si l'usuari omple el formulari i clica sobre "desa", envia les dades introduïdes incloent el fitxer adjuntat, cap al servidor, utilitzant `upload.html` com a adreça de destí. El servidor rep les dades i llegeix el fitxer línia a línia emmagatzemant la nova ruta i les corresponents posicions llegides del fitxer a la base de dades. Redirecciona a l'usuari cap a `principal.html` que actualitza de nou la taula.
5. Si l'usuari clica sobre alguna creu que apareix a una de les columnes de la taula, s'envia una petició a `esborrar.html` adjuntant el número de la ruta que es vol eliminar. El servidor esborra la corresponent ruta, així com les seves posicions associades, i l'usuari actualitza la taula de nou.
6. Si l'usuari fa clic sobre el nom d'alguna de les rutes que apareixen a la taula, es realitza una petició a l'adreça `detallRuta/idRuta.html`, on `idRuta`, és el número de la ruta que es vol consultar. El servidor envia el fitxer `detallRuta.html`, junt amb les dades necessàries per tal de dibuixar a Google Maps la ruta i per a omplir els camps que s'han de mostrar com per exemple la longitud, temps, etc.

Totes aquestes peticions són gestionades pel servidor fent servir una estructura com la següent:

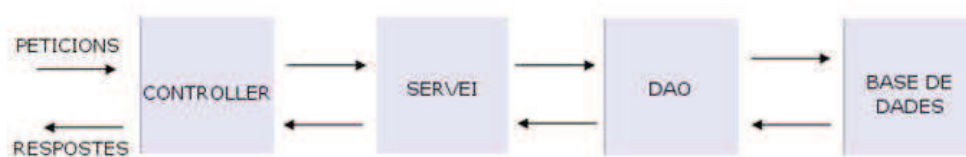


FIGURA 3.22: Diagrama de gestió de peticions

En aquest últim diagrama doncs, s'observa com les peticions que envien els usuaris al servidor, estan gestionades per una primera etapa que és el *controlador*. La feina del controlador és escoltar les possibles peticions dels usuaris i mitjançant un *servei*, dur a terme el resultat de la petició. En aquest cas només es reben peticions relacionades amb les rutes, per tant, només hi haurà un servei que s'encarregarà de posar-se en contacte amb el *DAO* per accedir a la base de dades. El DAO (*Data Access Object*), és l'únic que té accés directe a la base de dades gràcies a la configuració que s'ha fet de les seves classes mitjançant utilitats de *hibernate*. Les dades consultades a la base de dades es van retornant i transformant fins arribar de nou al controlador que s'encarrega de preparar les dades per ser enviades a l'Usuari.

### 5.2.1.3.- Descripció de les classes

A continuació es presenta l'esquema complet del projecte on es pot observar la jerarquia que s'ha seguit, amb cadascun dels directoris, les classes que formen part d'aquest directoris, llibreries, etc.

Primerament es descriuran els continguts programats en Java. Aquests continguts són els que s'observen a la part superior de l'esquemàtic, és a dir, el package *src/main/java*. Els paquets *JRE System Library* i *Maven Dependencies* són llibreries que s'han fet servir per a la programació.

#### CONTROLLER

Les classes que estan contingudes al package Controller són les que s'encarreguen de la comunicació entre el servidor i l'usuari. Cada classe que conté controla una plana diferent de la web. Concretament es controla les peticions HTML i també les transicions degudes a la navegació que l'usuari fa per la web, així com la resposta amb l'enviament de les dades a mostrar, ja sigui un document HTML o bé un conjunt de dades concret.

Això que pot semblar complicat és fàcilment implementable gràcies a les llibreries que proporciona la eina Spring. Aquestes llibreries permeten afegir una sèrie d'anotacions a documents java que configuren el servidor per tal de dur a terme l'entesa entre aquest i l'usuari.

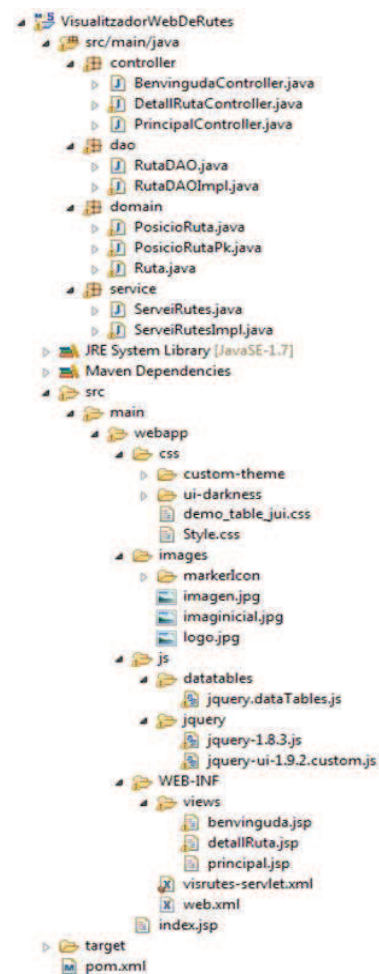


FIGURA 3.23: Esquema dels directoris

Per exemple, el codi que controla la petició a la pàgina benvinguda és el següent:

```
@RequestMapping(value = "/benvinguda", method = RequestMethod.GET)
public String getBenvinguda(){
    System.out.println("Benvinguda!");
}
```

FIGURA 3.24:petició a benvinguda.html

L'anotació que es pot veure sobre la funció “getBenvinguda()” s'encarrega de configurar el servidor de manera que quan es fa una petició a “VisualitzadorWebDeRutes/benvinguda.html” el servidor executa la funció que envia cap a l'usuari el fitxer “benvinguda.html”.

- BenvingudaController.java - Aquesta classe únicament s'encarrega de servir les peticions a la pàgina “benvinguda.html”
- PrincipalController.java - S'encarrega de 4 peticions al servidor:
  - *Petició a “/principal.html”* - Envia a l'usuari el fitxer “principal.html”
  - *Petició a “upload.html”* - Petició realitzada per l'usuari quan envia les dades d'una ruta. La funció s'encarrega de rebre aquestes dades i guardar-les a la base de dades.
  - *Petició a “/rutesEmmagatzemades.html”* - Petició realitzada de manera automàtica al carregar-se la pàgina principal desde la taula que mostra les rutes. El servidor respon a la petició consultant a la base de dades totes les rutes i enviant-les cap a l'usuari.
  - *Petició a “/esborrar.html”* - Petició realitzada des de l'usuari al indicar al servidor que vol esborrar una ruta.
- DetallRutaController.java - S'encarrega de servir la petició al conjunt de pàgines “/detallRuta/{idRuta}”, on “{idRuta}” pot variar prenent el valor de l'identificador d'alguna de les rutes emmagatzemades. El servidor envia el fitxer “detallRuta.html” modificat segons la ruta que es vol mostrar.

El codi d'aquesta part del projecte es troba a l'Annex I.



### **DAO (Data Access Object):**

Tal i com el seu nom indica, totes les classes que estiguin dins d'aquesta carpeta estaran relacionades amb l'accés a la base de dades, és a dir, s'encarregaran de la manipulació de les dades.

- RutaDAO.java - Interfície on estan declarades les capçaleres de les funcions a implementar.
- RutaDAOImpl.java - Aquí estan implementades les funcions declarades a RutaDAO. Aquestes funcions són les següents:

- *public void GuardarRuta(Ruta r)* - Tal i com el seu nom indica, guarda la ruta que se li passa com a paràmetre d'entrada, la ruta 'r' i la guarda a la base de dades.
- *public void GuardarPosicio(PosicioRuta p)* - Guarda l'objecte PosicioRuta 'p' a la base de dades. Aquest objecte representa una posició real GPS en un instant de temps concret.
- *public Ruta ObtenirRuta(in idRuta)* - Aquesta funció permet obtenir les rutes emmagatzemades a la base de dades una per una, utilitzant el paràmetre 'idRuta' per determinar quina és la ruta a extreure.
- *public PosicioRuta ObtenirPosicio(int idRuta, int idPosicio)* - Igual que l'anterior però amb una posició en concret, que com es pot veure té dos identificadors per distingir entre les posicions de les rutes i les posicions dins d'una mateixa ruta.
- *public List<Ruta> obtenirTotesLesRutes()* - Retorna una llista amb totes les rutes emmagatzemades fins aleshores a la base de dades.
- *public void esborrarRuta(int idRuta)* - Esborra la ruta amb identificador 'idRuta' de la base de dades.

El codi d'aquesta part del projecte es troba a l'Annex II.

### **DOMAIN:**

A aquest bloc hi pertanyen aquelles classes que estan relacionades directament amb la base de dades, és a dir, les classes que generaran els objectes amb els que s'ha de treballar i les taules de la base de dades corresponents.

- PosicioRutaPk.java - Aquesta classe relaciona els identificadors de Ruta.java i PosicioRuta.java per tal de que es pugui assignar cadascun dels objectes PosicioRuta amb un objecte Ruta en concret. D'aquesta manera conté els atributs `_idRuta` i `_idPosicio`. També les funcions necessàries per poder manipular-los.



- Ruta.java - Conté els següents atributs que defineixen a la perfecció el que serà un objecte ruta: Identificador de la ruta, nom, longitud, temps d'inici, temps final, comentaris, temps total i el llistat de les posicions que formen la ruta amb la que s'està treballant. A més, incorpora tots els mètodes pertinents per poder manipular cadascun dels atributs anteriorment mencionats (Getters&Setters).
- PosicioRuta.java - Una posició GPS es pot definir amb els paràmetres següents: longitud, latitud i alçada. Per tant, aquests són uns dels atributs de la classe. D'altra banda, també són paràmetres que la descriuen, l'instant de temps en que es captura l'esmentada posició i, per suposat el seu identificador, que permetrà l'accés a les seves característiques. A més a més, igual que en el cas de la classe anterior (Ruta.java), hi ha la implementació dels corresponents Getters i Setters de cada atribut.

El codi d'aquesta part del projecte es troba a l'Annex III.

### **SERVICE:**

Els serveis que ofereix la Web estan implementats en aquest package.

- ServeiRutes.java - Interfície on es declaren les funcions de la classe ServeiRutaImpl.
- ServeiRutesImpl.java - Conté els següents mètodes:
  - *public void insertarNovaRuta(String nomRuta, String comentari, CommonsMultipartFile file)* - Tal i com indica el seu nom, permet inserir una nova ruta. Per dur a terme aquesta inserció cal donar com a paràmetres el seu nom, un comentari i el fitxer del qual s'obté la informació capturada per l'Smartphone i al que ha d'anar a buscar les dades.
  - *public Object[][] obtenirRutesTaula()* - Retorna tot el contingut de la taula de rutes.
  - *public Ruta obtenirRuta(int idRuta)* - Crida a la funció obtenirRuta del DAO.
  - *public void esborrarRuta(int idRuta)* - Crida a la funció esborrarRuta del DAO. A més a més, aquesta classe incorpora altres mètodes que han servit per fer càlculs tals com la distància total de la ruta, l'alçada de cada posició, etc.

El codi d'aquesta part del projecte es troba a l'Annex IV.

### 5.2.1.4.- Hibernate i Base de dades

Hibernate és una eina que permet facilitar l'accés a la base de dades. Funciona d'una manera semblant a Spring ja que es fan servir anotacions sobre el codi Java. Concretament aquestes anotacions s'han utilitzat a les classes del package "Domain". Hibernate permet que cada classe que esta anotada es pugui convertir en una taula a la base de dades, de manera que cada classe es correspon amb una taula i els atributs de cada classe columnes de la corresponent taula. Un exemple es pot veure a continuació on hi ha anotacions com "@Entity@Table" que indiquen que aquesta classe serà una taula a la base de dades, i l'anotació "@Id" que indica que l'atribut "\_idRuta" serà l'identificador de la taula "RUTA".

```
@Entity
@Table(name = "RUTA")
public class Ruta {
    //Atributs: - Propietats/paràmetres que conté la classe Ruta
    private int _idRuta;
    private String _NomRuta;
    private double _Longitud;
    private Date _IniciTemps;
    private Date _FinalTemps;
    private String _Comentaris;
    private String _TempsTotal;
    private List<PosicioRuta> _Posicions;
```

FIGURA 3.25:Classe ruta.java

+ Opciones		_idRuta	_Comentaris	_FinalTemps	_IniciTemps	_Longitud	_NomRuta	_TempsTotal
<input type="checkbox"/>	Editar	1	Vaig trobar bolets a mig camí.	2012-12-01 14:22:00	2012-12-01 14:00:00	1.45	Vinya Vella	0h 22min 0s
<input type="checkbox"/>	Editar	2	Primera ruta real	2013-01-24 17:30:19	2013-01-24 17:06:59	1.549	Vinya Vella Real	0h 23min 20s

FIGURA 3.26:Taula de la classe ruta a la base de dades

### 5.2.1.5.- Interacció amb Google Maps

Google Maps, és un servei gratuït de Google que ofereix imatges via satèl·lit de tot el globus terraqüi, combinades, en el cas d'alguns països, amb els mapes de les seves ciutats. A part això, ofereix una gama de serveis també gratuïts molt més àmplia.

Aquests serveis compten amb una API (*interfície de programació d'aplicacions*) molt ben elaborada amb la qual es poden realitzar aplicacions independents que utilitzin Google Maps.

Amb les API de Google Maps disponibles, nombrosos llocs Web han pogut proposar les seves maneres d'extreure les dades de Google i presentar-les en una pàgina Web pròpia a través de Scripts específics. D'aquesta manera cada lloc Web pot proposar un servei particular per aconseguir els seus objectius.

Tota la informació necessària per utilitzar aquestes prestacions es troba a la pàgina oficial de *Google Maps API Developers*, la que també ofereix tutorials per introduir a l'usuari a l'àmbit.

Així doncs, gràcies a aquesta eina, dins d'aquest treball s'han pogut inserir alguns dels Scripts que, com s'ha citat, Google cedeix als usuaris per tal d'introduir els seus mapes dins de Pàgina Web. Aquesta però, no ha sigut la única prestació utilitzada, sinó que també s'han utilitzat serveis que permeten dibuixar figures dins del mapa i altres que permeten capturar informació sobre totes les posicions del planeta així mateix com de longituds entre punts.

A continuació es presenten alguns dels Scripts utilitzats en aquest projecte i conjuntament una explicació clarificadora del funcionament.

```
// Aquest Script serveix per descarregar les llibreries de Google Maps.  
Gràcies a això, es pot utilitzar el mapa i totes les funcions que google permet  
  
<script type="text/javascript"  
    src="https://maps.googleapis.com/maps/api/js?sensor=false">  
  
</script>
```

FIGURA 3.27: Crida a les llibreries de Google Maps

```
// Aquí comença l'Script principal.
És dins d'on es realitzen totes les funcions relacionades amb el mapa
<script type="text/javascript">
  /* Variables utilitzades */
  var map; /* Representa el mapa que apareix actualment */
  var mapOptions; /* Opcions escollides al mapa */
  var poly; /* Línia que representa la ruta */
  var elevator; /* Servei de Google que permet consultar l'alçada
d'un punt */
  var posicions = []; /* Vector de posicions de la ruta */
  var i = 0; /* comptador */
  var contorn = new google.maps.LatLngBounds(); /* Defineix un
contorn maxim de la ruta per tal de que el zoom s'ajusti a la
mida de la ruta*/
```

*FIGURA 3.28: Inici Script principal. Declaració de variables*

Un cop introduït l'Script i les seves variables, s'exposen algunes de les funcions utilitzades. La totalitat de les funcions comentades es troba a l'Annex VI.

La funció que exposada a continuació, s'executa quan s'ha acabat de carregar la pàgina. S'encarrega d'inicialitzar el mapa així com el servei per sol·licitar alçades i pintar la ruta sobre el mapa.

```
function init()
{
  /* Es defineixen les opcions del mapa */
  mapOptions = {
    zoom: 18, /* Nivell inicial de zoom */
    mapTypeId: google.maps.MapTypeId.SATELLITE, /* Permet decidir el
tipus de mapa que es vol mostrar (normal, imatges satèl·lit...)
*/
    mapTypeControlOptions:{
      style: google.maps.MapTypeControlStyle.DROPDOWN_MENU
      /* Activa el control de canvi de tipus de canvi de mapa*/
    }
  };
  map = new google.maps.Map(document.getElementById("map"),mapOptions);
  /* Crea un mapa amb les opcions decidides */
  elevator = new google.maps.ElevationService();
  /* Activa el servei de consulta d'alçades */

  google.maps.event.addListenerOnce(map,"idle",pintaCoordenades);
  /* Aquesta funció, només s'executa una vegada en el moment que el mapa està
preparat "idle". S'encarrega de cridar la funció "pintaCoordenades" */
}
```

*FIGURA 3.29: Definició de les funcions del mapa*

Aquesta funció s'encarrega de pintar sobre el mapa tots els punts que ha rebut des del servidor i a partir d'això crear una línia que els uneixi.

```
function pintaCoordenades(){
    var polyOptions = { /*Estableix les opcions de la línia que es pintarà */
        strokeColor: '#ffff1e', /* Defineix color de la línia */
        strokeOpacity: 1.0, /* Defineix transparència de la línia */
        strokeWeight: 3 /*Defineix nombre pixels d'amplada de línia */
    };
    poly = new google.maps.Polyline(polyOptions); /* Genera una línia que es
pot insertar sobre el mapa de google */
    poly.setMap(map);
    <c:forEach var="posicio" items="{ruta._Posicions}">
/* Bucle que recorre totes les posicions que el servidor ha enviat. Aquestes
estan emmagatzemades a la variable ruta_Posicions i es va consultant da cada
posició les dades que es contenen, com l'instant al que s'ha pres la mesura, la
latitud i la longitud. També es crida a la funció getElevation() que
s'encarrega de fer la petició del valor de l'alçada en un punt concret*/
        var instantPosicio = "${posicio._instantPosicio}";
        lat = <c:out value="{posicio._latitud}"/>;
        lng = <c:out value="{posicio._longitud}"/>;
        var latLng = new google.maps.LatLng(lat, lng);
        var path = poly.getPath();
        path.push(latLng);
        var posicio = new Object();
        posicio.lat = lat;
        posicio.lng = lng;
        posicio.instant = instantPosicio;
        posicio.alçada = 0;
        posicions.push(posicio);
        getElevation(latLng);
        contorn.extend(latLng);
    </c:forEach>
    map.fitBounds(contorn);
};
```

FIGURA 3.30: Inserció de punts i línies dins el mapa

La funció posterior fa una petició al servidor del valor de l'alçada en un punt concret:

```
function getElevation(latLng) {
    var locations = [];
    locations.push(latLng);

    var positionalRequest = {
        'locations': locations
    };
    elevator.getElevationForLocations(positionalRequest, getResultats);
};
```

FIGURA 3.31: Petició d'alçades

## 5.2.2.- Aplicació Android

L'objectiu d'aquesta aplicació és aconseguir generar fitxers de rutes mitjançant les capacitats que aporten els Smartphones, com poden ser els receptors GPS.

### 5.2.2.1.- Definició del funcionament

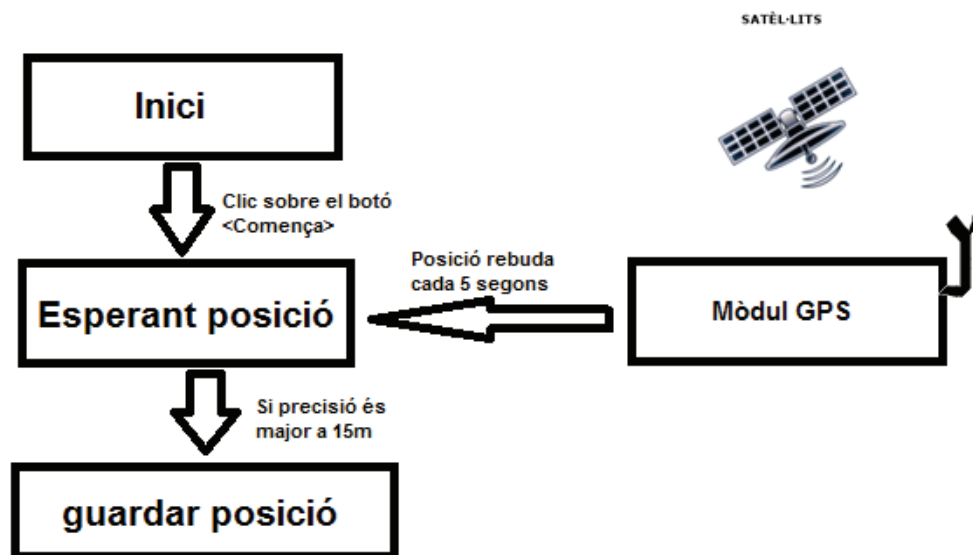


FIGURA 3.32: Diagrama de flux de l'aplicació Android

L'aplicació comença a una pantalla inicial (inici) on apareix un botó anomenat '<Comença>' i dos textos a la part superior de la pantalla que informen de l'estat del GPS i la quantitat de mostres de posicions capturades. Un cop es clica a comença, aquest botó desapareix i n'apareix un anomenat '<atura>' que permet finalitzar la ruta. Fins aleshores el programa inicia el mòdul GPS amb un temps mínim de resposta a 5 segons per tal d'obtenir mesures dels satèl·lits. Quan comença a rebre dades, es comprova que la mesura rebuda tingui una precisió menor a 15m de desviació i la guarda a la següent fila del fitxer, la resta de mesures es descarten.

En polsar sobre el botó atura (aquest apareix només un cop s'ha iniciat la ruta), l'aplicació deixa de prendre mesures. Així finalitza el procés de creació del fitxer.

A continuació es presenta una captura de pantalla:



FIGURA 3.33: Visualització de l'aplicació Android en els seus dos únics estats

#### 5.2.2.2.- Format del fitxer

Per escollir el tipus de dades que conté el fitxer, és molt rellevant estar en coneixement de les dades que poden proporcionar les llibreries d'Android. Així doncs, algunes de les dades que es poden aconseguir són les coordenades GPS (longitud, latitud) i els instants de temps. Tot i que no són les úniques dades que és capaç d'aconseguir quant a posicionament, aquestes són suficients per aconseguir altres paràmetres tals com alçades i distàncies, gràcies a les prestacions de Google Maps anteriorment citades.

El fitxer per tant, conté tres característiques de cada posició d'una ruta; instant de temps, longitud i latitud.

La organització d'aquesta informació dins del fitxer és la següent:

- Cada fila és una posició de la ruta. Un cop l'aplicació emmagatzema la informació d'una posició, salta de línia per començar-ne una de nova.
- Cada línia està formada pels tres paràmetres citats; es comença amb l'instant de temps, que es guarda amb un format concret de la següent manera: <yyyy/MM/dd HH:mm:ss>.
- Darrerament hi ha una coma que separa l'instant de temps del paràmetre que té al darrera; la latitud.
- Per finalitzar hi ha una altra coma i l'últim paràmetre, la longitud.
- No hi ha cap mena de separació entre paràmetres a part de la coma “,” citada.

D'aquesta manera es troba el format del fitxer compatible amb els dos aparells; mòbil i ordinador.

Aquí se'n exposa una mostra:

```
yyyy/MM/dd HH:mm:ss,lat1,long1
yyyy/MM/dd HH:mm:ss,lat2,long2
yyyy/MM/dd HH:mm:ss,lat3,long3
...
```

*FIGURA 3.34:Format fitxer generat*

### 5.3.- Verificació

La verificació del funcionament és un dels passos més importants en l'elaboració d'aplicacions de qualsevol tipus. És una tasca que es realitza de manera iterativa comprovant en cada iteració que les noves línies de codi i funcions implementades es comporten tal i com s'espera. D'aquesta manera es van solucionant els petits problemes que poden anar sorgint poc a poc i restringint on es poden trobar els errors sorgits.

Per tal de dur a terme la verificació resulta molt útil l'ús de la eina Debugger que proporciona un control de l'execució del programa línia a línia i permetent així trobar en quina d'elles es produeix l'error. Això permet començar a indagar sobre les causes de l'error per tal de poder solucionar-lo.

Tot i anar provant pas a pas cada part del codi, l'objectiu és que el conjunt global funcioni correctament.



### 5.3.1.- Ruta de prova - Fitxer de dades manual

Com que primerament s'ha implementat l'aplicació web, ha estat necessari generar manualment un fitxer de text que simula un possible fitxer obtingut amb l'aplicació mòbil. Per tal de que sigui el més realista possible s'han agafat unes quantes posicions GPS començant des de Guardiola de Berguedà simulant una ruta fins a la Font de la Vinya Vella. A cada posició se li ha assignat aproximadament un instant de temps de manera que el fitxer ja queda complet.

Un cop generat el fitxer s'ha procedit amb el test del sistema. Per fer-ho s'ha accedit a la pàgina Principal i s'ha omplert el formulari amb el nom de la ruta, un comentari i s'ha carregat el fitxer per tal de pujar-lo a servidor. S'ha comprovat que al pulsar el boto "Desa" s'executen al servidor les comandes d'Hibernate que s'encarreguen d'emmagatzemar la ruta enviada així com les seves posicions. Després d'això es comprova que aquestes dades s'hagin introduït a la taula correctament i que l'enllaç cap al detall de la ruta funciona i es mostra el mapa amb la ruta dibuixada, així com els valors calculats afegits: l'alçada màxima, mínima i el temps de tota la ruta.



*FIGURA 3.35: Mapa amb la ruta de prova i els seus respectius punts*

El fitxer que genera aquesta ruta es pot veure a l'Annex XI.

### 5.3.2.- Ruta real - La primera ruta

Un cop s'ha finalitzat la programació de la Pàgina Web i de l'Aplicació Android, és hora de provar la seva eficàcia. Per fer-ho, s'ha realitzat la mateixa ruta que la que es va utilitzar per fer la prova Web, i així poder-ne comparar resultats.

Un cop posada en pràctica l'eficàcia del conjunt complet del programa, s'ha obtingut el següent resultat:



*FIGURA 3.36: Mapa amb la ruta real i els seus respectius punts*

El fitxer que genera aquesta ruta es pot veure a l'Annex XI.

Com es pot observar, la ruta real té molta més precisió que la proposada anteriorment, i les posicions s'ajusten bastant al camí real, la qual cosa verifica que la informació plasmada no és una fal·làcia.

Des d'aquesta perspectiva, dona la sensació que no hi hagi línies entre punt i punt, però en apropar el mapa, s'observa el següent:

Aquesta petita separació es deguda a que s'està buscant la major precisió possible.



*FIGURA 3.37: Ampliació del zoon de la ruta*

## **6.- CONCLUSIONS**

### **6.1.- Reflexió del projecte**

En primer lloc, citar que gràcies al gran esforç realitzat, s'han pogut aconseguir els objectius inicials. L'aplicació creada funciona tal i com es desitja i, a més, s'han pogut afegir extres que la fan més pràctica i agradable a la vista de l'usuari.

Com el treball s'ha pogut dur a terme de manera gratificant, la meta personal també s'ha complert. Ha sigut possible aprendre els conceptes bàsics de programació, i no només això, sinó que també s'han pogut dur a la pràctica per mostrar la seva exitosa aprensió.

Quant al treball en sí, les reflexions que es poden extreure un cop ja s'ha finalitzat són moltes.

En primer lloc, pot fer-se una reflexió molt clara i evident; per senzill que pugui semblar un projecte en el qual hi intervé la programació, les hores de dedicació són moltes sempre més de les esperades. Qualsevol cosa, sense importar-ne el grau de dificultat, pot presentar errors que n'impedeixin el bon funcionament, fet que suposa molt de temps en dedicació.

Una altra conclusió a la qual s'ha arribat fent aquest projecte és que les circumstàncies requereixen una adaptació a les necessitats durant el transcurs del treball. Això significa que tot i partint d'una idea principal molt clara i precisa, finalment el treball mai acaba sent com s'esperava en un principi.

### **6.2.- Línies de continuïtat**

Un cop finalitzat el projecte i ja coneixent tot el que aquest aporta, es podria fer una llista de millores que podria incloure com a projecte futur.

Quant a la ruta:

- Acotació del temps entre posicions per aconseguir rutes més precises.
- Aconseguir més dades de la ruta, tals com velocitat mitjana de la ruta, velocitat instantània a cada posició, desnivell màxim i mínim de la ruta, etc.
- Poder posar marcadors i escriure un comentari a les posicions desitjades durant la ruta.
- Possibilitat de parar la ruta en punts entremitjos (parades puntuals a menjar, descansar), no només al inici i final.

En el cas de projectes amb visions de comercialització:

Quant a millores de sistema:

- Seria necessari contractar un servidor remot des del qual l'usuari a través d'Internet podria accedir a la pàgina Web.
- Cada usuari necessitaria una pàgina pròpia a la que només ell hi tingués accés. Així doncs, seria necessari crear una nova pàgina formulari amb la funció de permetre als usuaris ser registrats o accedir a la pàgina que ja s'han apropiat.

De cara a projectes més elaborats:

- Convertir de pàgina Web a aplicació Android per tal de visualitzar i realitzar totes les funcions possibles sense necessitat d'utilitzar l'ordinador, és a dir, tot tindria lloc al mateix Smartphone.
- En aquest cas, l'aplicació es podria introduir al *Play Store* del sistema Android. Per aconseguir-la doncs, s'hauria d'anar a la botiga intel·ligent del sistema (Play Store) i descarregar-la.

Aquests projectes però, suposarien una inversió de capital (tant el contracte d'un servidor remot, com el d'inserir una nova aplicació al Play Store) pel que s'hauria d'estudiar la idea amb deteniment i determinar si hi ha algun camí per fer viables les propostes.



## 7.- BIBLIOGRAFIA

Pàgines web:

- Manuals d'usuari, tutorials i adreces de descàrrega de llibreries:
  - <https://developers.google.com/maps/>
  - <http://economy.org.uk/gmap/>
  - <http://docs.oracle.com/javase/tutorial/>
  - <http://www.tutorialspoint.com/java/index.htm>
  - <http://www.w3schools.com/js/default.asp>
  - <http://www.tizag.com/javascript/>
  - <http://www.mysqltutorial.org/>
  - <http://www.tizag.com/mysqlTutorial/>
- Pàgines de recerca d'informació conceptual:
  - [http://es.wikipedia.org/wiki/Sistema\\_de\\_posicionamiento\\_global](http://es.wikipedia.org/wiki/Sistema_de_posicionamiento_global)
  - [http://www.gredosmil.es/index\\_archivos/gps.htm](http://www.gredosmil.es/index_archivos/gps.htm)
  - [http://es.wikipedia.org/wiki/Direcci%C3%B3n\\_IP](http://es.wikipedia.org/wiki/Direcci%C3%B3n_IP)
  - <http://www.monografias.com/trabajos34/base-de-datos/base-de-datos.shtml>
  - <http://es.wikipedia.org/wiki/Localhost>
  - [http://en.wikipedia.org/wiki/Hibernate\\_\(Java\)](http://en.wikipedia.org/wiki/Hibernate_(Java))
  - <http://www.hibernate.org/>
  - <http://www.apache.org/>

Llibres:

- A.PEÑA BASURTO, Marco; CELA ESPÍN, José M. *Introducción a la programación*. Barcelona: Edicions UPC, Setembre de 2000.
- BELMONTE FERNÁNDEZ, Oscar. *Introducción al lenguaje de programación Java*. Barcelona: CEPFC, 2004.



# *ANNEX*

## ***ÍNDEX ANNEX***

CODI COMPLET DE LA PROGRAMACIÓ DE LA PÀGINA WEB.....	II
Annex I - Package controller.....	III
Annex II - Package dao .....	VI
Annex III - Package domain.....	IX
Annex IV - Package service.....	XIII
Annex V - Fulla d'estil CSS.....	XVII
Annex VI - Pàgines HTML.....	XIX
CODI COMPLET DE LA PROGRAMACIÓ DE L'APLICACIÓ ANDROID .....	XXX
Annex VII - Línies de codi Android.....	XXXI
ESBOSSOS DUTS A TERME AMB EL PROGRAMA BALSAMIQ MOCKUPS .....	XXXVI
Annex VIII - Pàgina web.....	XXXVI
Annex IX - Aplicació android .....	XXXIX
FITXERS OBTINGUTS AMB L'EXECUCIÓ D'UNA RUTA.....	XL
Annex X - Fitxer de prova: ruta vinya vella .....	XL
Annex XI - Fitxer real: ruta vinya vella .....	XLI



## ***CODI COMPLET DE LA PROGRAMACIÓ DE LA PÀGINA WEB***

Seguidament s'adjunta la totalitat de codi necessari per al funcionament del programa Web *<VisualitzadorWebDeRutes>*.

El contingut està ordenat per packages tal i com s'explica a al punt *5.2.1.3 Descripció de les classes* inclòs dins del treball.

## *Annex I - Package controller*

BenvingudaController.java

```
package controller;

import org.springframework.stereotype.Controller;

@Controller
public class BenvingudaController {

    @RequestMapping(value = "/benvinguda", method = RequestMethod.GET)
    public String getBenvinguda(){
        System.out.println("Benvinguda!");
        return "benvinguda";
    }
}
```

## DetallRutaController.java

```
package controller;

import java.text.SimpleDateFormat;

@Controller
public class DetallRutaController {

    @Autowired
    private ServeiRutes serveiRutes;

    @RequestMapping(value = "/detallRuta/{idRuta}", method = RequestMethod.GET)
    public String getRutesEmmagatzemades(Model model, @PathVariable int idRuta){
        System.out.println("Detall Ruta!");
        System.out.println(idRuta);
        Ruta r = serveiRutes.obtenirRuta(idRuta);
        model.addAttribute("ruta",r);
        return "detallRuta";
    }
}
```

## PrincipalController.java

```
package controller;

import org.springframework.beans.factory.annotation.Autowired;

@Controller
public class PrincipalController {

    @Autowired
    private ServeiRutes serveiRutes;

    @RequestMapping(value = "/principal", method = RequestMethod.GET)
    public String getPrincipal(){
        System.out.println("Principal!");
        return "principal";
    }

    @RequestMapping(value = "/upload", method = RequestMethod.POST)
    public String guardarNovaRuta(@RequestParam(value="nomRuta") String nomRuta,
    @RequestParam(value="comentari") String comentari,
    @RequestParam(value="data") CommonsMultipartFile file){
        System.out.println("Upload!");
        System.out.println(nomRuta);
        System.out.println(comentari);
        System.out.println(file.getOriginalFilename());

        //Funcio guardar nova ruta i llegir fitxer amb les posicions
        serveiRutes.insertarNovaRuta(nomRuta, comentari, file);

        return "redirect:/principal.html";
    }

    @RequestMapping(value = "/rutesEmmagatzemades", method = RequestMethod.GET)
    public @ResponseBody Object[][] getRutesEmmagatzemades(){
        System.out.println("Rutes Emmagatzemades!");
        return serveiRutes.obtenirRutesTaula();
    }

    @RequestMapping(value = "/esborrar", method = RequestMethod.POST)
    public String esborrarRuta(@RequestParam(value="id") int id){
        System.out.println("Esborra la ruta!");
        System.out.println(id);
        serveiRutes.esborrarRuta(id);
        return null;
    }
}
```

RutaDAO.java

```
package dao; // Data Acces Object

import java.util.List;

// Declaració dels mètodes que s'utilitzaran per interaccionar amb la base de
dades
// Una Interface és l'encarregada de declarar el nom de les funcions i no de la
seva implementació.

public interface RutaDAO {
    // Guarda una nova ruta r a la base de dades.
    public void GuardarRuta(Ruta r);
    /** Guarda una nova posicioRuta p a la base de dades. Els identificadors de
l'objecte posicioRuta
han d'estar generats prèviament
@param p objecte posicio ruta
@return void */
    public void GuardarPosicio(PosicioRuta p);
    /** Obté una ruta de la base de dades que té com a identificador idRuta
@param idRuta Nombre enter identificador de la ruta que es vol obtenir
@return Objecte Ruta de la base de dades que té com a identificador idRuta.
Si no existeix una ruta amb aquest
identificador a la base de dades, retorna null.
*/
    public Ruta ObtenirRuta(int idRuta);
    /** Obté una posició de la base de dades que està identificada per idPosicio
i idRuta
* @param idRuta Nombre enter Identificador de la ruta a la qual pertany
aquesta posició.
* @param idPosicio Nombre enter que distingeix les diferents posicions
d'una mateixa ruta.
* @return Objecte PosicioRuta de la base de dades
*/
    public PosicioRuta ObtenirPosicio(int idRuta, int idPosicio);

    public List<Ruta> obtenirTotesLesRutes();

    public void esborrarRuta(int idRuta);
}
```

## RutaDAOImpl.java

```
package dao; // Data Acces Object

import java.util.List;

@Repository
public class RutaDAOImpl implements RutaDAO {

    private HibernateTemplate _hibernateTemplate;

    @Autowired
    public void setSessionFactory(SessionFactory sessionFactory) {
        this._hibernateTemplate = new HibernateTemplate(sessionFactory);
    }

    @Override
    public void GuardarRuta(Ruta r) {
        int ultimId = obtenerUltimIdentificador();

        r.set_idRuta(ultimId);

        List<PosicioRuta> posiciones = r.get_Posicions();

        int index = 0;
        int numPosicions = posiciones.size();
        for (index=0; index<numPosicions; index++){
            PosicioRuta p = posiciones.get(index);
            PosicioRutaPk Pk = new PosicioRutaPk();
            Pk.set_idRuta(ultimId);
            Pk.set_idPosicio(index+1);
            p.set_idPosicioPk(Pk);
        }

        _hibernateTemplate.save(r);
    }

    private int obtenerUltimIdentificador(){
        int ultimId = 1;

        List idList = _hibernateTemplate.find("select max(r._idRuta) from
Ruta r");

        Object id = idList.get(0);

        if(id == null){
            return ultimId;
        }else{
            ultimId = (int)id;
            return ultimId+1;
        }
    }

    @Override
    public void GuardarPosicio(PosicioRuta p) {
        _hibernateTemplate.save(p);
    }
}
```

```

@Override
public Ruta ObtenerRuta(int idRuta) {
    return _hibernateTemplate.get(Ruta.class, idRuta);
}

@Override
public PosicioRuta ObtenerPosicio(int idRuta, int idPosicio) {
    //Crea un objecte del tipus PosicioRutaPk
    PosicioRutaPk posicioRutaPk = new PosicioRutaPk();
    //Inserta idPosicio a l'atribut de l'objecte _idPosicio
    posicioRutaPk.set_idPosicio(idPosicio);
    //Inserta idRuta a l'atribut de l'objecte _idRuta
    posicioRutaPk.set_idRuta(idRuta);

    return _hibernateTemplate.get(PosicioRuta.class, posicioRutaPk);
}

public List<Ruta> obtenirTotesLesRutes(){
    return _hibernateTemplate.loadAll(Ruta.class);
}

@Override
public void esborrarRuta(int idRuta) {
    Ruta r = _hibernateTemplate.get(Ruta.class, idRuta);
    _hibernateTemplate.deleteAll(r.get_Posicions());
    _hibernateTemplate.delete(r);
}

}

```

Ruta.java

```
package domain;

import java.util.Date;

@Entity
@Table(name = "RUTA") //Taula anomenada Ruta (a la base de dades) on es guardaran
les diferents característiques de cada ruta

//Nova classe anomenada Ruta. Caracterització d'una ruta
public class Ruta {

    //Atributs: - Propietats/paràmetres que conté la classe Ruta
    private int _idRuta; //Nombre encarregat d'identificar la ruta amb la que es
treballa
    private String _NomRuta; // Paràmetre que contindrà el nom de la ruta
escollit per l'usuari
    private double _Longitud; // Paràmetre que contindrà la longitud total de la
ruta
    private Date _IniciTemps; // Paràmetre que contindrà la data de l'instant en
que va començar la ruta
    private Date _FinalTemps; // Paràmetre que contindrà el data de l'instant en
que va acabar la ruta
    private String _Comentaris; // Paràmetre que contindrà un comentari de la
ruta escollit per l'usuari
    private String _TempsTotal; // Paràmetre que contindrà el temps total en el
que s'ha efectuat la ruta
    private List<PosicioRuta> _Posicions; // Artibut format per una llista
d'objectes de la classe PosicioRuta

    //Mètodes - Gets i Sets:
    @Id
    public int get_idRuta() {
        return _idRuta; // Funció que permet obtenir el paràmetre
d'identificació de la ruta
    }
    public void set_idRuta(int _idRuta) { // Funció que permet guardar el
paràmetre d'identificació de la ruta
        this._idRuta = _idRuta;
    }
    public String get_NomRuta() {
        return _NomRuta;
    }
    public void set_NomRuta(String _NomRuta) {
        this._NomRuta = _NomRuta;
    }
    public double get_Longitud() {
        return _Longitud;
    }
    public void set_Longitud(double _Longitud) {
        this._Longitud = _Longitud;
    }
    public Date get_IniciTemps() {
        return _IniciTemps;
    }
}
```



```

    }
    public void set_IniciTemps(Date _IniciTemps) {
        this._IniciTemps = _IniciTemps;
    }
    public Date get_FinalTemps() {
        return _FinalTemps;
    }
    public void set_FinalTemps(Date _FinalTemps) {
        this._FinalTemps = _FinalTemps;
    }
    public String get_Comentaris() {
        return _Comentaris;
    }
    public void set_Comentaris(String _Comentaris) {
        this._Comentaris = _Comentaris;
    }
}

public String get_TempsTotal() {
    return _TempsTotal;
}
public void set_TempsTotal(String _TempsTotal) {
    this._TempsTotal = _TempsTotal;
}
//Crea una relació entre la taula Ruta i la PosicioRuta amb relació
OneToMany.
@OneToMany(cascade={CascadeType.ALL}, fetch=FetchType.EAGER)
@JoinColumn(name="_idRuta", referencedColumnName="_idRuta")
public List<PosicioRuta> get_Posicions() {
    return _Posicions;
}
public void set_Posicions(List<PosicioRuta> _Posicions) {
    this._Posicions = _Posicions;
}
}
}

```

## PosicioRuta.java

```
package domain;

import java.util.Date;

@Entity
@Table(name = "POSICIORUTA") // Taula on seran emmagatzemades a la base de dades
totes les posicions que s'utilitzin

//Classe PosicioRuta "Cada una de les posicions que conté una ruta pot definir-se
mitjançant aquesta classe.
public class PosicioRuta {
    //Atributs - paràmetres que conté la classe PosicioRuta:
    private PosicioRutaPk _idPosicioPk; //Nombre encarregat d'identificar cada
posició d'una ruta
    private double _longitud; // Paràmetre que contindrà la longitud de la
posició
    private double _latitud; // Paràmetre que contindrà la latitud de la
posició
    private double _alcada; // Paràmetre que contindrà l'alçada de la posició
    private Date _instantPosicio; // Paràmetre que contindrà l'instant de temps
en que es passa per la posició

    //Mètodes - Gets i sets:
    @EmbeddedId
    public PosicioRutaPk get_idPosicioPk() {
        return _idPosicioPk;
    }
    public void set_idPosicioPk(PosicioRutaPk _idPosicioPk) {
        this._idPosicioPk = _idPosicioPk;
    }
    public double get_longitud() {
        return _longitud;
    }
    public void set_longitud(double _longitud) {
        this._longitud = _longitud;
    }
    public double get_latitud() {
        return _latitud;
    }
    public void set_latitud(double _latitud) {
        this._latitud = _latitud;
    }
    public double get_alcada() {
        return _alcada;
    }
    public void set_alcada(double _alcada) {
        this._alcada = _alcada;
    }
    public Date get_instantPosicio() {
        return _instantPosicio;
    }
    public void set_instantPosicio(Date _instantPosicio) {
        this._instantPosicio = _instantPosicio;
    }
}
}
```

## PosicioRutaPK.java

```
package domain;

import java.io.Serializable;

// Classe que permet crear un objecte Primary Key de posicioRuta, la qual té dos
// identificadors.
// S'utilitzarà a Hibernate com a clau identificadora múltiple de la taula
// posicioRuta i servirà per relacionar-se
// amb la taula Ruta corresponent. Tipus de relació entre taules: "One to Many".

@Embeddable
public class PosicioRutaPk implements Serializable{
    //Atributs:
    private int _idPosicio;
    private int _idRuta;

    //Gets i sets:
    public int get_idPosicio() {
        return _idPosicio;
    }
    public void set_idPosicio(int _idPosicio) {
        this._idPosicio = _idPosicio;
    }
    public int get_idRuta() {
        return _idRuta;
    }
    public void set_idRuta(int _idRuta) {
        this._idRuta = _idRuta;
    }
}
```

ServeiRutes.java

```
package service;

import org.springframework.web.multipart.commons.CommonsMultipartFile;

public interface ServeiRutes {

    public void insertarNovaRuta(String nomRuta, String comentari,
CommonsMultipartFile file);

    public Object[][] obtenirRutesTaula();

    public Ruta obtenirRuta(int idRuta);

    public void esborrarRuta(int idRuta);

}
```

## ServeiRutesImpl.java

```
package service;

import java.io.BufferedReader;

@Service
public class ServeiRutesImpl implements ServeiRutes{

    @Autowired
    private RutaDAO rutaDAO;

    @Override
    public void insertarNovaRuta(String nomRuta, String comentari,
        CommonsMultipartFile file) {
        Ruta r = new Ruta();
        r.set_Comentaris(comentari);
        r.set_NomRuta(nomRuta);
        List<PosicioRuta> posicions = new ArrayList<PosicioRuta>();

        InputStream iSt;
        DataInputStream dataiSt;
        InputStreamReader iStReader;
        BufferedReader br;

        try {
            iSt = file.getInputStream();
            dataiSt = new DataInputStream(iSt);
            iStReader = new InputStreamReader(dataiSt);
            br = new BufferedReader(iStReader);
            String line = "";
            while( (line = br.readLine()) != null)
            {
                PosicioRuta p = new PosicioRuta();
                String[] columnes = line.split(",");
                SimpleDateFormat sdf = new SimpleDateFormat("yyyy/MM/dd
HH:mm:ss");

                p.set_instantPosicio(sdf.parse(columnes[0]));
                p.set_latitud(Double.parseDouble(columnes[1]));
                p.set_longitud(Double.parseDouble(columnes[2]));
                posicions.add(p);
            }
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (ParseException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        //Buscar la distància total a partir de cada segment de la ruta
        double distanciaTotal = 0;
        double distanciaParc = 0;
        int segments = posicions.size()-1;
        for (int i=0;i<segments;i++){
            PosicioRuta pact = posicions.get(i);
            PosicioRuta ppos = posicions.get(i+1);
```

```

                distanciaParc = distancia(pact.get_latitud(),
pact.get_longitud(), ppos.get_latitud(), ppos.get_longitud());

                distanciaTotal = distanciaTotal+distanciaParc;
            }

            distanciaTotal = distanciaTotal*1000; //Passem a metres
            distanciaTotal = Math.round(distanciaTotal); // Arrodoneix a la
unitat

            r.set_Longitud(distanciaTotal/1000);

            //Omplir els atributs iniciTemps i finalTemps de la classe ruta
            PosicioRuta p0 = posicions.get(0);
            r.set_IniciTemps(p0.get_instantPosicio());

            PosicioRuta pf = posicions.get(posicions.size()-1);
            r.set_FinalTemps(pf.get_instantPosicio());

            //Càlcul del temps total
            Date f = r.get_FinalTemps();
            Date o = r.get_IniciTemps();

            long TempsFinal = f.getTime();
            long TempsInicial = o.getTime();
            double DifTemps = TempsFinal-TempsInicial;

            DifTemps = DifTemps/1000; //Temps en segons
            int h;
            int m;
            int s;
            h = (int) Math.round(DifTemps/3600);
            double min0 = ((DifTemps/3600)-h)*60;
            m = (int) Math.round(min0);
            s = (int) Math.round((min0-m)*60);

            r.set_TempsTotal(h+"h "+m+"min "+s+"s");

            r.set_Posicions(posicions);
            rutaDAO.GuardarRuta(r);
        }

        public double convertirARadians(double graus){
            double rad = Math.PI*graus/180;
            return rad;
        }

        public double distancia(double lat1, double lng1, double lat2, double lng2){

            double R = 6371; // km
            lat1 = convertirARadians(lat1);
            lat2 = convertirARadians(lat2);
            lng1 = convertirARadians(lng1);
            lng2 = convertirARadians(lng2);
            double dLat = (lat2-lat1);
            double dLon = (lng2-lng1);

            double a = Math.sin(dLat/2) * Math.sin(dLon/2) +

```

```

        Math.sin(dLon/2) * Math.sin(dLon/2) * Math.cos(lat1) *
Math.cos(lat2);
        double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
        double d = R * c;

        return d;
    }

    @Override
    public Object[][] obtenirRutesTaula() {
        List<Ruta> llistaRutes = rutaDAO.obtenirTotesLesRutes();
        int numeroDeRutes = llistaRutes.size();
        Object[][] contingutTaula = new Object[numeroDeRutes][7];
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");

        int i=0;

        for (i=0;i<numeroDeRutes;i++){
            Ruta r = llistaRutes.get(i);

            contingutTaula[i][0] = r.get_idRuta(); //Columna0 = id
            contingutTaula[i][1] = "<a
href=\"detallRuta/\"+r.get_idRuta()+\".html\">"+r.get_NomRuta() +\"</a>"; //Columna1
= Nom de la ruta
            contingutTaula[i][2] =
sdf.format(r.get_IniciTemps()); //Columna2 = Instant inicial
            contingutTaula[i][3] =
sdf.format(r.get_FinalTemps()); //Columna3 = Instant final
            contingutTaula[i][4] = r.get_Longitud(); //Columna4 = Longitud
            contingutTaula[i][5] = r.get_Comentaris(); //Columna5 =
Comentari
            contingutTaula[i][6] = "<input type=\"button\"
id=\"\"+r.get_idRuta()+\"\" value=\"X\" onClick=\"esborrarRuta(this.id)\">";
//Columna 6 = Botó esborrar ruta
        }

        return contingutTaula;
    }

    @Override
    public Ruta obtenirRuta(int idRuta) {
        Ruta r = rutaDAO.ObtenirRuta(idRuta);
        return r;
    }

    @Override
    public void esborrarRuta(int idRuta) {
        rutaDAO.esborrarRuta(idRuta);
    }
}

```

Style.css

```
/* Estil general de la Pàgina Web */
body {
  padding-left: 11em;
  font-family: "Arial", Georgia,
    Times, serif;
  color: #000000;
  background-color: #aec469;
  width: 1100px;
}

ul.navbar {
  list-style-type: none;
  padding: 0;
  margin: 0;
  position: absolute;
  top: 2em;
  left: 1em;
  width: 9em;
  padding-top: 800px;
  padding-left: 1000px;
}

h1 {
  font-family: "Arial", Helvetica, Geneva, Arial,
    SunSans-Regular, sans-serif }

ul.navbar li {
  background: white;
  margin: 0.5em 0;
  padding: 0.3em;
  border-right: 1em solid black }

ul.navbar a {
  text-decoration: none }

a:Link {
  color: green }

a:visited {
  color: green }

address {
  margin-top: 1em;
  padding-top: 1em;
  border-top: thin dotted }

/* Botó <Desa> a principal.html */
#Boton1 {
  width: 200px;
  padding: 15px;
  background: #7a9b16;
}
```



```

/* Imatges */
#Logo{
    background-image: url('../images/Logo.jpg');
    width: 1000px;
    height: 76px;
    border: 2px;
    vertical-align: 3px;
    margin-top: 40px;
    margin-bottom: 70px;
}

#imagen{
    background-image: url('../images/imagen.jpg');
    width: 1000px;
    height: 250px;
    border: 2px;
    vertical-align: 3px;
    margin-top: 60px;
    margin-bottom: 50px;
}

#imaginicial{
    background-image: url('../images/imaginicial.jpg');
    width: 1000px;
    height: 615px;
    border: 2px;
    vertical-align: 3px;
    margin-top: 40px;
    margin-bottom: 70px;
}

/*Formulari per incloure noves rutes a principal.html */
#insertarNovesRutes{
    width: 100px;
    height: 200px;
    font-family: Arial;
    padding-top: 1em;
    font-size: 20px;
    position: absolute;
}

/* Mapa de Google Maps insertat a detallRuta.html */
#map{
    height: 400px;
    width: 1000px;
    padding-top: 200px;
}

/* Taula continguda a detallRuta.html que mostra les dades específiques de la ruta
*/
#taula2{
    font-family: Arial;
    background: white;
    margin: 4em 0;
    padding: 0.8em;
}

```

## Annex VI - Pàgines HTML

benvinguda.jsp

```
border-right: 0.8em solid black;
font-size: 20px;
}
```

Benvinguda.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-
1">
    <title>Benvinguda</title> <!-- Nom de la pàgina -->
    <link rel="stylesheet" href="css/Style.css"> <!-- Utilitza com
a patró d'estil el fitxer Style.css -->
  </head>
  <body>
    <a title="Clica sobre la imatge per continuar."
href="principal.html"> <!-- Comentari en passar el cursor sobre la imatge -->
    <div id="imaginicial"></div> <!-- Imatge -->
    </a>
    <address> Clica sobre la imatge per continuar.</address> <!--
Comentari de peu de pàgina -->
  </body>
</html>
```

## principal.jsp

```
<%@taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-
1">
        <title>EL MÓN EN MINIATURA</title> <!--Títol de la pàgina -->
        <link rel="stylesheet" href="css/Style.css"> <!-- Utilitza com a
patró d'estil el fitxer Style.css -->

        <script type="text/javascript"> /* Script que permet esborrar una
ruta de la taula dataTable */
            function uploadFile()
            {
                document.forms["upload"].submit();
            }
            function esborrarRuta(idBoto){
                $.post("esborrar.html",{id:idBoto},
                    function(){

document.location.href="/VisualitzadorWebDeRutes/principal.html";
                    });
            }
        </script>

        <!-- Llibreries per utilitzar la taula -->
        <script type="text/javascript" src="js/jquery/jquery-1.7.1.min.js">
</script>
        <script type="text/javascript" src="js/datatables/jquery.dataTables.js">
</script>
        <link rel="stylesheet" href="css/demo_table_jui.css" type="text/css" />
        <link rel="stylesheet" href="css/ui-darkness/jquery-ui-1.8.17.custom.css"
type="text/css" />

        <!-- Aquest codi, s'encarrega de fer la petició al servidor de les dades
necessàries per omplir la taula. S'executa automàticament en el moment en que el
client ja ha descarregat la pàgina (gràcies a $(document).ready) i envia una
petició a la url <rutesEmmagatzemades.html>. La resposta del servidor són les
dades que han estat generades adequadament a les característiques de la taula. -
->

        <script type="text/javascript">
            $(document).ready(
                function() {
                    $.getJSON("rutesEmmagatzemades.html",function(data) {
                        if (data) {
                            $('#rutesEmmagatzemades').html('<div
class="dataTables_wrapper" id="example_wrapper"> <table
cellpadding="0" cellspacing="0" border="0"
class="display" id="example"></table></div>');
                            $('#example').dataTable({
                                "iDisplayLength": 7,
                                "bJQueryUI": true,

```

```

        "bLengthChange": false,
        "bFilter": true,
        "bInfo": true,
        "sPaginationType": "full_numbers",
        "aaData": data,
        "aoColumns": [{"sTitle": "id"},
el nom de la ruta        {"sTitle": "Nom de la ruta"}, //Columna que conté
l'instant de temps inicial        {"sTitle": "Instant inicial"},//Columna que conté
l'instant de temps final        {"sTitle": "Instant final"},//Columna que conté
la longitud        {"sTitle": "Longitud (km)"}, //Columna que conté
comentari        {"sTitle": "Comentari"},// Columna que conté el
botó per esborrar la ruta        {"sTitle": "Esborrar"}] // Columna que conté un

        });
    } else {
        $('#rutesEmmagatzemades').html("");
    }
    });
}
);
</script>
</head>

<body>
    <div id="Logo"></div>

    <!-- Contingut principal -->
    <h1>EL MÓN EN MINIATURA</h1> <!-- títol visible de la pàgina -->
    <p>Benvingut a la nostra pàgina web! <!-- Subtítol visible -->
    <address>        Queda molt per descobrir.</address> <!--
Subtítol visible-->
    <div id="imagen"></div> <!-- imatge -->
    <form        name="esborrar"        action="upload.html"        method="post"
enctype="multipart/form-data">

        <div id="rutesEmmagatzemades"></div>
    </form>
    <h1>CREA NOVES RUTES:</h1>
    <div id="insertarNovesRutes">
        <form:form        name="upload"        action="upload.html"        method="post"
enctype="multipart/form-data">
            <table id ="tabla1"> <!-- A continuació hi ha el
formulari per descarregar noves rutes. A quest formulari està estructurat de
manera que és una taula -->
                <tr>
                    <td>
                        Nom ruta:
                    </td>
                    <td>
                </tr>
            </table>

```

```

                <input type="text" name="nomRuta"/>
<!-- Entrada de text -->
            </td>
        </tr>
        <tr>
            <td>
                Comentari:
            </td>
            <td>
                <textarea name="comentari"
style="margin: 2px; height: 88px; width: 304px;"></textarea>
<!-- Entrada de text -->
            </td>
        </tr>
        <tr>
            <td>
            </td>
            <td>
                <input type="file" name="data"
id="upload"/>
<!-- Botó que permet cercar dins l'ordinador i descarregar el fitxer del mòbil -->
            </td>
        </tr>
        <tr>
            <td>
            </td>
            <td>
                <input type="button" id="Boton1"
value="Desa" onClick="uploadFile()">
<!-- Botó <desa> -->
            </td>
        </tr>
    </table>
</form:form>
</div>

</body>
</html>

```

## detallRuta.jsp

```
<%@taglib uri="http://java.sun.com/jstl/core_rt" prefix="c"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-
1">
        <title>Rutes Emmagatzemades</title>
        <link rel="stylesheet" href="/VisualitzadorWebDeRutes/css/Style.css">

        <!-- Aquest Script serveix per descarregar les llibreries de Google
Maps
Gràcies a això, es pot utilitzar el mapa i totes les funcions que
google permet -->
        <script type="text/javascript"
            src="https://maps.googleapis.com/maps/api/js?sensor=false">
        </script>

        <!-- Aquí comença l'Script principal.
Es realitzen totes les funcions relacionades amb el mapa -->
        <script type="text/javascript">
            /* Variables utilitzades */
            var map; /* Representa el mapa que apareix actualment */
            var mapOptions; /* Opcions escollides al mapa */
            var poly; /* Línia que representa la ruta */
            var elevator; /* Servei de Google que permet consultar l'alçada
d'un punt */
            var posicions = []; /* Vector de posicions de la ruta */
            var i = 0; /* comptador */
            var contorn = new google.maps.LatLngBounds(); /* Defineix un
contorn màxim de la ruta per tal de que el zoon s'ajusti a la mida de la ruta*/

            /* FUNCIONS */

            /* Funció que s'executa quan s'ha acabat de carregar la pàgina.
S'encarrega d'inicialitzar el mapa així com el servei per sol·licitar alçades i
pintar la ruta sobre el mapa*/
            function init()
            {
                /* Es defineixen les opcions del mapa */
                mapOptions = {
                    zoom: 18, /* Nivell inicial de zoom */
                    mapTypeId: google.maps.MapTypeId.SATELLITE, /*
Permet decidir el tipus de mapa que es vol mostrar (normal, imatges de
satèl·lit...) */

                    mapTypeControlOptions:{
                        style:
google.maps.MapTypeControlStyle.DROPDOWN_MENU /* Activa el control de canvi de
tipus de canvi de mapa*/
                    }
                };

                map = new
google.maps.Map(document.getElementById("map"),mapOptions);
                /* Crea un mapa amb les opcions decidides */
            }
        }
    }
</script>
</head>
    <body>
        <div id="map">
        </div>
    </body>
</html>
```

```

        elevator = new google.maps.ElevationService();
        /* Activa el servei de consulta d'alçades */

        google.maps.event.addListenerOnce(map, "idle", pintaCoordenades);
        /* Aquesta funció, només s'executa una vegada en el
moment que el mapa
està preparat "idle". S'encarrega de cridar a la funció
"pintaCoordenades"*/
    }

    function pintaCoordenades(){
        var polyOptions = { /* Estableix les opcions de la línia
que es pintarà */
            strokeColor: '#ffff1e', /* Defineix color de
la línia */
            strokeOpacity: 1.0, /* Defineix transparència
de la línia */
            strokeWeight: 3 /* Defineix el nombre de
pixels d'amplada de la línia */
        };

        poly = new google.maps.Polyline(polyOptions); /* Genera
una línia que es pot insertar sobre el mapa de google */

        poly.setMap(map);

        <c:forEach var="posicio" items="{ruta._Posicions}">

            /* Bucle que recorre cadascuna de les posicions que el
servidor ha enviat. Les posicions estan emmagatzemades
a la variable ruta_Posicions i es va consultant da cada
posició les dades que es contenen, com l'instant al
que s'ha pres la mesura, la latitud i la longitud. També
es crida a la funció getElevation() que s'encarrega
de fer la petició del valor de l'alçada en un punt
concret*/

            var instantPosicio = "{posicio._instantPosicio}";
            lat = <c:out value="{posicio._latitud}"/>;
            lng = <c:out value="{posicio._longitud}"/>;
            var latLng = new google.maps.LatLng(lat, lng);
            var path = poly.getPath();
            path.push(latLng);
            var posicio = new Object();
            posicio.lat = lat;
            posicio.lng = lng;
            posicio.instant = instantPosicio;
            posicio.alçada = 0;

            posicions.push(posicio);
            getElevation(latLng);
            contorn.extend(latLng);
        </c:forEach>

        map.fitBounds(contorn);
    };

    // Font: google maps api - https://google-
developers.appspot.com/maps/documentation/javascript/examples/elevation-
simple.html

```

```

    function getElevation(latLng) {
        var locations = [];
        locations.push(latLng);

        var positionalRequest = {
            'locations': locations
        };

elevator.getElevationForLocations(positionalRequest,getResultats);
    };

    /* Funció que s'executa al rebre el resultat demanat.
    S'encarrega de generar un marcador circular per a cada posició que permet
    visualitzar les seves dades ( instant de captura, latitud,
    longitud i alçada). Els únics marxadors diferents són el de la posició
    inicial (que té la imatge d'un corredor) i el punt final
    (imatge d'una línia de meta). Com és la funció on es va obtenint l'alçada
    de cada punt, també s'ha aprofitat per anar calculant quina és
    l'alçada màxima i mínima de la ruta*/
    function getResultats(results, status) {
        if (status == google.maps.ElevationStatus.OK) {

            // Retrieve the first result
            if (results[0]) {
                var alcada = results[0].elevation;
                posicions[i].alcada = Math.round(alcada);
                var posicio = posicions[i];
                var title = "Latitud:
"+posicio.lat+"\nLongitud: "+posicio.lng+"\nAlçada: "+posicio.alcada+"\nInstant:
"+posicio.instant;

                if(i==0){
                    var markerImatge = new
google.maps.MarkerImage(
'/VisualitzadorWebDeRutes/images/markerIcon/corredor.png',
new
google.maps.Size(20,45),
new
google.maps.Point(0,0),
new
google.maps.Point(10,45)
);

                    var markerOmbra = new
google.maps.MarkerImage(
'/VisualitzadorWebDeRutes/images/markerIcon/ombra.png',
new
google.maps.Size(46,45),
new
google.maps.Point(0,0),
new
google.maps.Point(10,45)
);

                    var forma = {

```



```

                                coord:
[12,0,12,1,12,2,12,3,12,4,12,5,14,6,15,7,16,8,17,9,17,10,18,11,19,12,19,13,19,14,1
9,15,18,16,17,17,17,18,15,19,15,20,16,21,16,22,15,23,15,24,15,25,15,26,15,27,15,28
,15,29,15,30,15,31,15,32,15,33,14,34,13,35,13,36,11,37,11,38,12,39,12,40,13,41,13,
42,13,43,13,44,8,44,8,43,8,42,9,41,9,40,8,39,8,38,8,37,7,36,7,35,7,34,7,33,7,32,7,
31,7,30,7,29,6,28,6,27,6,26,6,25,6,24,6,23,5,22,5,21,5,20,5,19,1,18,0,17,0,16,0,15
,0,14,2,13,2,12,3,11,3,10,3,9,4,8,4,7,5,6,7,5,7,4,7,3,7,2,7,1,7,0,12,0],
                                type: 'poly'
                                });

                                var marker = new

google.maps.Marker({
                                position:
results[0].location,
                                title: title,
                                map: map,
                                icon:
markerImatge,
                                shadow:
markerOmbra,
                                shape: forma
                                });
                                }else if(i==posicions.length-1)
                                {
                                var markerImatge = new

google.maps.MarkerImage(
                                new
'/VisualitzadorWebDeRutes/images/markerIcon/meta.png',
                                new
google.maps.Size(20,17),
                                new
google.maps.Point(0,0),
                                new
google.maps.Point(10,8)
                                );
                                var marker = new

google.maps.Marker({
                                position: results[0].location,
                                title: title,
                                map: map,
                                icon: markerImatge
                                });

                                //calcul de l'alçada màxima i mínima
                                var max = trobarAlçadaMax();
                                var min = trobarAlçadaMin();

                                document.getElementById("max").innerHTML=max+" m";

                                document.getElementById("min").innerHTML=min+" m";

                                //Centrar mapa

                                }
                                else{
                                var markerImatge = new google.maps.MarkerImage(

```

```

'/VisualitzadorWebDeRutes/images/markerIcon/imatgeposicio.png',
    new google.maps.Size(8,8),
    new google.maps.Point(0,0),
    new google.maps.Point(4,4)
    );
    var marker = new google.maps.Marker({
        position: results[0].location,
        title: title,
        map: map,
        icon: markerImatge
    });
    }
    i++;
} else {
    alert("no s'ha trobat resultat");
}
} else {
    alert("Error: " + status);
}
};

// Càlcul alçada màxima
function trobarAlçadaMax(){
    var max=0;

    for(var i=0;i<posicions.length;i++){
        if(posicions[i].alcada>max){
            max = posicions[i].alcada;
        }
    }
    return max;
};

// Càlcul alçada mínima
function trobarAlçadaMin(){
    var min = posicions[0].alcada;

    for(var i=0;i<posicions.length;i++){
        if(posicions[i].alcada<min){
            min = posicions[i].alcada;
        }
    }
    return min;
};

// Taula de la part superior de la pàgina que mostra tots els
paràmetres d'una ruta en concret
</script>
</head>
<body onload="init()">
    <h1>${ruta._NomRuta}</h1>
    <table id ="taula2">
        <tr>
            <td id="columnaTitols">
                Nom ruta:
            </td>
            <td>

```

```

        ${ruta._NomRuta}
    </td>
    <td>
        Desnivell màxim:
    </td>
    <td>
    </td>
</tr>
<tr>
    <td>
        Longitud:
    </td>
    <td>
        ${ruta._Longitud} km
    </td>
    <td>
        Alçada màxima:
    <td id="max">
    </td>
</tr>
<tr>
    <td>
        Instant inicial:
    </td>
    <td>
        ${ruta._IniciTemps}
    </td>
    <td>
        Alçada mínima:
    <td id="min">
    </td>
</tr>
<tr>
    <td>
        Instant final:
    </td>
    <td>
        ${ruta._FinalTemps}
    </td>
    <td>
        Temps total:
    </td>
    <td>
        ${ruta._TempsTotal}
    </td>
</tr>
<tr>
    <td>
        Comentari:
    </td>
    <td>
        <textarea disabled="disabled">
    </td>
    <td>
        ${ruta._Comentaris} </textarea>
    </td>
</tr>

```

```
        <td>
        </td>
        <td>
        </td>
    </tr>
</table>
<div id="map"></div>
</body>
</html>
```

## ***CODI COMPLET DE LA PROGRAMACIÓ DE L'APLICACIÓ ANDROID***

A continuació es mostra la totalitat de codi necessari per al funcionament de l'aplicació Android <CapturadorDeRutes>.

Les fulles de codi descrites tenen adjunts els comentaris pertinents que permeten entendre el funcionament emprat.

MainActivity.java

```
public class MainActivity extends Activity {
    private Button stopButton, startButton; //Variables que fan referència als
    botons
    private SharedPreferences preferences; // Variable que permet recordar
    valors i compartir-los dins del programa
    private TextView estatGPS, posCapturades; // Variables que fan referència
    als llocs on apareix l'estat i el nombre de posicions capturades
    private ProcesLocalitzacio loc_thr; // Objecte que permet inicialitzar les
    mesures GPS.
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    //Inicialització del programa
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    //Amagar inicialment el boto de parada
    stopButton = (Button) findViewById(R.id.stopButton);
    stopButton.setVisibility(View.INVISIBLE);
    stopButton.setClickable(false);

    //Inicialització del boto comença
    startButton = (Button) findViewById(R.id.startButton);

    //Inicialitza els TextView per escriure resultats
    estatGPS = (TextView) findViewById(R.id.id_estatGPS);
    posCapturades = (TextView) findViewById(R.id.id_PosicionsCapturades);

    //Configuracio del les preferencies.q
    preferences = this.getPreferences(MODE_PRIVATE);
    preferences.edit().putBoolean("PREFS_STOP", false).commit();
```

```
//Es defineix un Listener que s'executa quan l'usuari clica el boto
"Comença"
```

```
startButton.setOnClickListener(new View.OnClickListener() {

    public void onClick(View v) {
        preferences.edit().putBoolean("PREFS_STOP", false).commit();
//Guarda l'estat del botó
        loc_thr = new
ProcesLocalitzacio(getApplicationContext(), preferences, estatGPS, posCapturades);
//Crea un procés de localització
        loc_thr.run(); //Inicialitza la localització
        startButton.setVisibility(View.INVISIBLE);
        startButton.setClickable(false);
        stopButton.setVisibility(View.VISIBLE);
        stopButton.setClickable(true);
    }
});
```

```
//Es defineix un Listener que s'executa quan l'usuari clica el boto
"Atura"
```

```
stopButton.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
```

```

        preferences.edit().putBoolean("PREFS_STOP",
true).commit();
        startButton.setVisibility(View.VISIBLE);
        startButton.setClickable(true);
        stopButton.setVisibility(View.INVISIBLE);
        stopButton.setClickable(false);
    }
});
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is
present.
    getMenuInflater().inflate(R.menu.activity_main, menu);
    return true;
}
}

```

ProcesLocalització.java

```

package com.example.cdr;

@SuppressLint("SimpleDateFormat")
public class ProcesLocalitzacio extends Thread{
    private int numPosicionsCapturades = 0;

    private TextView posCapturades, estatGPS; // Variables que fan referència
als llocs on apareix l'estat i el nombre de posicions capturades
    private SharedPreferences preferences; // Variable que permet recordar
valors i compartirlos dins del programa

    private Context appContext; //Context de l'aplicació
    private LocationManager locationManager; // Encarregat d'obtenir les
posicions
    private LocationListener locationListener; // Encarregat d'executar una
funció després de rebre una posició

    //Objectes per manipular fitxers
    private FileOutputStream loc_FOS;
    private OutputStreamWriter loc_OSW;
    private File loc_file;

    // CONSTRUCTOR
    @SuppressLint("SimpleDateFormat")
    public ProcesLocalitzacio(Context appContext, SharedPreferences preferences,
TextView estatGPS, TextView posCapturades){
        this.preferences = preferences;
        this.appContext = appContext;
        this.estatGPS = estatGPS;
        this.posCapturades = posCapturades;

        //Creació del format pel nom del fitxer on es guardaran les mesures
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy_MM_dd_HH_mm_ss");
        String currentDateandTime = sdf.format(new Date());

        //Creació d'un fitxer per tal d'emmagatzemar les posicions rebudes
        try {

```

```

        loc_file = new File(appContext.getString(R.string.path_ruta) +
        "_" + currentDateandTime + ".txt");
        if(!loc_file.exists())
            loc_file.createNewFile();
        else{
            loc_file.delete();
            loc_file.createNewFile();
        }
        loc_file.mkdirs();
        loc_FOS = new FileOutputStream(loc_file.getPath());
        loc_OSW = new OutputStreamWriter(loc_FOS);
    } catch (IOException e) {
        e.printStackTrace();
    }
}

//Inici del procés de localització
public void run(){
    locationManager = (LocationManager)
appContext.getSystemService(Context.LOCATION_SERVICE);
    //Definició del Listener de localització
    locationListener = new LocationListener(){
        //Funció que s'executa cada vegada que el mòdul GPS calcula una
posició
        public void onLocationChanged(Location location) {
            SimpleDateFormat sdf = new SimpleDateFormat("yyyy/MM/dd
HH:mm:ss");

            //Informació per mostrar a la conola
            Log.d("INFORMACIO", "LATITUD_" +
location.getLatitude());
            Log.d("INFORMACIO", "LONGITUD_" +
location.getLongitude());
            float precisió = location.getAccuracy();
            Log.d("INFORMACIO", "PRECISIO_" +
location.getAccuracy());

            if(precisió<=15){ //Escriu al fitxer si la precisió és
millor que 10 m
                //Actualitza al visor advertint a l'usuari que el
programa està preparat
                estatGPS.setText("
"+appContext.getString(R.string.Preparat));
                numPosicionsCapturades++;
                try {
                    //Escriptura segons el format decidit:
"yyyy/MM/dd HH:mm:ss , lat , lon, precisió"
                    loc_OSW.write(sdf.format(new Date())+",");
                    loc_OSW.write(location.getLatitude() +
",");
                    loc_OSW.write(location.getLongitude()+"");

                    //Actualitza al visor el nombre de
posicions capturades.

                    posCapturades.setText(Integer.toString(numPosicionsCapturades));

                    if(preferences.getBoolean("PREFS_STOP",
false)){

```



```

        loc_OSW.close();
        loc_FOS.close();

locationManager.removeUpdates(locationListener);
        Log.d("DEBUG_LOC", "FINISHED");
    }
    } catch (IOException e) {
        e.printStackTrace();
    }
} else { //Si no té prou precisió la mostra no s'escriu al
fitxer.
        Log.d("INFORMACIO", "Mostra descartada per tenir
poca precisió");
        estatGPS.setText("
"+appContext.getString(R.string.NoPreparat));
    }
}

public void onProviderDisabled(String provider) {
}

public void onProviderEnabled(String provider) {
}

public void onStatusChanged(String provider, int status,
Bundle extras) {
}
};
//Preparació del mòdul GPS per a que agafi mesures cada 5 segons
(5s*1000 = 5000ms) i executi la funció "locationListener()"
locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
5*1000, 0, locationListener);
}
}

```

## AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.cdr"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="7"
        android:targetSdkVersion="17" />

    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.READ_LOGS" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.cdr.MainActivity"
            android:label="@string/app_name"
            android:screenOrientation="portrait">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

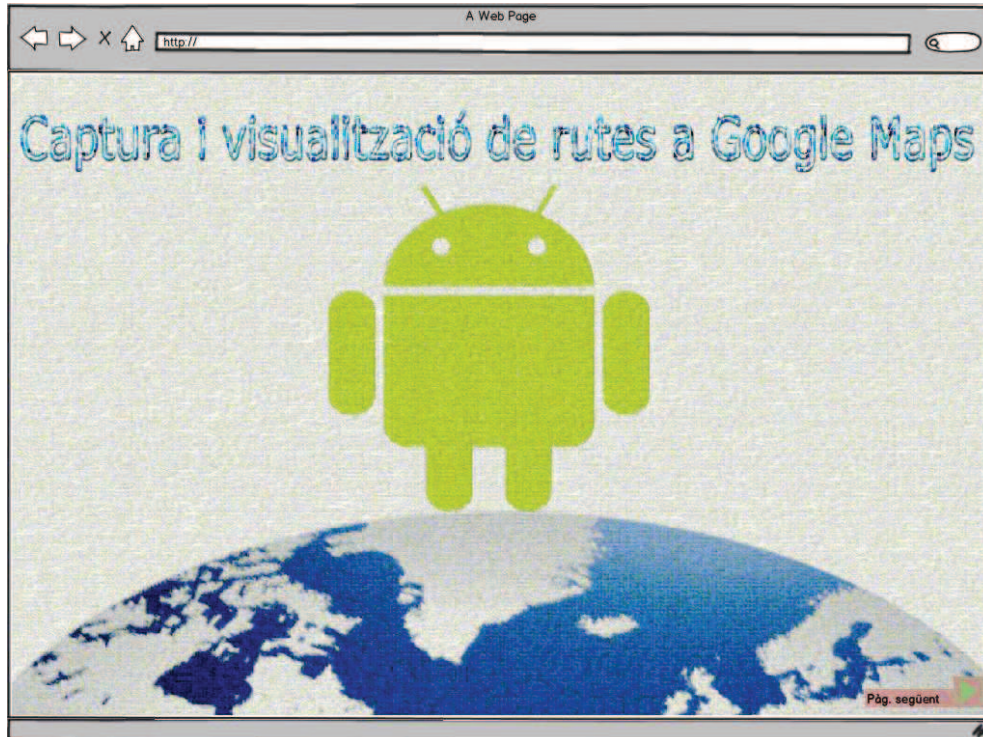
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

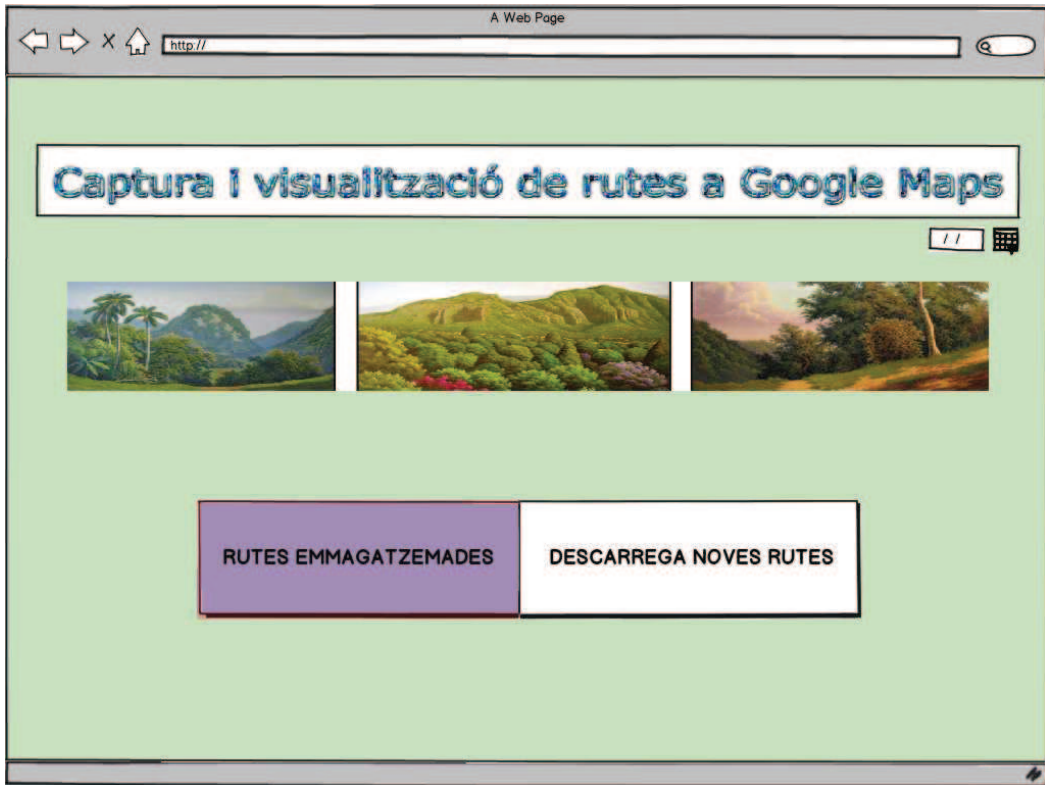
## ***ESBOSSOS DUTS A TERME AMB EL PROGRAMA BALSAMIQ MOCKUPS***

Seguidament es mostren els dissenys complets que es varen dur a terme amb el programa Balsamiq Mockups com a prova prèvia al disseny definitiu.

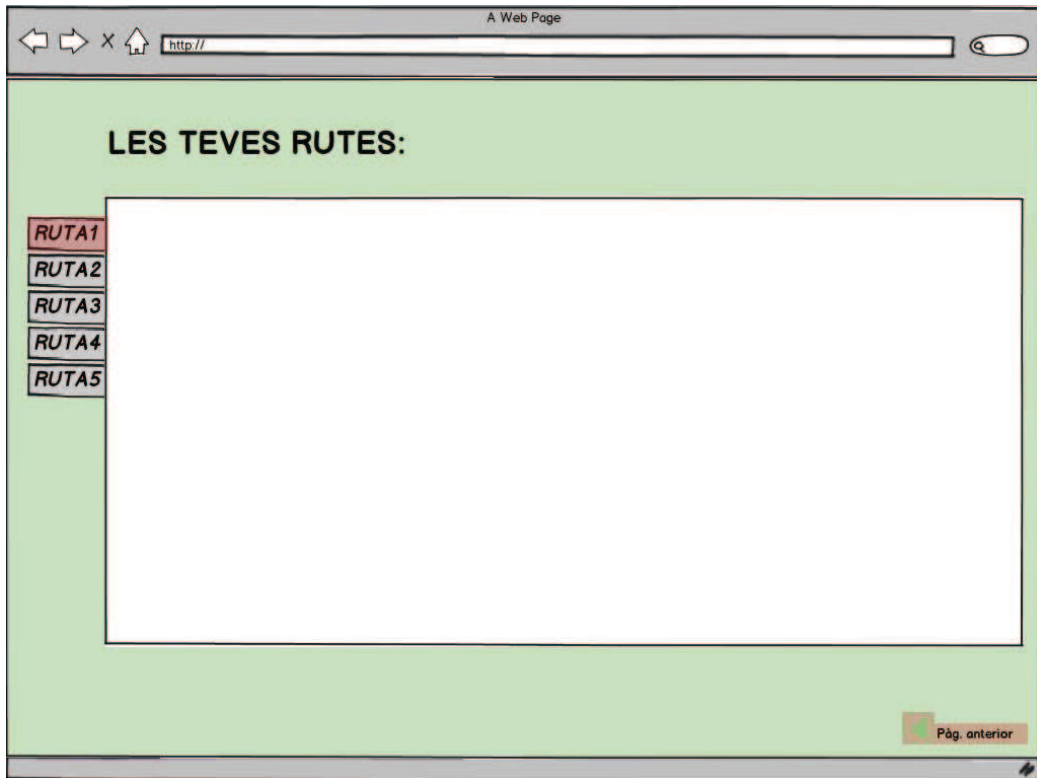
### ***Annex VIII - Pàgina web***



*FIGURA 1: Pàgina inicial*



*FIGURA 2: Pàgina principal*



*FIGURA 3: Menú Rutes emmagatzemades*

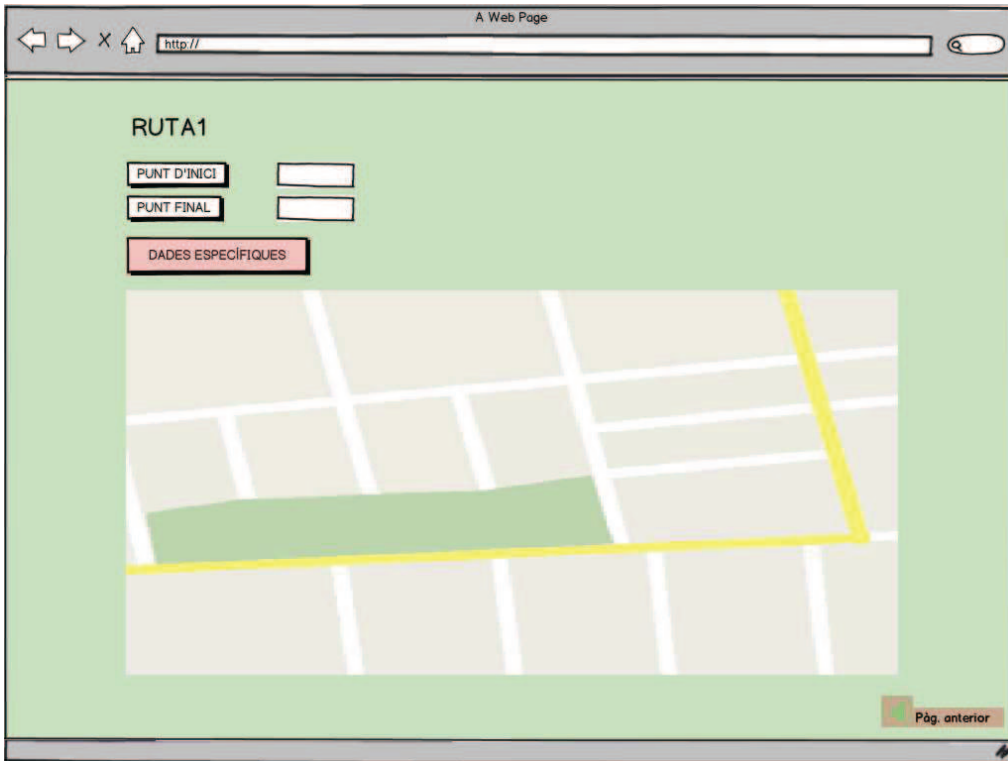


FIGURA 4: Ruta 1

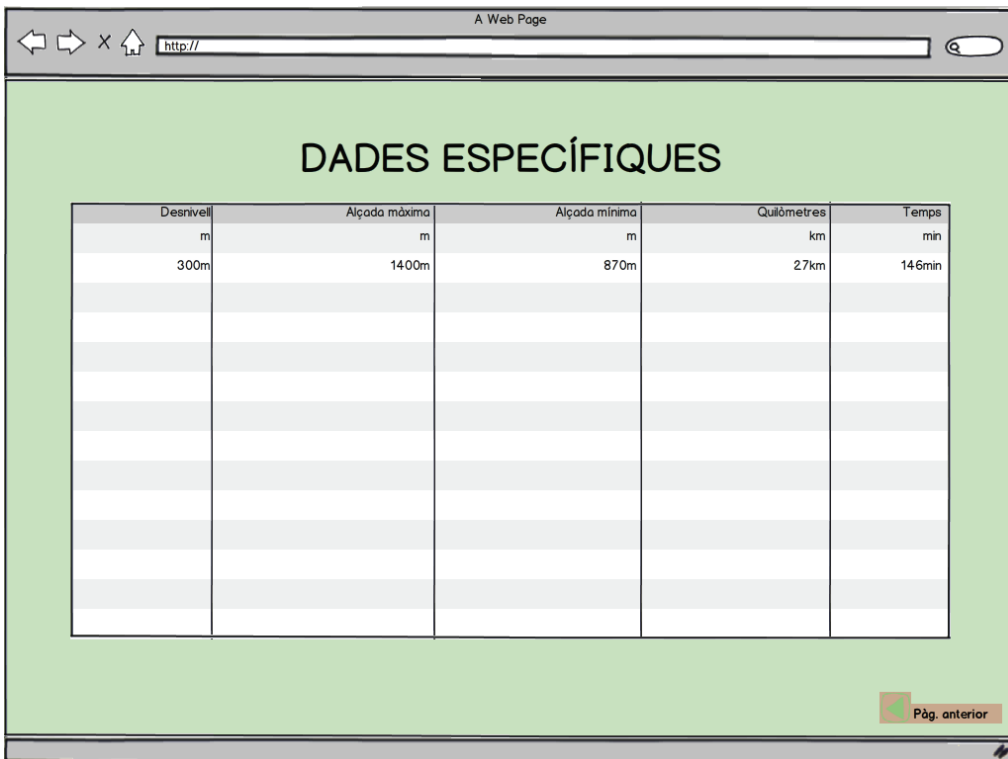


FIGURA 5: Dades específiques

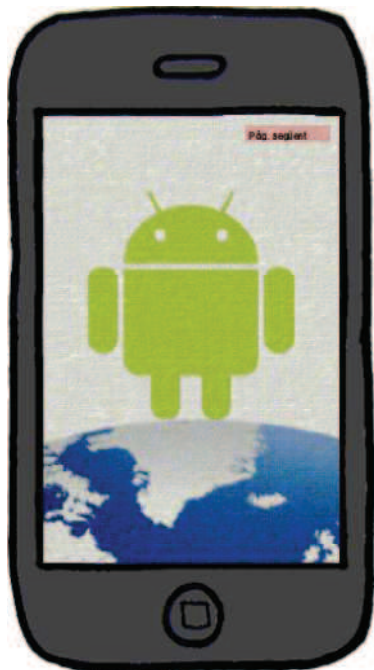


FIGURA 6: Pàgina inicial



FIGURA 7: Pàgina principal



FIGURA 8: Inici ruta



FIGURA 9: Acaba



FIGURA 10: Rutes emmagatzemades



FIGURA 11: Rutes emmagatzemades



## ***FITXERS OBTINGUTS AMB L'EXECUCIÓ D'UNA RUTA***

Un cop acabada la part pràctica, s'ha provat que tot funcionés a la perfecció.

Primerament s'ha fet un fitxer de prova manual per tal de provar el funcionament i a continuació, s'ha provat que el fitxer creat pel mòbil tingués les condicions requerides.

Aquí es mostra la totalitat dels dos fitxers:

### ***Annex X - Fitxer de prova: ruta vinya vella***

2012/12/01 14:00:00,42.234835,1.876484
2012/12/01 14:01:00,42.234303,1.876972
2012/12/01 14:02:00,42.234126,1.876830
2012/12/01 14:03:00,42.233570,1.877232
2012/12/01 14:04:00,42.233699,1.876854
2012/12/01 14:05:00,42.233748,1.875934
2012/12/01 14:06:00,42.233649,1.875612
2012/12/01 14:07:00,42.234314,1.875403
2012/12/01 14:08:00,42.234467,1.875231
2012/12/01 14:09:00,42.234839,1.874856
2012/12/01 14:10:00,42.235246,1.874762
2012/12/01 14:11:00,42.235591,1.874826
2012/12/01 14:12:00,42.235998,1.874620
2012/12/01 14:13:00,42.236551,1.873807
2012/12/01 14:14:00,42.236787,1.872908
2012/12/01 14:15:00,42.237635,1.871937
2012/12/01 14:16:00,42.237591,1.871283
2012/12/01 14:17:00,42.237714,1.87093
2012/12/01 14:18:00,42.237375,1.87006
2012/12/01 14:19:00,42.237434,1.86904
2012/12/01 14:20:00,42.237780,1.86745
2012/12/01 14:21:00,42.238066,1.86572
2012/12/01 14:22:00,42.237833,1.86540

*FIGURA 12: Fitxer de prova: 23 posicions.*

*Annex XI - Fitxer real: ruta vinya vella*

2013/01/24 17:06:59,42.23472154,1.87674142,12.0  
2013/01/24 17:07:04,42.23471867,1.87675007,10.0  
2013/01/24 17:07:09,42.23467376,1.87675915,10.0  
2013/01/24 17:07:14,42.23469518,1.87659174,6.0  
2013/01/24 17:07:19,42.23466802,1.87660378,3.0  
2013/01/24 17:07:24,42.23463302,1.87665359,3.0  
2013/01/24 17:07:29,42.23458399,1.87672011,6.0  
2013/01/24 17:07:34,42.23454174,1.87675138,12.0  
2013/01/24 17:07:39,42.23450195,1.87676231,10.0  
2013/01/24 17:07:45,42.23441377,1.87687334,4.0  
2013/01/24 17:07:50,42.23434608,1.87694624,3.0  
2013/01/24 17:07:55,42.23427389,1.87693711,3.0  
2013/01/24 17:08:00,42.23420316,1.87694629,4.0  
2013/01/24 17:08:05,42.2341661,1.87692686,4.0  
2013/01/24 17:08:10,42.23410221,1.8769365,3.0  
2013/01/24 17:08:15,42.23404736,1.87694183,3.0  
2013/01/24 17:08:20,42.23400017,1.87693382,3.0  
2013/01/24 17:08:25,42.23394432,1.87689843,3.0  
2013/01/24 17:08:30,42.23388342,1.87691264,4.0  
2013/01/24 17:08:35,42.23383909,1.87693698,4.0  
2013/01/24 17:08:41,42.23379034,1.8770071,3.0  
2013/01/24 17:08:46,42.233747,1.8770572,3.0  
2013/01/24 17:08:51,42.23370728,1.87710397,2.0  
2013/01/24 17:08:56,42.23368675,1.87712899,2.0  
2013/01/24 17:09:00,42.23364306,1.87716937,2.0  
2013/01/24 17:09:05,42.23354539,1.87717915,7.0  
2013/01/24 17:09:11,42.23347267,1.87720873,6.0  
2013/01/24 17:09:16,42.23342659,1.87714395,5.0  
2013/01/24 17:09:21,42.23344725,1.87709428,5.0  
2013/01/24 17:09:26,42.23347659,1.87704544,5.0  
2013/01/24 17:09:31,42.23352739,1.87699503,4.0  
2013/01/24 17:09:37,42.23358529,1.87694342,3.0  
2013/01/24 17:09:42,42.23362001,1.87687002,3.0  
2013/01/24 17:09:46,42.23363983,1.87681194,3.0  
2013/01/24 17:09:52,42.23367877,1.87674974,6.0  
2013/01/24 17:09:57,42.23370212,1.87670056,4.0  
2013/01/24 17:10:02,42.23372888,1.87664768,3.0  
2013/01/24 17:10:07,42.23374561,1.87658248,3.0  
2013/01/24 17:10:12,42.23377259,1.87649808,4.0  
2013/01/24 17:10:17,42.2337918,1.87642786,3.0  
2013/01/24 17:10:22,42.23379037,1.87635002,3.0  
2013/01/24 17:10:27,42.23379482,1.87628597,3.0  
2013/01/24 17:10:33,42.23378557,1.87621369,4.0  
2013/01/24 17:10:38,42.23378429,1.87613645,3.0  
2013/01/24 17:10:43,42.23375434,1.8761041,4.0  
2013/01/24 17:10:48,42.23373545,1.87603082,4.0  
2013/01/24 17:10:53,42.23372821,1.87594391,3.0  
2013/01/24 17:10:58,42.23370333,1.87587305,3.0  
2013/01/24 17:11:03,42.23368549,1.87581358,2.0  
2013/01/24 17:11:08,42.233656,1.87575191,3.0  
2013/01/24 17:11:13,42.23364544,1.87568615,2.0  
2013/01/24 17:11:18,42.23364455,1.87561024,3.0  
2013/01/24 17:11:23,42.23363432,1.87557303,4.0



2013/01/24 17:11:29,42.23367854,1.87555135,4.0  
2013/01/24 17:11:34,42.23374858,1.87551927,4.0  
2013/01/24 17:11:39,42.23381852,1.87550843,3.0  
2013/01/24 17:11:44,42.2338532,1.87551813,4.0  
2013/01/24 17:11:48,42.23390726,1.87550424,4.0  
2013/01/24 17:11:54,42.23398527,1.87548801,3.0  
2013/01/24 17:11:58,42.23401758,1.87549631,3.0  
2013/01/24 17:12:03,42.2340824,1.87551088,4.0  
2013/01/24 17:12:09,42.23411792,1.87548459,4.0  
2013/01/24 17:12:14,42.23420718,1.87547186,3.0  
2013/01/24 17:12:19,42.23424635,1.87543739,3.0  
2013/01/24 17:12:25,42.23428865,1.87542988,3.0  
2013/01/24 17:12:30,42.23435401,1.87541368,3.0  
2013/01/24 17:12:35,42.23439673,1.87543345,3.0  
2013/01/24 17:12:40,42.23443219,1.87540843,2.0  
2013/01/24 17:12:45,42.2344612,1.87536074,2.0  
2013/01/24 17:12:50,42.23448842,1.87529294,2.0  
2013/01/24 17:12:55,42.23451692,1.87523839,3.0  
2013/01/24 17:13:00,42.23453854,1.87519916,3.0  
2013/01/24 17:13:05,42.23458974,1.87516199,2.0  
2013/01/24 17:13:10,42.23464348,1.87514036,2.0  
2013/01/24 17:13:15,42.23467824,1.87508483,2.0  
2013/01/24 17:13:21,42.23472725,1.87502308,2.0  
2013/01/24 17:13:26,42.23478014,1.87496335,2.0  
2013/01/24 17:13:31,42.23482974,1.87492327,2.0  
2013/01/24 17:13:36,42.23486873,1.87488327,2.0  
2013/01/24 17:13:41,42.2349127,1.87483797,2.0  
2013/01/24 17:13:46,42.23496573,1.87479593,2.0  
2013/01/24 17:13:51,42.23501707,1.87477176,2.0  
2013/01/24 17:13:56,42.23506705,1.87474693,2.0  
2013/01/24 17:14:01,42.23514459,1.87475017,3.0  
2013/01/24 17:14:06,42.2351914,1.87475493,3.0  
2013/01/24 17:14:11,42.2352445,1.87477959,2.0  
2013/01/24 17:14:17,42.23532285,1.87480056,3.0  
2013/01/24 17:14:22,42.23537482,1.87479486,2.0  
2013/01/24 17:14:27,42.23543429,1.87479176,3.0  
2013/01/24 17:14:32,42.23549934,1.87478513,2.0  
2013/01/24 17:14:37,42.2355521,1.87480095,2.0  
2013/01/24 17:14:42,42.23557462,1.87482853,2.0  
2013/01/24 17:14:47,42.23560809,1.87485544,2.0  
2013/01/24 17:14:52,42.23565894,1.87488898,3.0  
2013/01/24 17:14:57,42.23569756,1.8748374,4.0  
2013/01/24 17:15:02,42.23574452,1.87484271,3.0  
2013/01/24 17:15:07,42.23580617,1.87476967,4.0  
2013/01/24 17:15:13,42.23587346,1.87472348,3.0  
2013/01/24 17:15:18,42.23593243,1.87467929,3.0  
2013/01/24 17:15:23,42.23599623,1.87463906,3.0  
2013/01/24 17:15:28,42.23603265,1.87459665,3.0  
2013/01/24 17:15:33,42.23607301,1.87456981,3.0  
2013/01/24 17:15:38,42.23614489,1.87453163,3.0  
2013/01/24 17:15:43,42.23617897,1.87449871,3.0  
2013/01/24 17:15:48,42.23619673,1.87443622,3.0  
2013/01/24 17:15:53,42.23623905,1.87434499,3.0  
2013/01/24 17:15:58,42.23626594,1.87428468,3.0  
2013/01/24 17:16:03,42.23629068,1.87420888,3.0  
2013/01/24 17:16:10,42.23634441,1.87411542,3.0

2013/01/24 17:16:15,42.2363797,1.87406809,3.0  
2013/01/24 17:16:20,42.23642215,1.87401957,3.0  
2013/01/24 17:16:25,42.2364694,1.87398439,3.0  
2013/01/24 17:16:30,42.23649254,1.87393157,3.0  
2013/01/24 17:16:35,42.23652858,1.87389123,3.0  
2013/01/24 17:16:40,42.23655787,1.87382131,3.0  
2013/01/24 17:16:45,42.23659436,1.87375724,3.0  
2013/01/24 17:16:50,42.23663545,1.87369371,3.0  
2013/01/24 17:16:55,42.23666959,1.87361788,2.0  
2013/01/24 17:17:00,42.23669245,1.87354383,2.0  
2013/01/24 17:17:09,42.23672058,1.87343647,2.0  
2013/01/24 17:17:14,42.23672742,1.87338358,2.0  
2013/01/24 17:17:19,42.23675093,1.87331906,2.0  
2013/01/24 17:17:24,42.23675975,1.87323182,2.0  
2013/01/24 17:17:29,42.23677157,1.87314141,2.0  
2013/01/24 17:17:34,42.23676957,1.87308055,3.0  
2013/01/24 17:17:39,42.23679319,1.87299644,2.0  
2013/01/24 17:17:44,42.23681841,1.87293659,2.0  
2013/01/24 17:17:49,42.23681656,1.87292913,2.0  
2013/01/24 17:17:54,42.23685048,1.87287558,2.0  
2013/01/24 17:17:59,42.23688147,1.87281842,2.0  
2013/01/24 17:18:05,42.236918,1.87275493,2.0  
2013/01/24 17:18:10,42.23695339,1.87267839,2.0  
2013/01/24 17:18:15,42.23701182,1.87261396,2.0  
2013/01/24 17:18:20,42.23706856,1.87257813,2.0  
2013/01/24 17:18:25,42.23712075,1.87253526,2.0  
2013/01/24 17:18:30,42.23717974,1.87249661,2.0  
2013/01/24 17:18:35,42.23722844,1.8724676,2.0  
2013/01/24 17:18:40,42.23727909,1.87242861,2.0  
2013/01/24 17:18:45,42.23734316,1.87237321,2.0  
2013/01/24 17:18:50,42.23739257,1.87231976,2.0  
2013/01/24 17:18:55,42.23742516,1.87226197,2.0  
2013/01/24 17:19:01,42.23745671,1.87221372,2.0  
2013/01/24 17:19:06,42.23748825,1.87216605,2.0  
2013/01/24 17:19:11,42.23752926,1.87214352,2.0  
2013/01/24 17:19:16,42.23756276,1.87209136,2.0  
2013/01/24 17:19:21,42.23761258,1.87203319,2.0  
2013/01/24 17:19:26,42.23763333,1.87195747,2.0  
2013/01/24 17:19:31,42.23765184,1.87187098,2.0  
2013/01/24 17:19:36,42.23765892,1.87179995,2.0  
2013/01/24 17:19:41,42.23765499,1.87173825,2.0  
2013/01/24 17:19:46,42.23763737,1.87167442,2.0  
2013/01/24 17:19:51,42.23763122,1.87159136,2.0  
2013/01/24 17:19:57,42.23762769,1.87148336,2.0  
2013/01/24 17:20:02,42.23761616,1.8714164,2.0  
2013/01/24 17:20:07,42.23761641,1.87136096,2.0  
2013/01/24 17:20:12,42.23761122,1.87128964,2.0  
2013/01/24 17:20:17,42.23761274,1.87121462,3.0  
2013/01/24 17:20:22,42.23758384,1.87115832,3.0  
2013/01/24 17:20:27,42.2375914,1.87108492,3.0  
2013/01/24 17:20:32,42.2375663,1.87102376,3.0  
2013/01/24 17:20:37,42.23755168,1.87094461,3.0  
2013/01/24 17:20:42,42.23752226,1.87092204,4.0  
2013/01/24 17:20:48,42.23749549,1.87083217,3.0  
2013/01/24 17:20:54,42.23747566,1.87072545,4.0  
2013/01/24 17:20:58,42.23748309,1.87065503,5.0

2013/01/24 17:21:03,42.23751976,1.87054244,5.0  
2013/01/24 17:21:08,42.23753599,1.8704644,4.0  
2013/01/24 17:21:13,42.23750655,1.8703978,4.0  
2013/01/24 17:21:18,42.23749631,1.87037064,5.0  
2013/01/24 17:21:23,42.23748889,1.87030503,6.0  
2013/01/24 17:21:28,42.23748275,1.87023834,4.0  
2013/01/24 17:21:33,42.23746542,1.87015705,4.0  
2013/01/24 17:21:38,42.23751646,1.8700774,4.0  
2013/01/24 17:21:44,42.23753082,1.87000664,6.0  
2013/01/24 17:21:49,42.23754122,1.87000692,4.0  
2013/01/24 17:21:54,42.23753442,1.86999379,3.0  
2013/01/24 17:21:59,42.23749637,1.86994106,4.0  
2013/01/24 17:22:04,42.23745204,1.86990269,4.0  
2013/01/24 17:22:10,42.23740966,1.86985975,4.0  
2013/01/24 17:22:14,42.23739618,1.86983133,6.0  
2013/01/24 17:22:20,42.23737591,1.86974119,3.0  
2013/01/24 17:22:24,42.23739585,1.86966823,4.0  
2013/01/24 17:22:29,42.23742161,1.86962501,4.0  
2013/01/24 17:22:34,42.23740799,1.86956441,3.0  
2013/01/24 17:22:39,42.23742306,1.86950878,3.0  
2013/01/24 17:22:47,42.2374317,1.86943335,5.0  
2013/01/24 17:22:52,42.23744704,1.86936478,6.0  
2013/01/24 17:22:57,42.23746523,1.86929192,5.0  
2013/01/24 17:23:02,42.2374937,1.86927643,6.0  
2013/01/24 17:23:07,42.23748563,1.8692813,5.0  
2013/01/24 17:23:12,42.23750773,1.86921458,5.0  
2013/01/24 17:23:17,42.23750099,1.8691727,4.0  
2013/01/24 17:23:22,42.23752358,1.86916958,4.0  
2013/01/24 17:23:27,42.23755134,1.86916942,4.0  
2013/01/24 17:23:32,42.23754908,1.86916212,3.0  
2013/01/24 17:23:37,42.23755002,1.8691528,3.0  
2013/01/24 17:23:43,42.23755139,1.86914781,4.0  
2013/01/24 17:23:48,42.23756387,1.86912052,3.0  
2013/01/24 17:23:53,42.23759161,1.86904974,4.0  
2013/01/24 17:23:58,42.23758217,1.86895106,3.0  
2013/01/24 17:24:03,42.23758288,1.8688993,4.0  
2013/01/24 17:24:08,42.23757907,1.86884773,4.0  
2013/01/24 17:24:13,42.23758819,1.86878327,3.0  
2013/01/24 17:24:18,42.23761019,1.86873234,3.0  
2013/01/24 17:24:23,42.23762322,1.86870646,2.0  
2013/01/24 17:24:28,42.2376309,1.868689,3.0  
2013/01/24 17:24:33,42.23763549,1.86864372,3.0  
2013/01/24 17:24:41,42.23765376,1.86857571,4.0  
2013/01/24 17:24:46,42.23766909,1.86857171,4.0  
2013/01/24 17:24:51,42.23768137,1.86850201,3.0  
2013/01/24 17:24:56,42.237696,1.86842399,5.0  
2013/01/24 17:25:01,42.23773269,1.86834843,4.0  
2013/01/24 17:25:06,42.23775205,1.86831444,3.0  
2013/01/24 17:25:11,42.2377893,1.86826391,4.0  
2013/01/24 17:25:16,42.23780322,1.86821112,3.0  
2013/01/24 17:25:21,42.23782165,1.86817111,3.0  
2013/01/24 17:25:26,42.23783518,1.86810362,3.0  
2013/01/24 17:25:31,42.23783958,1.86807286,3.0  
2013/01/24 17:25:37,42.23786443,1.86800658,4.0  
2013/01/24 17:25:42,42.23786113,1.86795597,4.0  
2013/01/24 17:25:47,42.23785477,1.86792043,4.0

2013/01/24 17:25:52,42.23786205,1.86787665,3.0  
2013/01/24 17:25:57,42.23788159,1.86782255,4.0  
2013/01/24 17:26:02,42.23788216,1.86776605,3.0  
2013/01/24 17:26:07,42.23788971,1.86768762,4.0  
2013/01/24 17:26:12,42.23794985,1.86766731,4.0  
2013/01/24 17:26:17,42.23794865,1.8676066,4.0  
2013/01/24 17:26:22,42.23795512,1.86758265,5.0  
2013/01/24 17:26:27,42.23795532,1.86753208,3.0  
2013/01/24 17:26:33,42.23795867,1.86749012,3.0  
2013/01/24 17:26:38,42.23795301,1.8674364,4.0  
2013/01/24 17:26:43,42.23796243,1.86738841,4.0  
2013/01/24 17:26:48,42.23792788,1.86733079,4.0  
2013/01/24 17:26:53,42.23794313,1.86724158,3.0  
2013/01/24 17:26:58,42.23795183,1.86718193,3.0  
2013/01/24 17:27:03,42.2379472,1.86712855,3.0  
2013/01/24 17:27:08,42.23794423,1.86707985,3.0  
2013/01/24 17:27:13,42.23794524,1.86702753,3.0  
2013/01/24 17:27:18,42.23793974,1.86698764,3.0  
2013/01/24 17:27:23,42.23793695,1.86695141,5.0  
2013/01/24 17:27:29,42.23793517,1.86687026,4.0  
2013/01/24 17:27:34,42.23792454,1.86683351,4.0  
2013/01/24 17:27:39,42.2379391,1.86679047,4.0  
2013/01/24 17:27:44,42.23794323,1.86676971,4.0  
2013/01/24 17:27:49,42.2379394,1.86673186,5.0  
2013/01/24 17:27:54,42.23795844,1.86669799,4.0  
2013/01/24 17:27:59,42.23795373,1.86663913,4.0  
2013/01/24 17:28:04,42.23796472,1.86658706,4.0  
2013/01/24 17:28:09,42.23795987,1.86652278,3.0  
2013/01/24 17:28:14,42.23798439,1.86646393,3.0  
2013/01/24 17:28:19,42.23801047,1.86639367,4.0  
2013/01/24 17:28:25,42.23800257,1.86635457,3.0  
2013/01/24 17:28:30,42.23801017,1.8663242,3.0  
2013/01/24 17:28:35,42.23802626,1.86628102,3.0  
2013/01/24 17:28:40,42.23806624,1.86628563,4.0  
2013/01/24 17:28:45,42.23807797,1.86626645,3.0  
2013/01/24 17:28:50,42.23808829,1.86624542,4.0  
2013/01/24 17:28:55,42.23809896,1.86624231,4.0  
2013/01/24 17:29:00,42.23811035,1.86620605,4.0  
2013/01/24 17:29:05,42.2381488,1.86613979,4.0  
2013/01/24 17:29:10,42.2381779,1.86609057,4.0  
2013/01/24 17:29:15,42.23818627,1.86607919,4.0  
2013/01/24 17:29:23,42.23819189,1.86601471,3.0  
2013/01/24 17:29:28,42.23818246,1.86594219,2.0  
2013/01/24 17:29:33,42.23815439,1.86586523,2.0  
2013/01/24 17:29:38,42.23812205,1.86578507,2.0  
2013/01/24 17:29:43,42.23808573,1.86572779,2.0  
2013/01/24 17:29:48,42.23804151,1.86565618,2.0  
2013/01/24 17:29:53,42.23799789,1.8655972,2.0  
2013/01/24 17:29:58,42.23794675,1.86557665,2.0  
2013/01/24 17:30:03,42.23789746,1.86555035,3.0  
2013/01/24 17:30:08,42.23785365,1.8655181,3.0  
2013/01/24 17:30:13,42.23779897,1.86549073,2.0  
2013/01/24 17:30:19,42.23778,1.86547041,3.0

*FIGURA 13: Fixter real; 275 posiciones.*

