
Pràctica 1

JsonTMB



**Transports
Metropolitans
de Barcelona**

"En el metro, todos los caminos
llevan aroma."

Anònim

ÍNDEX

1	Introducció	3
2	Especificació dels Requeriments	4
2.1	Context	4
2.1.1	Fitxer: localitzacions.json	4
2.2	Configuració	4
2.3	Funcionalitats	4
2.3.1	Opció 1: Gestió d'usuari	6
2.3.1.1	Opció a: Les meves localitzacions	7
2.3.1.2	Opció b: Historial de localitzacions	8
2.3.1.3	Opció c: Les meves rutes	9
2.3.1.4	Opció d: Parades i estacions preferides	10
2.3.1.5	Opció e: Estacions inaugurades el meu any de naixement	12
2.3.2	Opció 2: Buscar localitzacions	12
2.3.3	Opció 3: Planejar ruta	14
2.3.4	Opció 4: Temps d'espera del bus	17
2.4	Crides a Webservices	18
3	Consideracions	19
3.1	Procés de Desenvolupament	19
3.2	Disseny de Classes	19
3.3	Emmagatzemament de Dades	19
3.4	Lectura de Fitxers JSON i API's	20
3.5	API de TMB	20
4	Lliurament	21
5	Normativa	22

1 INTRODUCCIÓ

L'empresa TMB (Transports Metropolitans de Barcelona), encarregada de gestionar el servei de metro i bus a la nostra ciutat, ha tingut un problema amb la seva aplicació web i està temporalment inactiva.

Per sort, TMB té constància que a La Salle hi ha els millors programadors, que els podran ajudar i faran una nova versió de la seva aplicació.

Després de l'AC3 on vam llegir fitxers JSON amb la informació d'aigües, sucres i altres begudes per a tots els públics, ara ens tocarà fer-ho amb els llocs més emblemàtics i el transport públic de Barcelona.

Però aquest cop a part d'usar un fitxer JSON, que contindrà diferents localitzacions, utilitzarem l'API de TMB. Per això, ens haurem de registrar al Portal de desenvolupador de TMB, ens donarem d'alta una aplicació i així obtindrem les claus d'accés per després fer-les servir a les URL de les peticions.

2 ESPECIFICACIÓ DELS REQUERIMENTS

A continuació s'exposen les bases funcionals de la pràctica (amb una explicació dels recursos que us proporcionem) i els requeriments de software.

2.1 CONTEXT

El projecte consistirà en crear diverses funcionalitats amb la informació de l'API i el fitxer JSON.

Des de l'equip de l'assignatura us proporcionem un fitxer JSON amb localitzacions. És clau que el feu servir **sabent que els valors poden canviar i, per tant, no podeu *hardcodejar* cap de les seves dades**. L'estructura, per descomptat, serà sempre la mateixa.

2.1.1 FITXER: LOCALITZACIONS.JSON

Aquest fitxer conté la informació de diversos llocs que es faran servir en les funcionalitats de l'API de TMB. Cada localització té el seu nom, les seves coordenades i una breu descripció.

A més, també poden haver altres dades:

- Si la localització és un **monument**, trobareu informació de l'arquitecte que el va dissenyar i la data de la seva inauguració.
- Si la localització és un **restaurant**, tindrà un conjunt de característiques genèriques que defineixen el tipus de local.
- Si la localització és un **hotel**, sabrem quantes estrelles té.

2.2 CONFIGURACIÓ

El primer que farà l'aplicació quan s'executi serà carregar-se i tractar les dades del fitxer *localitzacions.json*, que s'hauran de guardar en una classe anomenada *DataModel*. **En cas que hi hagi un problema a l'hora d'obrir o llegir el fitxer, es sortirà de l'aplicació amb un missatge informatiu.**

Si tot ha anat bé, iniciarem l'aplicació de forma normal i començarem a executar la lògica detallada a continuació.

2.3 FUNCIONALITATS

Inicialment, es demanarà un nom d'usuari, correu electrònic i l'any de naixement. Assumirem que les dades introduïdes per l'usuari sempre estaran bé.

```
1  Benvingut a l'aplicació de TMBJson! Si us plau, introdueix les dades que se't demanen.  
2  
3  Nom d'usuari:  
4  CarlitosCheetoboy  
5  
6  Correu electrònic:  
7  carlitoscheetoboy@gmail.com  
8  
9  Any de naixement:  
10 1999  
11  
12 La informació s'ha registrat amb èxit!  
13  
14 /*Aquí hi hauria el menú (no printeu aquesta línia)*/
```

Snippet 1: Demanem la informació personal de l'usuari

Opcionalment, aquestes dades s'hauran d'emmagatzemar en un json anomenat *user.json*. Aquest fitxer es crearà un cop s'hagin introduït les 3 dades demanades anteriorment. A més, quan l'usuari surti de l'aplicació (opció 5 del menú) també s'hauran d'afegir al fitxer les localitzacions que hagi creat com les que hagi assignat com preferides. En cas que es vulgui entrar a l'aplicació i ja s'hagi creat un usuari, no es tornarà a demanar el nom, correu i any de naixement. També tota la informació emmagatzemada anteriorment al fitxer *user.json*, es recuperarà i així no perdrem res que s'hagi fet en una altra sessió.

```
1  Benvingut de nou a l'aplicació de TMBJson CarlitosCheetoboy!  
2  
3  /*Aquí hi hauria el menú (no printeu aquesta línia)*/
```

Snippet 2: OPCIONAL Accedim directament al menú perquè ja hem creat un usuari

A continuació, l'aplicació haurà de mostrar un menú per consola de comandes que permeti accedir a les diverses opcions descrites a continuació. No es deixarà de visualitzar fins que l'usuari decideixi sortir del programa (serà l'opció 5 del menú).

S'haurà de comprovar que l'usuari introdueixi un **enter dins el rang [1, 5]**. En cas que s'introdueixi una opció no vàlida, es mostrarà un missatge d'error explicatiu i un altre cop el menú.

Considerarem **en tots els menús** que l'usuari a l'hora d'escollir una opció, podrà introduir qualsevol cosa i per tant, s'haurà de tenir en compte a l'hora de gestionar errors.

Sempre que s'hagi d'introduir un *string* o un *char*, **el tractament de l'entrada serà *case-insensitive* en tota la pràctica**. Això vol dir que acceptarem lletres tant majúscules com minúscules.

```
1  1. Gestió d'usuari
2  2. Buscar localitzacions
3  3. Planejar ruta
4  4. Temps d'espera del bus
5  5. Sortir
6
7  Selecciona una opció:
```

Snippet 3: Menú principal del programa.

El funcionament de les diferents opcions s'exposa a continuació:

2.3.1 OPCIO 1: GESTIÓ D'USUARI

En aquesta opció apareixerà un menú on podrem crear localitzacions i veure les localitzacions creades, l'historial de localitzacions, les rutes que haguem fet anteriorment, les parades i estacions properes a les localitzacions preferides i, les estacions que es van inaugurar l'any de naixement de l'usuari de l'aplicació.

```
1  a)Les meves localitzacions
2  b)Historial de localitzacions
3  c)Les meves rutes
4  d)Parades i estacions preferides
5  e)Estacions inaugurades el meu any de naixement
6  f)Tornar al menú principal
7
8  Selecciona opció:
```

Snippet 4: Menú de l'opció 1

Aquest menú no es deixarà de visualitzar fins que l'usuari decideixi sortir de l'apartat de *Gestió d'Usuari* (amb l'opció *f*). Un cop es surti, es mostrarà el menú principal.

S'haurà de comprovar que l'usuari introdueixi una **lletra entre la a i la f**. En cas que s'introdueixi una altra opció, es mostrarà un missatge d'error explicatiu.

2.3.1.1 Opció a: Les meves localitzacions

En aquest apartat, l'usuari podrà veure totes les localitzacions que hagi creat anteriorment i crear una nova localització. Les localitzacions es mostraran llistades per pantalla, on apareixerà el nom que li hagi assignat. Per crear una nova localització, s'introduirà el nom, les coordenades i una descripció.

Demanarem si es vol crear una nova localització o no i, en cas que s'esculli *no*, es mostrarà el menú de l'opció 1. Si es *sí*, s'haurà d'omplir la informació sobre la localització preferida i es tornarà a mostrar la llista de totes les localitzacions creades.

Si s'introdueix una opció no vàlida mostrarem un missatge d'error explicatiu. A més, com que la majoria d'usuaris seran enginyers i sabem que l'ortografia no és el seu punt fort, també acceptarem *sí* sense accent.

Haurem de vigilar que el nom de la localització no sigui el mateix que el d'un lloc que ja s'hagi creat per l'usuari anteriorment o estigui al fitxer *localitzacions.json*. A més, la longitud i la latitud hauran de ser coherents, **seguint el sistema de coordenades en EPSG:4326**. En cas que el nom ja existeixi o les coordenades estiguin malament, es mostrarà un missatge d'error i es tornarà a preguntar fins que l'usuari l'introdueixi bé.

A l'hora de modelar les classes de l'aplicació i les seves relacions, haurem de tenir en compte que tant el *DataModel* com l'usuari, contindran el llistat de les localitzacions creades.

```
1  No tens cap localització creada.
2
3  Vols crear una nova localització? (sí/no)
4  d1
5
6  Error! S'ha d'introduir "sí" o "no".
7  Vols crear una nova localització? (sí/no)
8  si
9
10 /*Aquí hi hauria l'Snippet 6 (no printeu aquesta línia)*/
```

Snippet 5: Les meves localitzacions

```
1  Nom de la localització:
2  University
3
4  Error! Aquesta localització ja existeix.
5  Nom de la localització:
6  La Salle Campus Barcelona
7
8  Longitud:
9  2.130064
10
11 Latitud:
12 41.408385
13
14 Descripció:
15 Universitat on em treuré el títol d'enginyer en 4 anys.
16
17 La informació s'ha registrat amb èxit!
18
19 -La Salle Campus Barcelona
20
21 Vols crear una nova localització? (sí/no)
```

Snippet 6: Introduïm una nova localització i tornem a mostrar el llistat

2.3.1.2 Opció b: Historial de localitzacions

Totes les localitzacions que s'hagin buscat a l'opció 2 del menú principal, s'hauran de mostrar per pantalla, sempre i quan aquestes s'hagin trobat. Aquestes estaran ordenades segons el moment que es van buscar (quan més amunt de la llista estigui la localització, més recent haurà estat la recerca d'aquesta).

En cas que encara no hi hagi cap localització que s'hagi buscat, mostrarem un missatge.


```
1 Localitzacions buscades:
2   -La Pedrera
3   -Laberint d'Horta
4   -El Pirulí
5   -Torres Bessones
6   -Els búnkers
```

Snippet 7: Historial de localitzacions b1

```
1 Encara no has buscat cap localització!
2 Per buscar-ne una, accedeix a l'opció 2 del menú principal.
```

Snippet 8: Historial de localitzacions b2

2.3.1.3 Opció c: Les meves rutes

Totes les rutes que s'hagin creat en l'opció 3 del menú principal, s'hauran de guardar perquè després puguin ser mostrades numerades en aquest apartat. Més endavant es donaran més detalls sobre com planejar les rutes, la informació que demanarem a l'usuari i d'on surt el resultat que es mostra per pantalla.

En cas que no s'hagi planejat cap ruta encara, es mostrarà un missatge.

```
1 Encara no has realitzat cap ruta :(
2 Per buscar-ne una, accedeix a l'opció 3 del menú principal.
```

Snippet 9: Les meves rutes Error

```
1  ->Ruta 1:
2  -Origen: Old House
3  -Destí: 41.432173, 2.174134
4  -Dia de sortida: 11-30-2019 a les 17:30pm
5  -Màxima distància caminant: 800 metres
6  -Combinació més ràpida:
7      Temps del trajete: 25 min
8      Origen
9      |
10     caminar 7 min
11     |
12     L5 Sagrada Família(523) -> Virrei Amat(529) 9 min
13     |
14     caminar 5 min
15     |
16     Destí
```

Snippet 10: Les meves rutes

2.3.1.4 Opció d: Parades i estacions preferides

Un cop s'hagin seleccionat localitzacions preferides, en aquesta opció es mostraran totes les parades de bus i estacions de metro que estiguin en un radi de 0.5 km de cada localització.

Les parades preferides podran ser tipus *BUS* o tipus *METRO*. Aquestes apareixeran ordenades segons la proximitat a la localització preferida i, es mostra el nom de la parada i el seu codi.

En aquesta opció farem servir l'API de TMB, on haureu de recorre dos JSON per trobar les parades i les estacions que estiguin com a màxim a 500 m (500 inclòs). Per trobar la informació sobre aquestes, **s'haurà de fer servir l'apartat de *Transit* de l'API.**

A continuació podeu veure com seria el resultat, tenint en compte que a l'opció 2 hem buscat la localització creada anteriorment (Snippet 6) i l'hem assignat com a preferida i, també hem afegit una localització preferida del json, la qual no es troba a prop de cap parada ni de cap estació.

1	-La Salle Campus Barcelona
2	1) Bellesguard - Immaculada (2437) BUS
3	2) Jesús Maria - Quatre Camins (2440) BUS
4	3) Jesús Maria - Pg Sant Gervasi (2441) BUS
5	4) Bellesguard - Valeta d'Arquer (2438) BUS
6	5) Pg Bonanova - Mandri (995) BUS
7	6) Pg Sant Gervasi - Pl Bonanova (936) BUS
8	7) Pg Bonanova - Mandri (996) BUS
9	8) Balsareny - Vista Bella (2439) BUS
10	9) Pl Bonanova (1360) BUS
11	10) Ronda de Dalt - Bellesguard (71) BUS
12	11) Esperança - Vilana (2433) BUS
13	12) Mandri - Bigai (2343) BUS
14	13) Cister - Melilla (2431) BUS
15	14) Pg Sant Gervasi - Folgueroles (97) BUS
16	15) Muntaner - Pl Bonanova (612) BUS
17	16) Rotonda de Bellesguard (1540) BUS
18	17) Quatre Camins - Teodor Roviralta (2432) BUS
19	18) Benedetti - Ronda de Dalt (74) BUS
20	-Miami Beach
21	TMB està fent tot el possible perquè el bus i el metro arribin fins aquí.

Snippet 11: Parades i estacions preferides d1

També haurem de vigilar en cas que no s'hagi escollit cap localització preferida, s'haurà de mostrar un missatge.

1	Per tenir parades i estacions preferides es requereix haver creat una localització preferida anteriorment.
---	--

Snippet 12: Parades i estacions preferides d2

Opcionalment, podem mostrar per cada parada i estació les diferents línies que paren. Per exemple, en la parada *Bellesguard - Immaculada*, després de *BUS* apareixeria *123*.

2.3.1.5 Opció e: Estacions inaugurades el meu any de naixement

Finalment, en l'últim apartat del menú *Gestió d'Usuari*, podrem veure el nom de totes aquelles estacions de metro que s'hagin inaugurat en l'any de naixement que hagi introduït l'usuari. Per fer-ho, **s'haurà de fer servir l'apartat de Transit de l'API**.

A més, s'haurà de mostrar el nom de la línia de la qual forma part l'estació. Per exemple, l'estació *Trinitat Nova* és de la línia amb codi de línia 4 i amb nom de línia *L4*. Us recomanem que useu el llistat de *Línies de Metro* de l'API per saber l'equivalència entre el codi i el nom de les diverses línies que hi ha. **No si val hardcodejar el codi ficant una L davant del codi de línia**.

```
1 Estacions inaugurades el 1999:  
2 -Trinitat Nova L4
```

Snippet 13: Estacions inaugurades el meu any de naixement e1

Si no hi ha cap estació inaugurada el mateix any, es mostrarà un missatge per pantalla.

```
1 Cap estació de metro es va inaugurar el teu any de naixement :(
```

Snippet 14: Estacions inaugurades el meu any de naixement e2

2.3.2 OPCió 2: BUSCAR LOCALITZACIONS

Aquesta opció permetrà buscar tant els llocs del fitxer *localitzacions.json* com els que hagi creat l'usuari. Juntament amb el nom de la localització, la posició i la descripció, en el cas que sigui un monument, un hotel o un restaurant, es mostrarà més informació.

En cas que no existeixi el nom de la localització, es mostrarà un missatge d'error per pantalla.

```
1 Introduueix el nom d'una localització:
2 König Barcelona Centre
3
4 Posició: 41.387664, 2.167685
5 Descripció:
6 Restaurant perfecte per anar a prendre unes braves i unes hamburgueses després d'haver vist a l'esecretary
7 el suspès del punt de control de mates. Potser no ets sentiràs millor però segur que se't passa la gana.
8 Característiques: Menjar fins a la matinada · Esmorzar · Terrassa exterior
9
10 /*Aquí es mostraria l'Snippet 17 (no printeu aquesta línia)*/
```

Snippet 15: Buscar localitzacions 2a

```
1 Introduueix el nom d'una localització:
2 Plaça Elsa Capunta
3
4 Ho sentim, no hi ha cap localització amb aquest nom.
```

Snippet 16: Buscar localitzacions 2b

Després d'haver trobat el lloc, es demanarà a l'usuari si vol assignar aquesta localització com a preferida preguntant *sí* o *no*, on com hem vist ja més d'un cop, haurem de tenir en compte si l'usuari ens introdueix una opció vàlida.

Un cop haguem assignat una localització com a preferida, de la relació entre l'usuari i les localitzacions, sorgirà l'entitat localització preferida. Aquesta contindrà la data de quan hem seleccionat el lloc com a preferit i el tipus que li ha assignat l'usuari. El tipus haurà de ser un dels 5 especificats. En cas que l'usuari introdueixi un altre cosa, es mostrarà error.

Quan l'usuari hagi respost correctament, es tornarà al menú principal.

```
1 Vols guardar la localització trobada com a preferida? (sí/no)
2 Sí
3
4 Tipus(casa/feina/estudis/oci/cultura):
5 jalar
6
7 Error! S'ha d'introduir "casa", "feina", "estudis", "oci" o "cultura".
8 Tipus(casa/feina/estudis/oci/cultura):
9 oci
10
11 König Barcelona Centre s'ha assignat com a una nova localització preferida.
```

Snippet 17: Buscar localitzacions 2c

2.3.3 OPCIO 3: PLANEJAR RUTA

En aquest apartat, a partir de dos punts, un dia de sortida i la màxima distància caminant, mostrarem la ruta més ràpida per arribar fins al destí. Per fer-ho, **s'haurà de fer servir l'apartat de Planner de l'API**. En relació als punts que introduïrem, podran ser coordenades o localitzacions del *DataModel*.

Com a resposta, direm el temps total i els diferents trams de la ruta, on s'indicarà si es fa caminant o amb una línia de transport. En cas que sigui una línia, haurem de dir el nom de la ruta i el nom i el codi de les parades d'inici i final. També haurem de dir quant de temps es triga a fer cada tram. Tant el temps total com el de cada tram, es mostrarà **fent una aproximació a minuts enters**.

Si ens fixem a l'API, ens demanen moltes més dades per calcular la millor ruta en transport públic. Tots els paràmetres que no siguin els mencionats anteriorment, els suposarem i sempre seran els mateixos. Per exemple, podem fer que el paràmetre *mode* sempre sigui *TRANSIT,WALK*.

Haurem de vigilar que les coordenades siguin reals i si s'introdueix el nom d'una localització aquesta existeixi (pot ser del fitxer json o creada per l'usuari). En cas contrari, mostrarem un missatge d'error. La resta de dades no les haurem de controlar des del sistema, ja que si hi ha una dada incorrecta la crida a l'API donarà error. En aquest cas, es mostrarà també un missatge per pantalla. També si les dades estan ben introduïdes però no arriba el servei de metro i bus a una de les localitzacions, mostrarem un missatge en concret.

Per saber si esteu mostrant correctament les dades de la ruta, podeu comprovar-ho usant [el web de TMB l'apartat Vull anar](#), així veureu si aquesta mostra un resultat semblant.

```
1  Origen? (lat,lon/nom localització)
2  Eurovegas
3
4  Ho sentim, aquesta localització no és vàlida :(
5  Origen? (lat,lon/nom localització)
6  Old House
7
8  Destí? (lat,lon/nom localització)
9  41.432173, 2.174134
10
11 Dia/hora seran de sortida o d'arribada? (s/a)
12 idk
13
14 Error! S'ha d'introduir "s" o "a"!
15 Dia/hora seran de sortida o d'arribada? (s/a)
16 s
17
18 Dia? (MM-DD-YYYY)
19 11-30-2019
20
21 Hora? (HH:MMam/HH:MMpm)
22 17:30pm
23
24 Màxima distància caminant en metres?
25 800
```

Snippet 18: Planejar Ruta 3a Part 1

```
1 Combinació més ràpida:
2   Temps del trajecte: 25 min
3   Origen
4   |
5   caminar 7 min
6   |
7   L5 Sagrada Família(523) -> Virrei Amat(529) 9 min
8   |
9   caminar 5 min
10  |
11  Destí
```

Snippet 19: Planejar Ruta 3a Part 2

```
1 /*Suposem que hem ficat el dia 11-31-2019*/
2
3 Error, hi ha algun paràmetre erroni :(
4
5 /*Tornem a fer les preguntes de l'Snippet 18*/
```

Snippet 20: Planejar Ruta 3b


```
1  /*Suposem que hem ficat a origen Miami Beach*/  
2  
3  TMB està fent tot el possible perquè el bus i el metro facin aquesta ruta en un futur.  
4  
5  /*Mostrarem el menú principal (no printeu aquesta línia)*/
```

Snippet 21: Planejar Ruta 3c

2.3.4 OPCIÓ 4: TEMPS D'ESPERA DEL BUS

En aquesta opció haurem de demanar un codi de parada i mostrar el temps d'espera en minuts i la destinació de cada línia que pari en la parada introduïda. Les línies que apareguin estaran ordenades en funció del temps que falta perquè passi el bus. Per fer-ho, **s'haurà de fer servir l'apartat d'iBus de l'API**.

Si el codi de parada correspon a una de les parades preferides (Snippet 11), s'haurà d'indicar quan es mostri el resultat. També si s'introdueix un codi de parada no vàlid, s'haurà de mostrar un missatge d'error per pantalla i tornar a demanar un codi correcte. Un cop s'hagi mostrat el resultat, tornarem al menú principal.

Per verificar que la informació que esteu mostrant per pantalla és correcta, podeu accedir al [web de TMB](#) l'apartat d'*iBus* per comprovar-ho.

```
1  Introdueix el codi de parada:  
2  3369  
3  
4  H8 - Camp Nou - 3 min  
5  47 - Pg. Marítim - 7 min
```

Snippet 22: Temps d'espera del bus 4a

```
1 Introdueix el codi de parada:
2 1360
3
4 Parada preferida!
5 H4 - Bon Pastor - 2 min
6 V13 - Av. Tibidabo - 5 min
7 75 - Av. Tibidabo - 6 min
```

Snippet 23: Temps d'espera del bus 4b

```
1 Introdueix el codi de parada:
2 Av. Elena Nito del Bosque - Pl. de la Débora Dora
3
4 Error, codi de parada no vàlid!
5 Introdueix el codi de parada:
```

Snippet 24: Temps d'espera del bus 4c

2.4 CRIDES A WEBSERVICES

Per obtenir tota la informació per mostrar-la per pantalla i guardar-la al nostre sistema, requerirem del webservice de TMB.

Un webservice no és més que una interfície a la que podem accedir on amb (o sense) certs paràmetres, i on podem obtenir la informació que requerim.

Els webservices amb que tractarem segueixen el patró REST, que es distingeixen per retornar dades en format JSON (o similars), entre d'altres característiques que podeu investigar si us interessa el tema.

Trobareu més informació al document auxiliar, com es comentarà a l'apartat 3.5.

3 CONSIDERACIONS

A continuació s'expliquen uns punts que haureu de tenir presents abans de començar el desenvolupament de la pràctica.

3.1 PROCÉS DE DESENVOLUPAMENT

Abans de posar les mans sobre el teclat i començar a codificar, penseu. Analitzeu què cal implementar i dissenyeu quines classes i amb quines relacions tindrà el sistema, agafeu paper i llapis o utilitzeu l'eina StarUML per tal de crear un diagrama de classes UML.

Segurament aquest diagrama no serà el definitiu, i anirà canviant a mida que avanceu la pràctica, però d'aquesta manera assentareu unes bases pel desenvolupament del sistema i us estalviareu hores de codificació i mals de cap.

El desenvolupament d'un projecte informàtic real típicament comprèn les fases: presa de requeriments, especificació, anàlisi, disseny, implementació, proves i implantació. Podeu pensar que les fases inicials d'aquesta seqüència ja han estat realitzades per l'equip de l'assignatura i que ara us passem el relleu per tal que acabeu el sistema. Ara us toca acabar d'analitzar, de dissenyar, implementar i provar el sistema abans de ser usat en entorns reals.

3.2 DISSENY DE CLASSES

El disseny de classes que feu és summament important, cal que elaboreu un modelat de classes que presenti una alta cohesió (que cada classe s'encarregui d'una única tasca conceptual, per exemple, una classe per les connexions al servidor, una classe pel menú, etcètera.) i que feu ús de les característiques i els avantatges que us ofereix el paradigma de la programació orientada a objectes (incloent, si aplica, herència i polimorfisme). Si feu un mal ús del paradigma de l'orientació a objectes la pràctica no serà aprovada.

3.3 EMMAGATZEMAMENT DE DADES

Referent a l'emmagatzemament de les dades, no és necessari que codifiqueu les estructures de dades. Podeu usar les classes del paquet `java.util` del SDK de Java, com les classes `LinkedList<E>` i `ArrayList<E>`, entre d'altres.

D'aquesta manera obtindreu, "gratuïtament", estructures de dades amb funcionalitats interessants, lliures d'errors i ràpides, amb el que agilitzareu molt el procés de codificació de la pràctica.

3.4 LECTURA DE FITXERS JSON I API'S

Per la lectura del fitxers JSON podeu usar les classes que ofereix la llibreria de Google per aquesta finalitat. No cal que extraieu la informació dels fitxers tractant-los com si fossin fitxers de text pla. Busqueu informació sobre l'analitzador sintàctic o parser Gson.

Altrament, també podeu utilitzar altres llibreries com ara JSON-Java per parsejar els fitxers. Podeu utilitzar la llibreria que preferiu. Tingueu en compte que, si bé interessa llegir totes les dades d'un fitxer local, sovint no cal desar tots els resultats de webservices i molts cops amb tractar-los com a JsonObject (o equivalents) i entrar manualment als diferents camps ja és suficient.

3.5 API DE TMB

En aquesta pràctica haurem d'usar l'API de TMB. Per accedir-hi, haurem d'anar al [Portal de Desenvolupador de TMB](#) i registrar-nos amb una de les diferents maneres que ens ofereixen. A continuació, haurem de crear una nova aplicació per obtenir l'identificador i la clau, que ens permetran fer les crides per obtenir la informació.

L'API està dividida en quatre serveis: *Transit*, *Planner*, *iBus* i *Static*.

El servei de *Transit* l'usarem en l'apartat de *Les meves parades i estacions*, en la primera opció del menú principal. Dels 4 mètodes amb les seves diferents variants que hi ha, haureu d'usar aquells que creieu útils per mostrar el nom de les parades, les estacions i, en cas que feu l'apartat opcional, les línies que hi paren.

El servei de *Planner* ens servirà en l'opció de *Planejar ruta*, on amb les dades introduïdes per l'usuari podrem trobar els diferents recorreguts proposats per l'API. En aquest apartat, haurem de tenir especial cura a l'hora de tractar les dades, ja que ens demanen un format per cadascuna molt específic.

Finalment, el servei de *iBus* serà essencial per obtenir la informació per l'opció *Temps d'espera del bus*.

L'API conté molt bones explicacions de com fer les crides i de com està estructurat el contingut que ens retorna. L'únic inconvenient que hi podrem trobar és l'excés d'informació que obtindrem. Per això, us proposem una manera per llegir els JSON de l'API de manera segura. Tot està explicat detalladament al document addicional que acompanya l'enunciat de la pràctica.

4 LLIURAMENT

Dipositar al pou corresponent de l'eStudy un arxiu comprimit **.ZIP o .RAR** amb el login dels membres del grup separats per barra baixa, seguint el format **P1_<login1>_<login2>.zip**. Aquest fitxer comprimit ha de contenir 2 carpetes, una anomenada **TMBJson** i una altra amb el nom **Documentació**. Pel que fa al seu contingut:

- **TMBJson:** Carpeta del projecte generat amb l'IntelliJ IDEA. Ha d'incloure també la carpeta doc que conté el javadoc.
- **Documentació:** Memòria en format PDF i arxius del projecte de l'*StarUML* (.mdj).

La memòria ha de contenir els apartats següents:

- Portada (amb el nom complet i login dels dos membres del grup)
- Índex numerat (apartats i número de pàgina que ocupa cadascun dins el document)
- Resum de l'enunciat (no copiar de l'enunciat, explicar breument amb paraules vostres)
- Diagrama de classes UML de disseny (el diagrama que finalment heu implementat) *
- Explicació de les classes dissenyades (atributs, mètodes i aspectes més rellevants de cadascuna)
- Opcionals i extres (explicació de tot allò que heu implementat que no sigui estrictament necessari o no estigui a l'enunciat, si és el cas)
- Mètode de proves utilitzat (explicació del mètode de proves que heu seguit)
- Dedicació (estimació detallada del temps emprat per la realització de la pràctica)
- Conclusions (a nivell personal i a nivell tècnic)
- Bibliografia (segons la norma ISO 690:2010)

* En el diagrama de classes han d'aparèixer tots els atributs i mètodes de les classes, excepte els *getters* i *setters*.

La bona redacció i el bon ús de la gramàtica serà valorat positivament a l'hora de posar la nota de la memòria.

5 NORMATIVA

Grups de dues persones. Només una de les dos ha de realitzar l'entrega via eStudy.

Eines:

- Java com a llenguatge de programació orientat a objectes.
- javadoc per comentar el codi i generar la documentació del programari.
- IntelliJ com a entorn de desenvolupament integrat (IDE) per codificar l'aplicació.
- StarUML per l'elaboració de diagrames UML.

Entrega sobre 10
05/01/2020
Entrega sobre 8
03/05/2020
Entrega sobre 6
28/06/2020

EL NO COMPLIMENT D'ALGUN DELS PUNTS ESPECIFICATS EN AQUESTS APARTATS SUPOSARÀ LA NO ACCEPTACIÓ DE LA PRÀCTICA

LA DETECCIÓ DE COPIA COMPORTARÀ SUSPENDRE TOTS ELS INTEGRANTS DELS GRUPS IMPLICATS, TANT ELS QUE HAN COPIAT COM ELS QUE HAN ESTAT COPIATS, I S'APLICARÀ LA NORMATIVA DE L'ASSIGNATURA
