

LSBicing

Practica 1 DPO

Albert Martínez Oliver (albert.martinez.2015)

Marc Segura Duran (marc.segura.2015)

INDEX

Resum de l'enunciat	-----	3
Diagrama de classes UML de disseny	-----	4
Explicació de les classes dissenyades	-----	5
Mètode de proves utilitzat	-----	9
Dedicació	-----	10
Conclusions	-----	11
Bibliografia	-----	12

Resum de l'enunciat

La pràctica consisteix en la creació d'una aplicació que creï rutes amb el servei Bicing de Barcelona fent servir google maps.

Resum de l'objectiu de les funcions:

Opció 1: Ha de geolocalitzar l'adreça que l'usuari introdueixi i mostrar-ho a google maps.

Opció 2: Ha de trobar l'estació de bicing més pròxima a la ubicació que l'usuari introdueixi.

Opció 3: Ha de crear una ruta entre dues direccions distingint les parts en bici de les parts caminant, ja que l'usuari haurà d'anar a l'estació més pròxima amb bicicletes i arribar a l'estació de destí més pròxima.

Opció 4: Ha de mostrar una imatge amb totes les estacions que tinguin un mínim de bicicletes que demana l'usuari.

Opció 5: L'usuari escriu un tipus de negoci o servei i es mostren com a màxim 5 resultats donant-li l'opció de guardar-ne un.

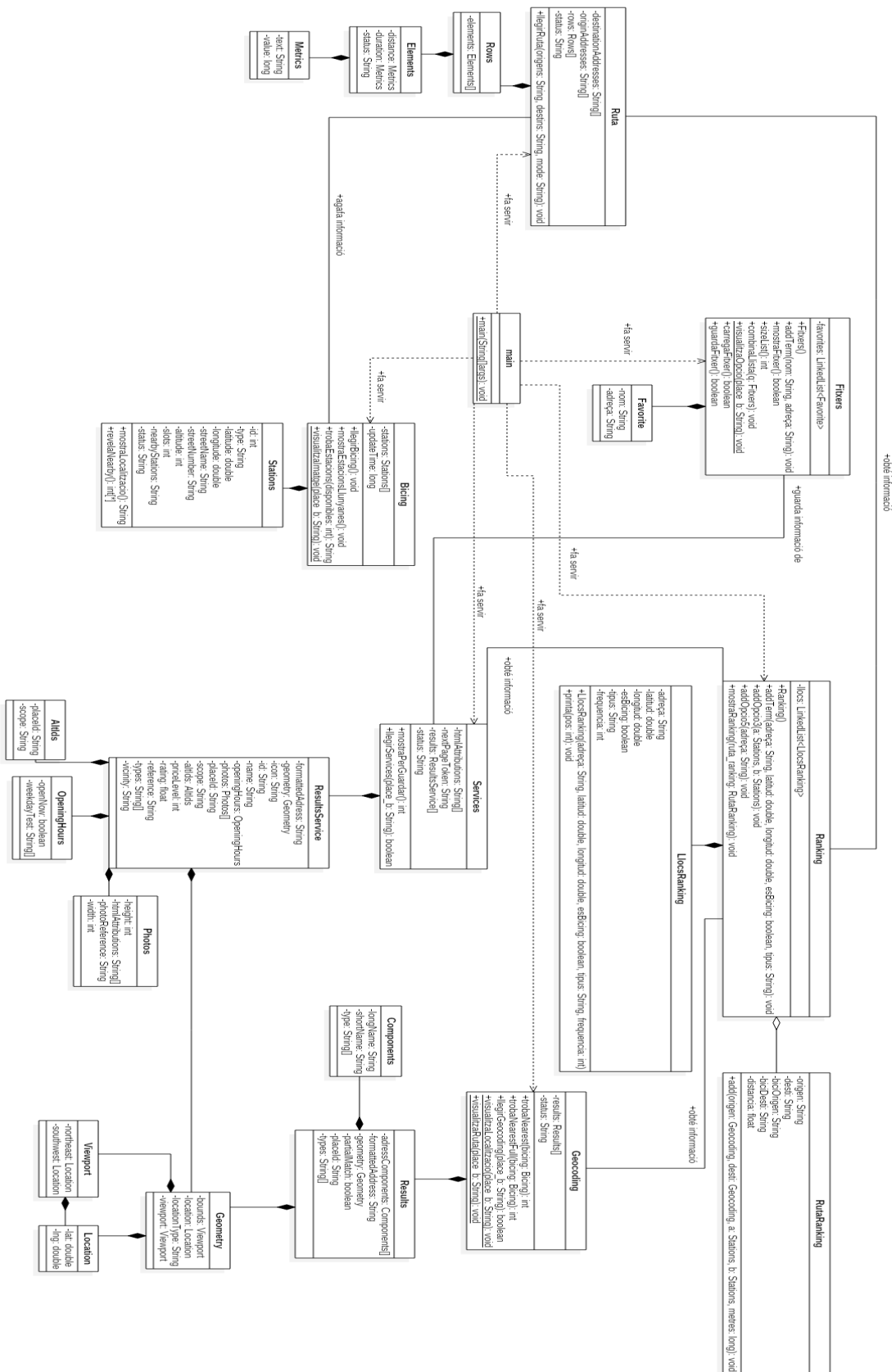
Opció 6: Es mostren totes les ubicacions guardades, l'usuari en selecciona una i es mostra la ubicació a google maps.

Opció 7: Calcula les dues estacions que estan més llunyanes i mostra la informació d'aquestes.

Opció 8: Mostra el rànquing de les estacions de bicing més emprades i de els llocs més guardats. També mostra les dades de la ruta cercada amb més distància.

Opció 9: Surt del programa.

Diagrama de classes UML de disseny



Explicació de les classes dissenyades

Ruta: Es la classe on s'emmagatzema la informació extreta de l'api de google de les rutes. Per tant els atributs que conté són els extrems del fitxer Json d'aquesta. A continuació s'expliquen els mètodes:

LlegirRuta = Aquest mètode llegeix de l'api de google i emmagatzema la informació en els atributs de la classe.

Rows: Aquesta classe és la que conté la informació dels elements de la ruta cercada en la classe "Ruta".

Elements: Aquesta classe és la que conté les dades de distancia, duració i estat de una de les rutes trobades a l'api de google de l'array de l'atribut de la classe de "Rows".

Metrics: Conté la informació d'un valor com el que tenen els atributs "distance" i "duration" de la classe "Elements".

Fitxers: Conte la llista de la informació de les ubicacions del fitxer favorite_places.json i guarda i carrega d'aquest fitxer. A continuació s'expliquen els mètodes:

Fitxers: és el constructor de la llista.

addTerm: afegeix un element a la llista.

mostraFitxer: mostra totes les ubicacions guardades i deixa a l'usuari que n'esculli una. Quan l'usuari en selecciona una, visualitza la ubicació a google maps.

sizeList: retorna els elements de la llista.

combinaLlista: combina dues llistes en una.

visualitzaOpcio: carrega una ubicació a google maps.

carregaFitxer: carrega la informació del fitxer json i ho introdueix a la llista.

guardaFitxer: guarda la informació de la llista en el fitxer json.

Favorite: Aquesta classe és la que conté les dades dels llocs preferits de l'usuari. Guardant el nom i l'adreça.

Bicing: Conte la informació extreta de la pàgina oficial de Bicing de Barcelona. A continuació s'expliquen els mètodes utilitzats:

`llegirBicing`: carrega la informació desde la pàgina de bicing i la introdueix als atributs de la classe.

`mostraEstacionsLlunyanes`: calcula les estacions més llunyanes de la base de dades i mostra les seves dades per pantalla.

`trobaEstacions`: Busca quantes estacions tenen el mínim de bicicletes introduïdes per l'usuari en l'opció 4.

`visualitzalimatge`: Mostra la imatge amb totes les estacions de bicing que es vulguin mostrar en aquesta. Utilitza la api de google que retorna una imatge.

Stations: Conté tota la informació d'una estació de bicing extreta de la pàgina de bicing. Conté un identificador, la direcció amb l'adreça, latitud, longitud i altitud, les estacions més properes, el tipus d'estació de la que és tracta i les bicicletes i espais buits que té i per últim si esta oberta o no l'estació. A continuació s'expliquen els mètodes utilitzats:

`mostraLocalitzacio`: mostra la localització, la quantitat de bicicletes i la quantitat d'espais possibles de l'estació de bicing.

`revelaNearby`: transforma el String de `nearbyStations` en una array de int's.

Ranking: Conté una llista amb els llocs que s'han guardat i les estacions emprades en l'execució en les rutes per a crear un rànkung dels 10 més emprats. A continuació s'exposen els mètodes utilitzats amb la seva explicació corresponent:

`Ranking`: crea la llista de llocs pel rànkung.

`addTerm`: afegeix un lloc a la llista. Si aquest lloc ja existeix, li augmenta en 1 la freqüència en el objecte.

`addOpcio3` : afegeix les estacions de bicing emprades en una ruta de l'opció 3.

`addOpcio5`: afegeix a l llista una ubicació que s'ha seleccionat per guardar en el fitxer json.

mostraRanking: mostra els 10 llocs més emprats en l'execució i la ruta més llarga cercada en l'opció3 durant l'execució.

LlocsRanking: Conté la informació necessària dels llocs que es necessiten per fer el rànquing dels 10 llocs més emprats durant l'execució. A més de la freqüència amb la que s'ha buscat el lloc. A continuació s'expliquen els mètodes emprats:

LlocsRanking: crea un lloc.

Printa: mostra les dades del lloc.

RutaRanking: Conté la informació necessària per a la posterior mostra de la ruta més llarga cercada per l'usuari. A continuació es mostren els mètodes emprats:

add: afegeix la informació d'una ruta si és més llarga que l'anteriorment guardada.

Geocoding: Conté la informació extreta d'una ubicació des de l'api de google maps. A continuació s'expliquen els mètodes que s'han creat:

trobaNearest: busca l'estació de bicinç més pròxima a la ubicació de la direcció introduïda a la classe i retorna l'identificador de l'estació de bicinç més propera.

trobaNearestFull: fa el mateix que trobaNearest amb la diferència que l'estació de bicinç que retorna no pot estar buida.

llegirGeocoding: carrega l'informació d'una ubicació des de l'api de google i la introdueix als atributs de la classe.

visualitzaLocalitzacio: redirigeix a google maps mostrant una localització.

visualitzaRuta: mostra una ruta a la pàgina de google maps.

Results: conté la informació dels diferents resultats que dona una cerca d'una ubicació.

Components: conté els diferents components d'informació sobre l'adreça que google ens facilita a través de la seva api.

Geometry: conté la informació de la localització de la ubicació.

Location: mostra la latitud i longitud d'una ubicació.

Viewport: mostra la ubicació de la cantonada superior dreta i la cantonada inferior esquerra del viewport.

Services: emmagatzema tota la informació extreta de l'api de serveis de google. A continuació s'expliquen els diferents mètodes implementats:

mostraPerGuardar: mostra com a màxim 5 serveis i dona la opció a l'usuari de guardar-ne un al fitxer json.

llegirServices: carrega la informació de la api de serveis de google i les introdueix als atributs de la classe.

ResultsServices: conté tota la informació sobre cadascun dels locals que s'extreuen de la api de google.

AltIds: conté l'identificador del local.

OpeningHours: mostra si està obert o no el local.

Photos: mostra la informació de les fotografies d'un local.

Main: és la classe principal que executa totes les opcions i utilitza totes les classes.

Mètode de proves utilitzat

Per comprovar els possibles errors que tenia el programa hem seguit el següent mètode:

Al acabar d'escriure una opció, es provaven tots els possibles errors explicats durant les sessions de classe en el curs (variable incorrecte, error en les dades, carrega d'arxius defectuosa...).

En el cas de que hi hagués algun error, s'intentava solucionar utilitzant mètodes del "try" o si l'error era del propi script, solucionant-lo.

Dedicació

Les hores aproximades dedicades a aquesta pràctica que hem calculat són unes 30 per fer l'script del programa, unes 5 hores pel diagrama de classes amb els corresponents canvis al llarg de la implementació del programa. I per últim unes 2 hores en la realització d'aquesta memòria.

Conclusions

Aquesta pràctica ens ha servit per consolidar tots els coneixements teòrics apresos a les sessions del curs. Si més no és la primera pràctica en la que fem servir arxius a temps real d'una gran aplicació com seria google maps, la qual cosa, ha incrementat les ganes de realitzar aquesta.

Al principi de la realització de la pràctica ens vam sentir una mica perduts per la poca informació rebuda al respecte. Tot i això se'ns van brindar totes les eines de cerca necessàries i per tant poc a poc vam anar aprenent a utilitzar tot el necessari per realitzar la pràctica.

Bibliografia

Geocoding api [en línia] [data de consulta: 19 Desembre 2016]. Disponible a: <https://developers.google.com/maps/documentation/geocoding/start>

Distance matrix api [en línia] [data de consulta: 23 Desembre 2016].

Disponible a:

<https://developers.google.com/maps/documentation/distance-matrix/intro?hl=es->

Places api web service [en línia] [data de consulta: 23 Desembre 2016]

Disponible a : <https://developers.google.com/places/web-service/?hl=es-419>

Static Maps api [en línia] [data de consulta: 29 Desembre 2016] Disponible

a: <https://developers.google.com/maps/documentation/static-maps/intro?hl=en->

StackOverflow qüestions [en línia] [data de consulta: 2 Gener 2017]

Disponible a: <http://stackoverflow.com/questions/2220400/how-do-i-make-my-string-comparison-case-insensitive>

Google Maps [en línia] [data de consulta: 15 Desembre 2016] Disponible a:

<https://www.google.es/maps>

Json Parser Online [en línia] [data de consulta: 29 Desembre 2016]

Disponible a: <http://json.parser.online.fr/>

GitHub [en línia] [data de consulta: 29 Desembre 2016] Disponible a:

<https://github.com/google/gson/blob/master/UserGuide.md>