Ontologia sobre el format obert GTFS aplicada al Consorcio Regional de Transportes de Madrid

Autor: Alonso López i Vicente

Tutor: Felipe Geva Urbano

Professor: Ferran Prados Carrasco

Grau en Enginyeria Informàtica

Web Semàntica

10/01/2019

Crèdits/Copyright

FITXA DEL TREBALL FINAL

Títol del treball:	*Ontologia sobre el format obert GTFS aplicada al Consorcio Regional de Transportes de Madrid*
Nom de l'autor:	*Alonso López i Vicente*
Nom del col·laborador/a docent:	*Felipe Geva Urbano*
Nom del PRA:	*Ferran Prados Carrasco*
Data de lliurament (mm/aaaa):	*01/2019*
Titulació o programa:	*Grau d'Enginyeria Informàtica*
Àrea del Treball Final:	*Web Semàntica*
Idioma del treball:	*Català*
Paraules clau	*Web Semàntica, Ontologia, Transport públic, GTFS*

Resum del Treball (màxim 250 paraules): *Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball*

La World Wide Web, que, en un principi, es va pensar per compartir documents i informació entre persones, ha anat acumulant cada vegada més dades i continguts fins a unes dimensions que, a la pràctica, estan fora de l'abast dels éssers humans com a individus. És per això que cal un sistema per compartir les dades a la web pensat per a que les màquines puguin llegir-les, gestionar-les i presentar-les. La Web Semàntica i els seus estàndards donen resposta a aquesta necessitat, etiquetant les dades per tal que siguin tractades per sistemes automàtics.
D'altra banda, emmarcat en el moviment Open Data, cada vegada s'ofereixen més dades per compartir lliurement per part dels particulars i, sobre tot, per part de les administracions públiques. Aquestes dades es presenten en fitxers de diferents formats per ser descarregats o bé mitjançant aplicacions interactives. GTFS és un estàndard creat per Google per proporcionar les dades estàtiques d'una xarxa de transport.
Aquest treball crea una ontologia, és a dir, un conjunt de conceptes bàsics representatius amb els quals es pot modelar un domini de coneixement o tema, sobre una xarxa de transport públic a partir les dades proporcionades per un paquet GTFS. El resultat del treball permet obtenir respostes a preguntes sobre els elements de la xarxa de transport inclosa en el paquet mitjançant consultes SPARQL. El paquet utilitzat és un dels que proporciona el Consorcio Regional de Transportes de Madrid, però, al ser un estàndard, és pot aplicar sobre qualsevol paquet o paquets GTFS.

Abstract (in English, 250 words or less):

World Wide Web was created as a tool for sharing files and information between users. It has been collecting data and content and has finally reached a dimension that we can consider that for many of us is out of reach. That's why it was necessary to find a system that allow us to share the data on the web and designed for letting the machines to read, manage and present this data. Semantic Web and its standards gives a solution to that need, tagging the data in a way that they can be treated by automatic systems.
On the other hand, and as a part of the Open Data movement, there are more and more data openly shared mainly by public administrations but also by individuals, in files on different formats that can be downloaded or on interactive applications.
This paper creates an ontology about the public transport network based on the data given by one GTFS pack of the Consorcio Regional de Transportes de Madrid. GTFS is a standard created by Google that gives the static data of a transport network. The result of this work is an ontology that could be filled with data of any GTFS pack and which allows to obtain answers to the questions about the elements of the transport network inside the pack thanks to the SPARQL queries.

Dedicatòria

Per la Maria Lluïsa

Però hem viscut per salvar-vos els mots,
per retornar-vos el nom de cada cosa,...

Salvador Espriu
Inici de càntic en el temple

Agraïments

A tots aquells que han fet possible la UOC, i als que la continuen fent possible cada dia.
I a la meva família, la meva mare, les meves filles, la meva germana, que sempre m'han recolzat.

Resum

La World Wide Web, que, en un principi, es va pensar per compartir documents i informació entre persones, ha anat acumulant cada vegada més dades i continguts fins a unes dimensions que, a la pràctica, estan fora de l'abast dels éssers humans com a individus. És per això que cal un sistema per compartir les dades a la web pensat per a que les màquines puguin llegir-les, gestionar-les i presentar-les. La Web Semàntica i els seus estàndards donen resposta a aquesta necessitat, etiquetant les dades per tal que siguin tractades per sistemes automàtics.

D'altra banda, emmarcat en el moviment Open Data, cada vegada s'ofereixen més dades per compartir lliurement per part dels particulars i, sobre tot, per part de les administracions públiques. Aquestes dades es presenten en fitxers de diferents formats per ser descarregats o bé mitjançant aplicacions interactives. GTFS és un estàndard creat per Google per proporcionar les dades estàtiques d'una xarxa de transport.

Aquest treball crea una ontologia, és a dir, un conjunt de conceptes bàsics representatius amb els quals es pot modelar un domini de coneixement o tema, sobre una xarxa de transport públic a partir les dades proporcionades per un paquet GTFS. El resultat del treball permet obtenir respostes a preguntes sobre els elements de la xarxa de transport inclosa en el paquet mitjançant consultes SPARQL. El paquet utilitzat és un dels que proporciona el Consorcio Regional de Transportes de Madrid, però, al ser un estàndard, és pot aplicar sobre qualsevol paquet o paquets GTFS.

Abstract

World Wide Web was created as a tool for sharing files and information between users. It has been collecting data and content and has finally reached a dimension that we can consider that for many of us is out of reach. That's why it was necessary to find a system that allow us to share the data on the web and designed for letting the machines to read, manage and present this data. Semantic Web and its standards gives a solution to that need, tagging the data in a way that they can be treated by automatic systems.

On the other hand, and as a part of the Open Data movement, there are more and more data openly shared mainly by public administrations but also by individuals, in files on different formats that can be downloaded or on interactive applications.

This paper creates an ontology about the public transport network based on the data given by one GTFS pack of the Consorcio Regional de Transportes de Madrid. GTFS is a standard created by Google that gives the static data of a transport network. The result of this work is an ontology that could be filled with data of any GTFS pack and which allows to obtain answers to the questions about the elements of the transport network inside the pack thanks to the SPARQL queries.

Paraules clau

Web Semàntica, Ontologia, Transport públic, GTFS

Índex

Figures i taules

Índex de figures

Índex de taules

1. Introducció

1.1. Presentació

Amb la proposta de Tim Berners-Lee per utilitzar l'hipertext com a mecanisme per bescanviar informació a la xarxa, neix la World Wide Web l'any 1989. Gràcies a ella, i a les seves evolucions, cada vegada més dades es troben disponibles a la xarxa per tal que les persones les utilitzin, ja sigui directament o mitjançant aplicacions.

La web semàntica és un pas més en la manipulació de la informació disponible, un volum cada vegada més ingent. Així, més enllà del plantejament inicial de la WWW, pensada per bescanviar informació entre persones, la web semàntica cerca bescanviar la informació entre aplicacions, de manera que aquestes siguin capaces d'entendre el contingut, relacionar la informació i oferir respostes a les qüestions plantejades..

D'altra banda, les administracions, com a forma de transparència, cada vegada ofereixen més dades sense restriccions (open data) per tal que el ciutadà pugui accedir a elles. Un dels problemes que es presenten, però, és que, en general, les dades no estan estructurades de manera homogènia, i cada administració les presenta en formats diferents. Per tal d'unificar la presentació de la informació en el camp del transport col·lectiu, en 2006 Google va presentar GTFS, un format estructurat de dades de transport (horaris, parades, localització geogràfica) per tal que els operadors oferissin les seves dades de manera estandarditzada i aquesta es pogués utilitzar en aplicacions o directament per l'usuari final.

1.2. Context i justificació del treball

El treball es centra en una part de la web semàntica, com és la definició d'una ontologia a partir de les dades que proporciona el format GTFS-estàtic, la part de GTFS que ofereix dades sobre la infraestructura de transport que gestiona un operador, diferent del GTFS-temps real, que s'utilitza per comunicar prediccions d'arribada, posicions de vehicles i avisos de serveis en temps real. El resultat s'aplica sobre les dades que ofereix el Consorcio Regional de Transportes de Madrid (CRTM) en aquest format per tal de respondre preguntes sobre la xarxa de transport, com ara on està situada una estació, quins accessos té una estació o quina línia porta a l'estació X. Evidentment, aquesta ontologia podrà ser utilitzada sobre les dades de qualsevol paquet GTFS proporcionat per qualsevol dels molts operadors que els ofereixen.

1.3. Objectius

D'acord amb el Pla Docent de l'assignatura, els objectius del treball són els següents:

- Entendre què és una ontologia.
- Entendre què és l'Open Data.
- Analitzar els repositoris proporcionats.
- Obtenir una ontologia basada en la informació que contenen els repositoris.

- Implementar un script que permeti carregar les dades dels fitxers a la ontologia en format OWL/XML.
- Interrogar l'ontologia mitjançant SPARQL per obtenir informació.

Així, doncs, l'objectiu del treball és implementar una ontologia sobre l'estructura dels paquets GTFS, poblar l'ontologia amb les dades que proporciona el CRTM i, finalment, realitzar consultes SPARQL a l'ontologia.
A nivell personal, això em permetrà consolidar els coneixements apressos a l'assignatura Representació del Coneixement, i conèixer l'estat actual de la web semàntica i de les diferents parts de la seva arquitectura, en especial les que fan referència a la representació del coneixement, així com enfrontar la creació pràctica d'una ontologia, utilitzant les eines disponibles més adients.

1.4. Metodologia i procés de treball

Per realitzar el treball, primerament he realitzar una cerca àmplia dels conceptes teòrics, però també exemples d'aplicació pràctica d'ontologies, així com treballs en l'àmbit de la Web Semàntica. També he analitzat el contingut del paquet GTFS, descarregant els fitxers en fulles de càlcul. A partir d'aquí, pel disseny de l'ontologia, he adaptat el que proposen Noy i McGuiness en el seu article, treballant amb Protégé, ja que és l'eina idònia per fer-ho.
Per realitzar l'script, vaig decidir fer-ho amb Apache Jena, pensant que també podria fer les consultes SPARQL i, fins i tot, una petita aplicació per realitzar-les. Però vaig veure que no era una opció realista, per qüestió de temps, i vaig seguir el guió del treball de Fernando Rubí, en el sentit d'utilitzar Jena només per fer l'script i realitzar les consultes en Stardog. Això, finalment, ha estat molt interessant, ja que he treballat amb una aplicació comercial de BD RDF, cosa que no havia fet mai, i ha resultat molt enriquidor.

1.5. Planificació

El treball s'estructura temporalment d'acord amb les dates d'entrega de les PAC. Es divideix en blocs de dies per tal de poder gestionar el temps de manera dinàmica en funció de la disponibilitat (feina, família, imprevistos,...). Aquestes fases es corresponen amb el cicle de vida d'un projecte, d'acord amb la divisió del PMBOK.
La primera PAC presenta el plantejament del treball i la planificació. Podem considerar-la una fase prèvia on s'estructura el treball de manera teòrica. Seria la fase d'iniciació.
La segona PAC recull les fases d'anàlisi i requeriments, i modelització. En concret, els requeriments detallats i l'anàlisi de les fonts, així com la modelització de les dades, amb les dimensions, les taules de fets i el model de dades. Correspondria a les fases de definició i planificació.
La tercera PAC conté les fases de desenvolupament i producció, amb la instal·lació de les eines, la creació de l'ontologia, la població d'aquesta i les consultes SPARQL. Aquesta seria la fase d'execució.
La quarta PAC és el lliurament final, el qual presentarà el producte resultant i la memòria, amb les conclusions del treball. La memòria s'ha anat construint de manera transversal en cada PAC,

incorporant el contingut de cada fase. La redacció final integra totes les PACs en un sol document coherent. Seria la fase de seguiment i control del projecte. Inclou, així mateix, la presentació final del resultat del treball.

El resum de la planificació seria el que mostra la taula 1.

PAC	Inici	Entrega (fites)	Tasques
PAC1	20/09/2018	**03/10/2018**	Estudi del tema de la proposta i planificació del projecte.
PAC2	04/10/2018	**28/10/2018**	Estudi sobre la web semàntica, ontologies,... Estudi sobre el format GTFS. Esborrany de modelització.
PAC3	29/10/2018	**16/12/2018**	Creació de l'ontologia. Població de l'ontologia. Consultes sobre les dades.
PAC4	17/12/2018	**10/01/2019**	Redacció final de la memòria Preparació de la presentació

Taula 1: Planificació del treball

1.6. Estructura de la resta del document

Aquesta memòria s'estructura en tres parts diferenciades. Una primera part teòrica, on s'exposen els conceptes fonamentals relacionats amb la Web Semàntica, així com la situació actual de la mateixa. Una segona part en que es presenta la creació de l'ontologia i el seu poblament amb el contingut del paquet GTFS. I una tercera que mostra les dades mitjançant consultes SPARQL.

2. Estat de l'art

Per tal de contextualitzar el treball, cal presentar de manera esquemàtica la Web Semàntica i les tecnologies relacionades amb ella. La quantitat d'informació que hi ha disponible és molt gran, i no és l'objecte d'aquest treball realitzar un estudi exhaustiu de la seva situació actual, encara que no puc estar-me de comentar que, després de cercar en moltes pàgines, la sensació és que, en aquest moment, la Web Semàntica es troba en una situació "madura" en el sentit que fa temps que s'han establert una sèrie d'estàndards que s'estan demostrant molt útils per representar el coneixement en moltes àrees, com ara la de la salut, i que s'han desplegat en moltes eines per manipular les dades. Per exemple, Oracle Spatial and Graph integra RDF Knowledge Graph que ofereix gestió i anàlisi de dades avançat, suportant RDF, OWL i SPARQL.

La situació d'alguns dels grups de W3C relacionats amb aquests estàndards es consistent amb el que apuntem:

Grup de treball	Adreça	Situació segons l'enllaç
Semantic Web	https://www.w3.org/2001/sw/	*Page frozen on december 2013. Subsumed on W3 Data Activity.*
Semantic Web Interest Group	https://www.w3.org/2001/sw/interest/	*Closed on October 2018*
eGovernnment	https://www.w3.org/egov/	*Page frozen on december 2013. Subsumed on W3 Data Activity.*
RDF Working Group	https://www.w3.org/2011/rdf-wg/wiki/Main_Page	*End activities on July 2014*
OWL Working Group	https://www.w3.org/2007/OWL/wiki/OWL_Working_Group	*Closed on*
SPARQL Working Group	https://www.w3.org/2009/sparql/wiki/Main_Page	*Closed on*
Semantic Web Health Care and Life Sciences (HCLS) Interest Group	https://www.w3.org/2001/sw/hcls/	*Closed on February 2018*
Linked Data Platform Working Group	https://www.w3.org/2012/ldp/wiki/Main_Page	*Closed July 2015*

Taula 2: Situació actual d'alguns dels grups del W3C

Aquests grups han proporcionat els estàndards que ara s'estan utilitzant per construir la Web de Dades, objectiu del grup W3C Data Activity. La conclusió, agosarada, seria que s'han posat els fonaments, establint els estàndards i creant les eines de desenvolupament, per a tractar les dades d'una manera diferent, més adequada que la que proporcionen les bases de dades relacionals. Malgrat això, és possible que els recursos que s'han dedicat i es dediquen per etiquetar no siguin suficients, cosa que impedeix que les aplicacions puguin proporcionar els serveis de valor afegit que possibilitarien dedicar més recursos i crear, així, un cercle "virtuós". La realització del treball m'ha permès contrastar aquestes conclusions i copsar que, malgrat no és un concepte "popular" com, per exemple, el Big Data, la Web Semàntica està molt viva.

A continuació presento els conceptes bàsics per atacar la part pràctica d'aquest treball: la Web Semàntica i les seves parts, com són les ontologies i els estàndards en que es suporta, i el concepte d'Open Data.

2.1. Web semàntica

La idea que hi ha hagut des del principi darrera la web és compartir informació. En un primer moment, en la web "clàssica", es tractava de compartir documents, ja sigui documents en format HTML com continguts multimèdia o arxius de diferents tipus. Com que la informació es compartia entre humans eren aquests els que interpretaven la informació i la contextualitzaven. Aquesta manera de compartir la informació no és adequada per ser tractada de manera automàtica, especialment si es fa imprescindible fer-ho així a conseqüència del creixement exponencial del volum d'informació disponible que fa que, avui en dia, sigui inabastable per ser tractat per les persones. Així doncs, calia donar una solució a aquest problema. La resposta va ser la Web Semàntica, proposada per Tim Berners-Lee com una extensió de la clàssica "web de documents", una "web de dades" que permet als ordinadors manegar la informació que es troba a la xarxa sense la intervenció de persones. La solució tècnica va ser etiquetar els continguts i relacionar-los, de manera que la web es convertís en una immensa base de dades a la qual els ordinadors poguessin accedir per recollir les dades i tractar-les.

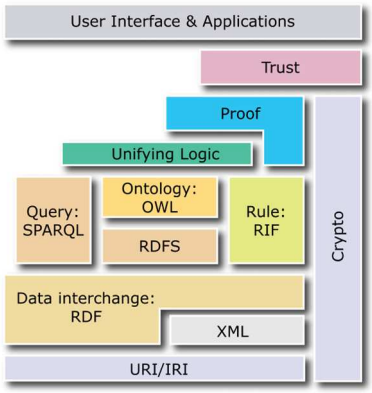


Figura 1: Les capes de la Web Semàntica

El W3C va proposar una estructura per la web semàntica basada en capes, que ha anat evolucionant en el temps, conforme s'han anat incorporant nous estàndards, però que bàsicament està constituïda per:

- Una capa lèxica, on es defineixen els símbols que s'utilitzen per construir-la.
- Una capa sintàctica, on es defineix la manera d'agrupar els símbols de la capa anterior.
- Una capa semàntica, el nucli de la Web Semàntica, que dota de significat els elements.
- Una capa de modelització, on es defineix el coneixement en relació als elements, és a dir, com es relacionen entre ells i quines propietats tenen.
- Una capa de raonament, que, mitjançant raonadors, relaciona el coneixement i permet fer-hi inferències, ja sigui per trobar incoherències com per crear nou coneixement.
- Una capa de confiança, que assegura la integritat de la informació, establint el grau de confiança d'una font o la confidencialitat de les dades.
- Una capa d'aplicacions que permet als usuaris humans accedir al coneixement o bescanviar informació de manera automàtica entre aplicacions.

En les capes que són la base de la web semàntica, les quatre primeres, W3C ha proposat una sèrie d'estàndards:

- En la capa lèxica, la inferior, proposa utilitzar URI i Unicode per localitzar i codificar els recursos.

- En la capa sintàctica, utilitza XML per tal de representar i estructurar la informació.
- En la capa semàntica, utilitza RDF per modelar la informació mitjançant triplets, grafs dirigits que contenen dos nodes (subjecte i objecte) i un vincle entre ells (predicat).
- En la capa de modelització, utilitza RDFS, una extensió de RDF, o OWL per representar els termes i les seves relacions, i SPARQL per realitzar consultes sobre els grafs.

2.2. Ontologia

Segons Tomas Gruber, podem definir una ontologia, en el context de les ciències d'informació, com un conjunt de conceptes bàsics representatius amb els quals es pot modelar un domini de coneixement o tema.

Per tal que una ontologia sigui útil, doncs, ha de tenir unes característiques clau: ha de ser explícita, compartida i limitada. Explícita perquè ha d'estar disponible en algun suport permanent per la seva consulta i modificació, compartida perquè ha de ser assumida per una comunitat de persones i limitada perquè ha de representar només una part de la realitat.

Els conceptes bàsics de les ontologies són les classes (o conjunts), els atributs (o propietats) i les relacions (o relacions entre membres de les classes).

Hi ha diferents maneres de classificar les antologies. Una d'elles, proposada per Nicola Guarino, les classifica segons el grau de generalitat. Així, les ontologies poden ser:

- Ontologies d'alt nivell (top-level), si descriuen conceptes generals que són independents d'un domini en particular. Aquest nivell no entraria en la definició d'ontologia proposada per Gruber.
- Ontologies de domini (domain), que descriurien un domini genèric.
- Ontologies de tasques (task), que descriuen el vocabulari especialitzat d'una tasca genèrica.
- Ontologies d'aplicació (application), que descriuen els conceptes d'una aplicació concreta.

En aquesta classificació, cada nivell és una especialització del superior. En el cas de l'ontologia proposada en aquest treball ens trobaríem amb una ontologia d'aplicació.

Els elements que formen una ontologia són els conceptes, les relacions, les instàncies i els atributs. Els conceptes representen les categories ontològiques que són rellevants en el domini de l'antologia. Les relacions connecten semànticament els conceptes. Les instàncies representen els objectes concrets del domini, que es classifiquen mitjançant les relacions amb els conceptes. I, finalment, els atributs representen les propietats i permeten definir les característiques dels elements que modelitzem. És important diferenciar les ontologies de les taxonomies. Mentre que aquestes últimes només relacionen pares i fills o classes i subclasses de manera jeràrquica, les ontologies permeten modelitzar relacions molt més complexes.

2.3. Estàndards

El World Wide Web Consortium (W3C) és un consorci internacional que treballa per a desenvolupar i promocionar estàndards pel World Wide Web. Va ser fundat l'octubre de 1994 i està presidit i dirigit pel

seu fundador, Tim Berners-Lee. La seu central es troba al Massachusetts Institut of Technology, a Cambridge, USA.

En la seva presentació podem llegir que la seva visió del futur és una web que involucra la participació, compartint el coneixement, i així construir confiança a escala global. Una part d'això vindrà donat per un web de dades i serveis, un gran repositori de dades enllaçades amb un gran conjunt de serveis que bescanviaran missatges.

Una de les tasques fonamentals d'aquest consorci és establir els estàndards sobre els que es construeix aquesta Web, el qual inclou la Web Semàntica.

2.3.1. URI

L'Uniform Resource Identifier és un conjunt de caràcters que identifica de forma inequívoca un recurs particular. La Web Semàntica utilitza l'esquema HTTP URI per identificar documents i conceptes en el món real. La sintaxi genèrica està establerta en el document RFC3986 del Network Working Group, de gener de 2005, la qual consta d'una seqüència jeràrquica de cinc components.

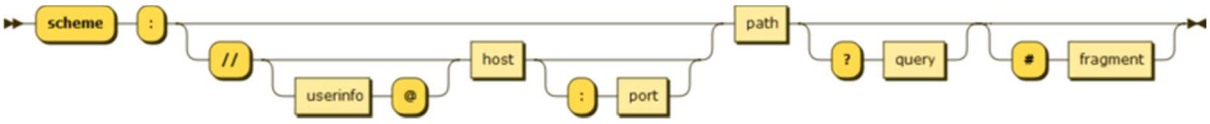


Figura 2: Diagrama d'URI

2.3.2. XML

eXtensible Markup Language (llenguatge de marques extensible) és un llenguatge d'etiquetes que defineix un conjunt de regles per codificar documents en un format que pot ser llegit tant per màquines com per humans. L'especificació XML 1.1 del W3C, i d'altres relacionades, totes elles estàndards oberts, defineixen XML.

Un document XML és un arxiu de text codificat amb Unicode. Ha d'estar format únicament per etiquetes (caràcters entre < i >) i caràcters entre etiquetes. El document té una capçalera, amb metadades que descriuen el document, i el cos on es troba la informació dins d'una estructura jeràrquica que delimiten les etiquetes d'inici i de final.

2.3.3. RDF

Resource Description Framework és un model conceptual per definir dades en la Web Semàntica i, en general, per a representar la informació. És un estàndard del W3C, i en aquest moment l'especificació és la 1.1.

El model RDF es basa en enunciats, també anomenats triplets, que relacionen entitats de manera binària. Així, un triplet està format per un subjecte, l'entitat origen de la relació, un objecte, l'entitat destí de la relació, i un predicat, o propietat, que relaciona el subjecte i l'objecte. Així, doncs, un triplet és un graf dirigit.

RDF-S o RDF Schema és una extensió de RDF que proporciona els elements per descriure ontologies: classes, jerarquies i propietats.

2.3.4. OWL

Ontology Web Language és un llenguatge per a la definició d'ontologies estructurades basades en la web. Amplia RDF Schema per a poder expressar relacions complexes entre diferents classes i proporciona eines per establir restriccions i propietats específiques.
Actualment hi ha dues versions d'OWL: OWL1 i OWL2. La segona és una revisió i extensió d'OWL1, de manera que les ontologies OWL1 són vàlides en OWL2.
OWL és un llenguatge basat en lògica descriptiva i es presenta en tres nivells de complexitat creixent: OWL Lite, OWL DL i OWL Full.

- OWL Lite és la versió més senzilla i ofereix només la definició d'una jerarquia i restriccions simples de cardinalitat.
- OWL DL amplia la versió anterior, i garanteix la completesa i decibilitat computacional amb la màxima expressivitat possible.
- OWL Full amplia OWL DL, però no proporciona cap garantia computacional. És a dir, que poden haver processos d'inferència que no finalitzin mai, conseqüència de relaxar la separació entre els conceptes instància, classe i propietat.

2.3.5. SPARQL

SPARQL Query Language for RDF és un llenguatge utilitzat per realitzar cerques sobre dades en format RDF. És un estàndard de W3C i actualment l'especificació és la 1.1.
SPARQL permet fer cerques complexes, però la seva estructura bàsica és simple i es basa en dues clàusules: SELECT i WHERE. Les consultes es realitzen en un processador de consultes SPARQL, com ara el Wikidata Query.

2.3.6. GeoSPARQL

És un estàndard per representar i consultar dades geoespacials en la Web Semàntica definit per l'Open Geospatial Consortium. GeoSPARQL defineix un vocabulari per representar dades en format RDF i una extensió del llenguatge SPARQL per processar dades geoespacials. Així, GeoSPARQL proporciona una petita ontologia topològica en RDFS/OWL, basada en estàndards de l'OGC, per representació i una interfície de consultes SPARQL, amb una sèrie de funcions topològiques i un conjunt de regles d'inferència.
Hi ha diferents implementacions de GeoSPARQL, entre elles Stardog.

2.4. Open Data

La idea que hi ha darrera d'Open Data, com darrera d'altres moviments "Open", és compartir. Les dades haurien d'estar disponibles lliurament per tal que tothom que ho desitgi pugui utilitzar-les sense restriccions en forma de copyrights, patents o d'altres mecanismes de control. La filosofia darrera

d'aquest moviment no és nova i entronca amb la tradició científica de compartir el coneixement per tal de fer avançar la humanitat. Amb l'arribada de W3 i la possibilitat de bescanviar qualsevol informació fàcilment, aquesta va escapar al control dels seus creadors. Aleshores es va veure la necessitat de crear mecanismes per tal que aquests veiessin reconegudes les seves creacions. Les solucions ofertes primerament van incrementar els drets de propietat intel·lectual i, com a conseqüència, la burocràcia, la judicialització,…

Una solució alternativa per resoldre el problema de reconeixement, va ser identificar totes aquelles creacions que es poden distribuir lliurament mitjançant el concepte d'Open Data, que es pot resumir de la següent manera:

Obert vol dir que qualsevol pot lliurement accedir, utilitzar, modificar i compartir per qualsevol propòsit. Per tant, les dades obertes poden ser lliurament utilitzades, modificades i compartides per qualsevol per qualsevol propòsit.

Així, per tal d'identificar les creacions o dades que estan disponibles seguint els principis anteriors, s'han establert una sèrie de llicències, com ara les que proporciona Creative Commons.

Aquest concepte va guanyar molta popularitat quan els governs van veure la possibilitat de posar moltes de les dades de que disposen a disposició del públic mitjançant W3, seguint polítiques de transparència. Es van obrir molts portals en quasi totes les administracions públiques, però moltes vegades les dades estan obsoletes, no són complertes o exigeixen identificació per disposar d'elles, cosa que viola el principi "open" i fa pensar més en una intenció propagandística més que en un servei públic. Malgrat això, cal dir que algunes administracions realment creuen en el concepte i posen a l'abast de tothom dades actualitzades i aplicacions adequades per tal de consultar les seves dades. Un exemple és la Unió Europea, que disposa d'un portal amb infinitat de dades en múltiples formats estàndard i les eines necessàries per consultar i tractar aquestes dades.

Per tal d'estendre la difusió de les dades obertes en internet, en 2012, Tim Berners-Lee i Nigel Sahdbolt van crear l'Open Data Institut, amb seu a Londres. L'ODI s'articula amb nodes distribuïts per tot el món. En l'actualitat hi ha 28 nodes, un d'ells a Barcelona.

Tim Berners-Lee també ha proposat el sistema Cinc Estrelles, disponible com a llicència, per qualificar la qualitat de les dades obertes que s'ofereixen. Aquest sistema estableix cinc passes de desenvolupament que van des de publicar dades sota una llicència oberta (en qualsevol format) a publicar dades enllaçades per tal de proveir context. Les dades que proporciona GTFS estarien situades en el tercer nivell, ja que utilitza formats no propietaris.

2.4.1. GTFS

GTFS (General Transit Feed Specification) defineix un format obert estàndard per bescanviar dades sobre l'estructura, l'horari, la informació geogràfica i la tarifa d'un sistema de transport públic. El format GTFS permet a les agències de transport públic lliurar les seves dades en un format comú que es pot consumir i reutilitzar en aplicacions independents de les esmentades agències.

Més d'un miler de companyies proporcionen dades en el format GTFS i centenars d'aplicacions les utilitzen per presentar-les de diferents formes als usuaris finals.

El format GTFS es divideix en dos: el propi GTFS (també anomenat GTFS estàtic) que proporciona dades dels serveis programats, i el GTFS Realtime, que s'utilitza per subministrar dades en temps real, com ara el temps estimat d'arribada, les posicions dels vehicles o els avisos del servei.

Cal esmentar, però, que GTFS és només un dels models de dades sobre transport públic que existeixen. Per exemple, el model de referència europeu és Transmodel, que, a diferència de GTFS, està basat en UML.

Un paquet GTFS està format per una col·lecció d'entre sis i tretze fitxers CVS, amb extensió TXT, empaquetats en un fitxer ZIP. Tenen una estructura de BD relacional, amb identificadors que relacionen entre ells el contingut dels fitxers.

Fitxer	Requerit	Contingut
agency.txt	**Sí**	Les agències de transport que proveeixen les dades d'aquest paquet
stops.txt	**Sí**	Localitzacions individuals on els vehicles agafen i deixen passatgers.
routes.txt	**Sí**	Rutes de trànsit. Una ruta és un grup viatges que es realitzen en un servei..
trips.txt	**Sí**	Viatges de cada ruta. Un viatge és una seqüència de dos o més aturades que s'efectuen en un temps específicat..
stop_times.txt	**Sí**	Temps en el que un vehicle arriba i surt de les parades individuals de cada viatge..
calendar.txt	**Sí**	Dates per servei utilitzant un calendari setmanal. Especifica quan comença i acaba un servei, així com els dies de la setmana en que el servei està disponible.
calendar_dates.txt	Opcional	Excepcions al servei en relació a les definides en el fitxer **calendar.txt**. Si calendar.txt inclou TOTES les dates del servei, aquest fitxer pots ser especificat en lloc de calendar.txt.
fare_attributes.txt	Opcional	Informació de les tarifes de les rutes de l'agència de transport.
fare_rules.txt	Opcional	Normes per aplicar les tarifes de les rutes de l'agència de transport.
shapes.txt	Opcional	Normes per dibuixar línies en un mapa per representar les rutes d'una agència de transport.
frequencies.txt	Opcional	Intervals (temps entre viatges) per rutes amb freqüències de servei variables.
transfers.txt	Opcional	Normes per fer connexions en punts de transbord entre rutes..
feed_info.txt	Opcional	Informació addicional sobre el paquet mateix, com ara l'editor, la versió i quan expira la informació.

Taula 3: Fitxers del paquet GTFS

Tots els fitxers han de complir una sèrie de requeriments:

- Els camps han d'estar separats per comes.
- La primera línia ha de contenir els noms dels camps.
- Els noms dels camps són "case-sensitive".
- Els camps no poden contenir tabuladors, retorns de línia o noves línies.
- Els valors dels camps que tinguin cometes han d'anar entre cometes, seguint els criteris del format CSV.

- Els camps no han de contenir etiquetes HTML, comentaris o seqüències d'escapada.
- Cal remoure els espais entre els camps o entre els noms dels camps.
- Cada línia ha d'acabar amb un caràcter CRLF o LF.
- Els fitxers han d'anar codificats en UTF-8 per suportar tots els caràcters Unicode.
- El paquet ha d'estar empaquetat en un fitxer ZIP.

D'altra banda, cadascun d'aquest fitxers té definits una sèrie de camps, requerits o opcionals, amb característiques determinades. A l'annex C es pot trobar la descripció, en anglès, de les actuals especificacions dels camps dels fitxers que hem utilitzat en aquest treball.

2.4.2. GTFS del Consorcio Regional de Transportes de Madrid

Seguint la filosofia de dades obertes, el Consorcio Regional de Transportes de Madrid posa a disposició del públic un Portal de Dades Obertes "amb el propòsit de difondre la informació del transport públic de la Comunitat de Madrid, per tal que empreses, organitzacions no lucratives, i tots els ciutadans puguin trobar i reutilitzar aquesta informació...".
Les dades estan disponibles en diversos formats, entre ells GTFS, agrupades per xarxes:

Xarxa	Data creació	Data actualització	Nom fitxer
Metro	05/04/2016	12/02/2018	Google_transit_M4.zip
Autobuses Urbanos de Madrid (EMT)	05/04/2016	12/02/2018	Google_transit_M6.zip
Autobuses Urbanos de la Comunidad de Madrid	05/04/2016	12/02/2018	Google_transit_M9.zip
Autobuses Interurbanos de la Comunidad de Madrid	05/04/2016	12/02/2018	Google_transit_M89.zip
Metro Ligero/Tranvía	05/04/2016	12/02/2018	Google_transit_M10.zip
Cercanías	05/04/2016	12/02/2018	Google_transit_M5.zip

Taula 4: Paquets GTFS proporcionats per CRTM

Tots els paquets contenen els mateixos arxius en format CSV: agency.txt, calendar.txt, calendar_dates.txt, fare_attributes.txt, fare_rules.txt, feed_info.txt, frequencies.txt, routes.txt, shapes.txt, stop_times.txt, stops.txt i trips.txt.
En el cas del paquet **Google_transit_M4.zip**, que és el que hem utilitzat pel treball, els arxius fare_attributes.txt i fare_rules.txt no contenen dades.

3. Proposta

Una vegada realitzada la recerca, limitada pel temps disponible, de les diferents eines que hi ha a l'abast en aquest moment, i consultats altres TFG d'anys anterior, he triat Protégé, Apache Jena i Stardog per implementar l'ontologia, seguint els passos del TFG de Fernando Rubí. Per fer la tria, he tingut en compte que aquestes aplicacions tenen versions actualitzades, ja que moltes de les eines consultades o bé ja no existeixen o no s'actualitzen des de fa anys. Així mateix, també he valorat que les dues primeres són eines obertes de les quals existeix molta informació. Finalment, Stardog, malgrat ser propietària, és una aplicació totalment actualitzada que ofereix versions de prova renovables i dona suport a GeoSPARQL, que en aquesta ontologia proporciona un gran valor afegit.

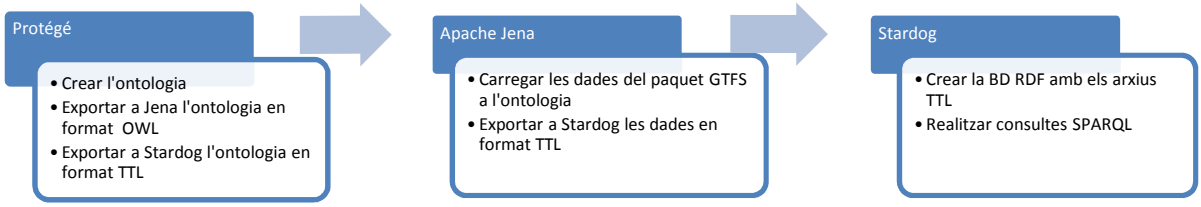


Figura 3: Flux de treball de les aplicacions del projecte

3.1. Protégé

Protégé, en la seva versió 5.2.0, s'ha utilitzat per la construcció i prova de l'ontologia. És una plataforma lliure de codi obert que proporciona eines per construir ontologies de models de domini i coneixement. Està mantingut per l'Stanford Center for Biomedical Informatics Research. Desgraciadament, el gran volum de dades ha impossibilitat fer ús de moltes de les seves API, ja que el programari deixava de funcionar quan havia de tractar molts individus. Això ha dificultat fer regles i provar a fons els raonadors per inferir noves classes o propietats.

3.2. Apache Jena

Apache Jena és una plataforma de codi obert en Java pel Web Semàntic que proporciona una API per extreure i escriure dades des de RDF. També dona suport per OWL. Disposa de diferents raonadors i una interfície (Fuseki) en HTML que permet actualitzar RDF i realitzar consultes SPARQL S'ha utilitzar la versió 3.9.0 per crear l'script amb el que he poblat l'ontologia, creada amb Protégé, amb les dades del paquet GTFS.

3.3. Stardog

Stardog és una BD RDF comercial que, segons W3C, és molt ràpida realitzant consultes SPARQL, transaccions i raonadors sobre OWL. He instal·lat la versió 6.0.1 i Stardog Studio, una nova interfície que facilita la càrrega de les dades en la BD, la gestió de la pròpia BD (activar/desactivar opcions i la

pròpia BD, treballar SPARQL,...). També dóna suport a consultes GeoSPARQL, importants en el context d'aquest treball.

4. Disseny

4.1. Metodologia

Pel disseny de l'ontologia he adaptat les regles que enumeren Noy i McGuiness en el seu article, així com els passos bàsics que elles ens proposen, tenint sempre present que no hi ha cap metodologia que podem considerar perfecte o la millor.

Les regles són tres:

- No hi ha una manera correcta de modelar un domini, dependrà del propi domini i de l'ús que es vulgui donar.
- El procés de construcció és iteratiu.
- Els conceptes han de ser propers als objectes i les relacions en el domini.

Els passos bàsics són els següents:

- Determinar el domini i l'abast de l'ontologia.
- Considerar la reutilització d'ontologies existents
- Enumerar els termes rellevants de l'ontologia
- Definir les classes i la jerarquia
- Definir les propietats de les classes
- Definir les restriccions de les propietats
- Crear instàncies

En el nostre cas, cal tenir present en tot moment que l'objectiu final és instanciar les dades que hi ha en el paquet GTFS en l'ontologia creada. Per això, s'han adaptat els passos de la manera següent:

- El domini i l'abast de l'ontologia ve condicionat pel contingut del paquet GTFS.
- Considerar la reutilització d'ontologies existents sobre el domini del transport públic.
- Enumerar els termes rellevants.
- Definir les classes i la jerarquia.
- Definir les propietats de les classes.
- Definir les restriccions de les propietats, d'acord amb les definicions de GTFS.
- Poblar la ontologia amb el contingut del paquet GTFS.

4.2. Domini i abast de l'ontologia

Tal com s'ha explicat, l'ontologia que s'ha creat està centrada en una xarxa de transport públic, en concret la de la regió de Madrid, amb les dades subministrades pel CRTM. La xarxa es composa de diverses subxarxes, cadascuna de les qual té un paquet GTFS individualitzat. Per tal de desenvolupar l'ontologia he utilitzat la del metro, encara que tot el procés es pot fer servir per qualsevol dels altres paquets o per tots alhora. L'ontologia es pot utilitzar per determinar la ubicació de les parades, el recorregut de les línies, el temps de pas i les parades més properes a llocs d'interès. En última instància, hauria de permetre, per exemple, planificar rutes a través de la ciutat.

4.3. Reutilització d'ontologies existents

Hi ha força documentació sobre ontologies en el camp del transport públic, les quals coincideixen, lògicament, en plantejar una sèrie de classes: agències (operadors), línies, parades, calendaris, temps de pas, vehicles,… Dues de les més interessants són la de Katsumi i Fox i la de Mnasser et al. La primera fa una descripció teòrica sobre el que hauria de tenir qualsevol ontologia de transport i la segona, més pràctica, centra el focus en una ontologia que proporcioni suport a l'usuari quan aquest planifica un viatge.

Ara bé, donat que l'objectiu del treball era fer una ontologia sobre el contingut dels paquets GTFS, en lloc d'adaptar les propostes, he optat per crear l'ontologia a partir dels conceptes que proporciona el paquet GTFS. El resultat és una ontologia menys complexa, però amb individus en totes les classes.

4.4. Enumerar els termes importants en l'ontologia

Tal com s'ha explicat anteriorment, els paquets GTFS consten d'una sèrie de fitxers plans txt, amb els camps separats per comes. Tots plegats conformen una mena de base de dades relacional, amb identificadors per poder relacionar els diferents fitxers.

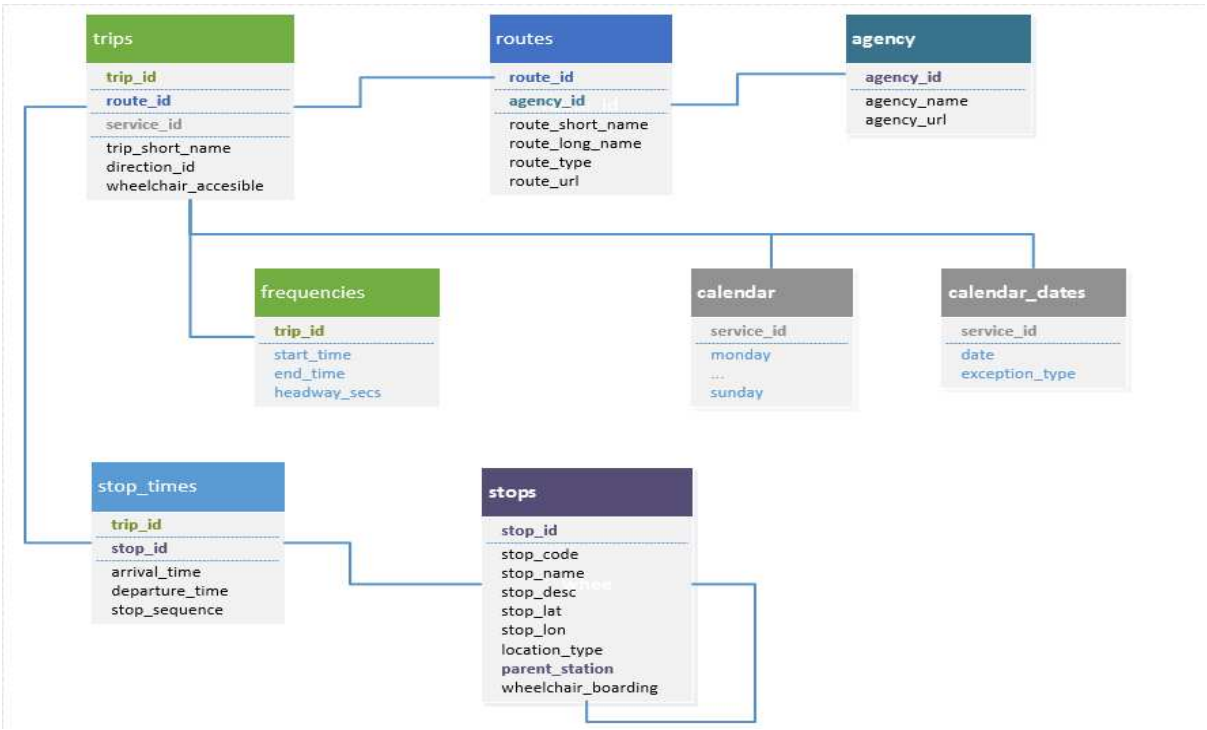


Figura 4: Estructura del paquet GTFS i relacions entre els fitxers

De tots els fitxers, treballarem amb les dades contingudes en els següents: agency, routes, stop_times, stops i trips. De la resta, hi ha dos que no tenen dades (els corresponents a les tarifes, fare_attributes i fare_rules), un que fa referència al propi paquet (feed_info), un altre que conté els recorreguts de les línies amb coordenades geogràfiques i distàncies i, finalment, els dos fitxers de calendaris (calendar i

calendar_dates) i el de freqüències. Aquests últims no ha estat possible incorporar-los a l'ontologia per qüestions de temps.
Així, doncs, els termes de que tractarà aquesta ontologia serà sobre les línies de transport públic que gestiona una agència, les seves parades, els seus viatges i els temps de pas (successos) que realitzen per les parades els vehicles.
Amb això respondrem a preguntes com ara:

- On es troba una parada?
- Quines parades té una línia de transport?
- Quina parada puc utilitzar per canviar entre línies?
- Quina parada està a prop d'un lloc concret?

4.5. Definir les classes i la jerarquia

En aquest treball, per tal d'establir la jerarquia m'he basat en els conceptes fonamentals que he tractat abans, de manera que el procés ha estat relacionar aquests conceptes i realitzar alguna especialització allà on ha calgut. Un dels punts importants a remarcar és que he fet un ús "exhaustiu" del procés d'iteració, procés que, com comentaré després, es pot continuar fent per tal d'afinar l'ontologia i incloure una sèrie de classes noves, com ara els calendaris, els recorreguts i les freqüències..

Així, doncs, la classe Agencies representa les operadores de les línies. En aquest cas, només hi haurà una, la CRTM.
La classe Linies representa les línies (routes) de transport públic, amb subclasses segons el tipus de vehicle que utilitzen per donar el servei.
La classe Parades representa els llocs (stops) on les línies s'aturen per embarcar o desembarcar passatgers.
La classe Viatges representa els diferents viatges (trips) que realitzen les línies, complint uns horaris concrets.
La classe Successos representa les aturades que fan els vehicles en les parades durant un viatge.
La classe Vehicles representa els vehicles que realitzen els viatges i té com a subclasses cada tipus de vehicles, a partir dels quals es pot inferir els tipus de Linies (LiniaMetro, LiniaBus,...) en funció del vehicle que utilitzen. En els paquets GTFS no hi ha informació específica sobre els vehicles.
La classe Geometry està manllevada de les especificacions de l'OGC i serveix per utilitzar les funcions geogràfiques que permetran fer consultes GeoSPARQL. Conté la subclasse Point, que està definida a http://www.opengis.net/ont/sf#Point. Aquesta classe crea una instància de la classe Parades amb el punt geogràfic on està ubicada.

4.6. Definir les propietats de les classes

En aquest punt establirem les propietats de les classes i entre les classes. El final del procés iteratiu que he seguit està resumit en la següent taula. Per anomenar les propietats d'objecte he establert la convenció d'anomenar-les com a "te"+Classe i la seva inversa com a "es"+Classe+"De", tret de

hasGeometry que és una propietat de GeoSPARQL i esSuccesDe i teSucces que, encara que segueixen la convenció, no són classes inverses perquè apliquen de manera diferent. Els noms de les propietats tenen la convenció lowerCamelCase.

Les propietats entre classes seran les següents:

Propietat	Domini	Rang	Inversa
esAccesDe	Acces	Estacio	teAcces
esAgenciaDe	Agencies	Linies	teAgencia
esEstacioDe	Estacio	Acces or Parada	teEstacio
esParadaDe	Parades	Viatges	teParada
esSuccesDe	Successos	teParada exactly 1 Parades and teViatge exactly 1 Viatges	
teSucces	Parades or Viatges	Successos	
esVehicleDe	Vehicles	Linies	teVehicle
esViatgeDe	Viatges	Linies	teViatge
hasGeometry	Parades	Geometry	

Taula 5: Propietats entre classes

I les propietats de les classes seran les següents:

Propietat	Domini	Rang
asWKT	Geometry	wktLiteral
codiParada	Acces or Estacio or Parada	xsd:string
esAccessible		xsd:string
geo:lat	Point	xsd:float
geo:long	Point	xsd:float
horaArribada	Successos	xsd:dateTime
horaSortida	Successos	xsd:dateTime
nom		xsd:string
ordreParada	Successos	xsd:int
ubicacio		xsd:string
URL		xsd:anyURI

Taula 6: Propietats de les classes

En les darreres iteracions vaig provar d'establir regles per tal d'aplicar el raonador. Així, per exemple, vaig crear la regla següent en SWRL:

:esParadaDe(?p, ?v) ^ :esViatgeDe(?v, ?l) -> :esParadaDe(?p, ?l)

que es tradueix per: si p és parada del viatge v i v és un viatge de la línia l, aleshores p és parada de la línia l. Aquesta regla ens proporciona directament les parades d'una línia. Implementar la regla implica modificar el rang de la propietat esParadaDe, i el domini de la propietat teParada, a Linies or Parades, ja que, en cas contrari, el raonador dedueix que Viatges i Parades són el mateix. Per problemes de memòria en Protégé, he prioritzar l'entrega del treball i, finalment, no he implementat cap regla en l'ontologia final.

4.7. Definir les restriccions de les propietats

A continuació, cal establir les característiques de les propietats en relació a les classes.

Classe	Subclasse de	Disjunta amb
Agencies	esAgenciaDe some Linies	
Geometry		
Point	Geometry	
Linies	teAgencia exactly 1 Agencies teViatge some Viatges	
LiniaMetro	Linies teVehicle exactly 1 Metro	LiniaTren,LiniaTram,LiniaBus
Parades	es ParadaDe some Viatges hasGeometry exactly 1 Point teSuccess some Successos	
Acces	Parades	Parada, Estacio
Estacio	Parades	Acces, Parada
Parada	Parades	Acces, Estacio
Successos		
Vehicles	esVehicleDe some Linies	
Metro	Vehicles	Tren,Bus,Tram,...
Viatges	esViatgeDe exactly 1 Linies teParada some Parades teSucces some Successos	
ViatgeAnada	Viatges	ViatgeTornada
ViatgeTornada	Viatges	ViatgeAnada

Taula 7: Característiques de les propietats de les classes

Els altres tipus de Linies tenen les mateixes característiques que LiniaMetro, però amb el vehicle corresponent. Així mateix, la resta de Vehicles tenen les mateixes característiques que Metro.
A l'hora d'especificar els Viatges ha quedat pendent de definir la classe ViatgeCircumvalacio, que GTFS no especifica, ja que el codifica com un ViatgeAnada. Per solucionar-ho es pot establir una regla, com per exemple que l'inici i el final és la mateixa Parada o que la Línia no te cap ViatgeTornada.

4.8. Crear instàncies

Una vegada establertes classes i propietats, cal poblar l'ontologia amb el contingut dels paquets GTFS. Encara que el procés de càrrega està pensat per carregar tots els paquets de les diferents xarxes, finalment només he treballat amb el paquet corresponent al metro, ja que era molt dificultós realitzar les proves amb el volum que suposava la càrrega de les dades de tots els paquets.

5. Implementació

5.1. Instal·lació de Protégé

Per realitzar la instal·lació, només cal descarregar l'aplicació, descomprimir-la i executar el run.bat (si es treballa en Windows) que es troba a l'arrel.

5.2. Instal·lació d'Apache Jena a l'Eclipse

Una vegada baixat i descomprimits els paquets, per poder utilitzar Apache Jena, com en tot projecte Java, cal afegir les llibreries al Java Build Path del projecte. Per tal d'evitar un avís que apareix a l'executar les aplicacions, cal afegir el fitxer log4j.properties a la carpeta bin del projecte. Aquest fitxer es troba a la carpeta arrel apache-jena-3.9.0 com a jena-log4j.properties. Cal canviar-li el nom quan es copia a la nova ubicació.

Així mateix, per tal de llegir els fitxers txt, també cal descarregar la llibreria javacsv-2.0.jar i incorporar-la al Java Build Path del projecte.

Per tal de tenir les versions controlades i facilitar el direccionament, he creat unes carpetes on deixar els fitxers necessaris per executar l'aplicació. En la carpeta GTFS estan els paquets de les dades, en la carpeta OWL està l'ontologia desenvolupada amb Protégé i en la carpeta TTL estan l'ontologia de Protégé en format TTL i el fitxer amb les dades, també en format TTL, després del procés de poblament, i que estan destinats a la càrrega en Stardog Studio.

En aquest treball s'ha utilitza Eclipse Luna i Java 1.8.

5.3. Instal·lació d'Stardog

Per utilitzar Stardog, he sol·licitat una versió d'avaluació. La versió que he utilitzat és la 6.0.1 que presenta Stardog Studio com a interfície de treball en lloc de l'antiga consola, cosa que facilita la gestió de l'aplicació. Per posar en marxa el servidor només cal configurar el SET PATH (en Windows) per tal que apunti al directori bin d'Stardog i executar la comanda *stardog-admin server start* en un terminal.

5.4. Creació de l'ontologia amb Protégé

Una vegada instal·lades les eines, he utilitzat Protégé per crear les classes i les propietats descrites anteriorment. La jerarquia de classes és la següent, presentada a partir d'una imatge de l'Ontograf de Protègè.

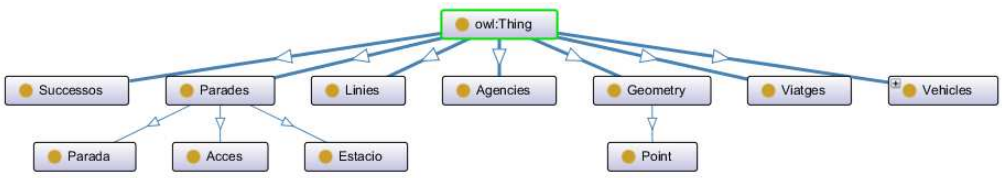


Figura 5: Classes de l'ontologia

Una vegada afegides les propietats que relacionen les classes, el gràfic que resulta és el següent:

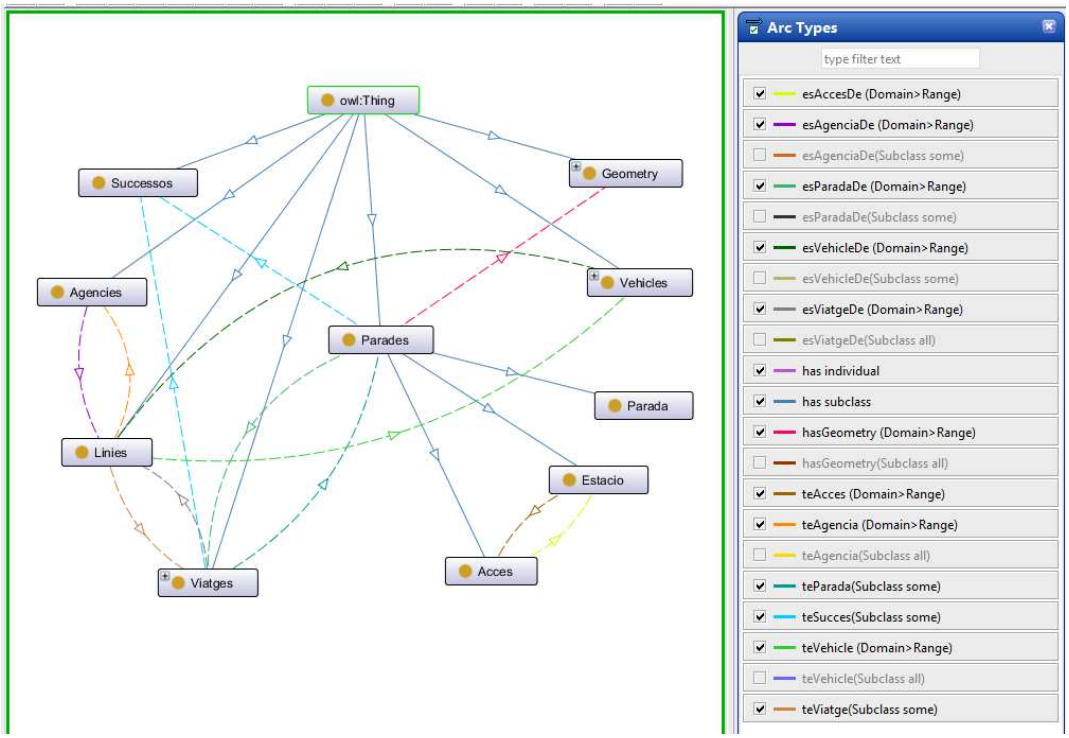


Figura 6: Propietats entre les classes de l'ontologia

La relació de les propietats que he determinat queden així en Protégé:

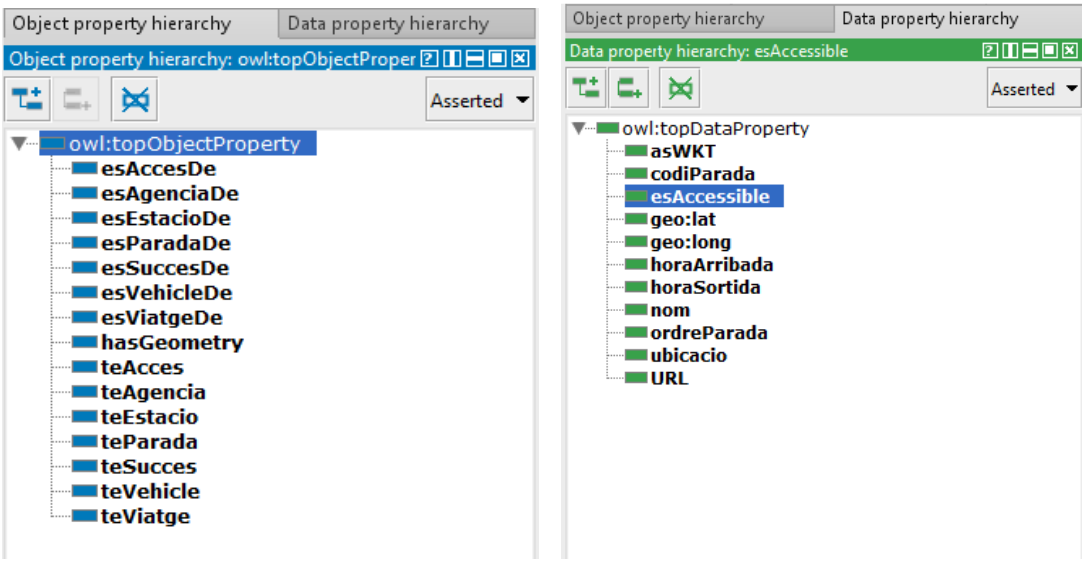


Figura 7: Les propietats en Protégé

5.5. Població de l'ontologia amb Apache Jena

Per tal de poblar l'ontologia, he creat una classe java amb una sèrie de mòduls que carreguen de manera seqüencial el contingut dels fitxers CSV, assegurant la coherència de les dades.

Principal
+ lecturaFitxer(String fitxerCsv) + carregaAgencies(String fitxerCsv, OntModel ontologia, String ns) : OntModel + carregaLinies(String fitxerCsv, OntModel ontologia, String ns) : OntModel + carregaParades(String fitxerCsv, OntModel ontologia, String ns) : OntModel + carregaViatges(String fitxerCsv, OntModel ontologia, String ns) : OntModel + carregaViatgeParada(String fitxerCsv, OntModel ontologia, String ns) : OntModel

L'aplicació va cridant cada mòdul, passant-li l'ontologia i el fitxer csv a carregar. En cada pas es creen els individus i es relacionen amb les propietats adequades. L'esquema de la càrrega seqüencial seria el següent:

Mòdul	Individus	Propietats de classe	Propietats entre classes
carregaAgencies	Agencies	nom, URL	
carregaLinies	Linies	nom, URL	teAgencia, teVehicle, esAgenciaDe, esVehicleDe
carregaParades	Parada	nom, codiParada,esAccessible, geo:lat, geo:long	hasGeometry
	Acces	nom, codiParada,ubicacio, esAccessible, geo:lat, geo:long	teEstacio, hasGeometry
	Estacio	nom, codiParada,ubicacio, esAccessible, geo:lat, geo:long	esEstacioDe, hasGeometry
	Point	POINT(long,lat)asWKT	
carregaViatges	ViatgeAnada	nom, esAccessible	esViatgeDe, teViatge
	ViatgeTornada	nom, esAccessible	esViatgeDe, teViatge
carregaViatgeParada	Successos	horaArribada, horaSortida, ordreParada	teParada, esParadaDe, teSuccess, esSuccesDe

Taula 8: Mòduls de càrrega de la classe Principal

Al final del procés, es lliura un fitxer en format Turtle. En les primeres proves, amb poques classes, el fitxer final era en format RDF/XML per comprovar en Protégé la càrrega dels individus i les propietats i les coherències entre ells, però en versions posteriors, amb gran volum de dades, Protégé deixava de respondre per problemes de memòria.

Per executar l'script de càrrega, una vegada importat el projecte en l'Eclipse com "Archive File", només cal executar la classe "Principal.java", que es troba a src/GTFS com a "Aplicació Java". A la carpeta TTL es desa l'arxiu Turtle generat.

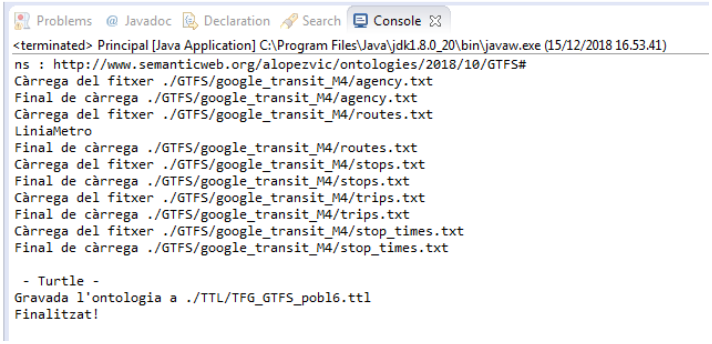


Figura 8: Resultat de l'execució de l'aplicació

5.6. Creació de la BD a l'Stardog

Una vegada que el servidor està en marxa podem crear la base de dades a partir de l'arxiu TTL creat amb Protégé, que conté l'ontologia. La comanda és *stardog-admin db create*, tal com s'explica en el manual d'Stardog. Amb la BD creada, l'Studio ens facilita la tasca de parametritzar-la per tal d'habilitar les funcions espacials i carregar les dades.

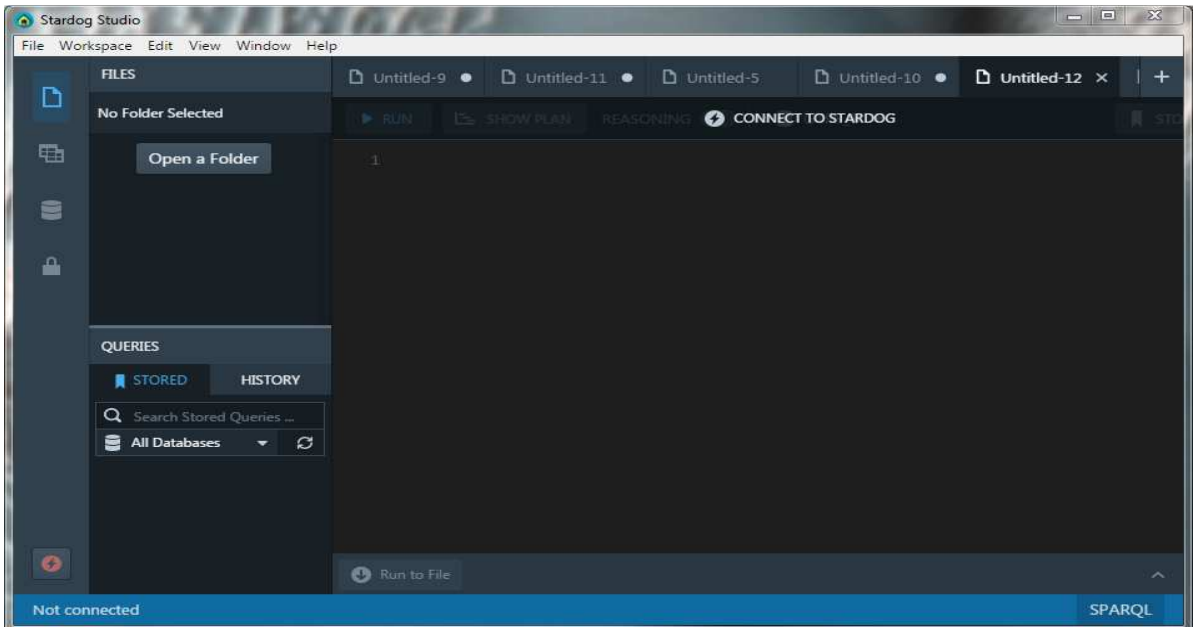


Figura 9: Interfície d'Stardog Studio

Quan arranquem l'Studio, el primer que cal fer, si no hem parametritzat la connexió automàtica, és connectar-se a Stardog.

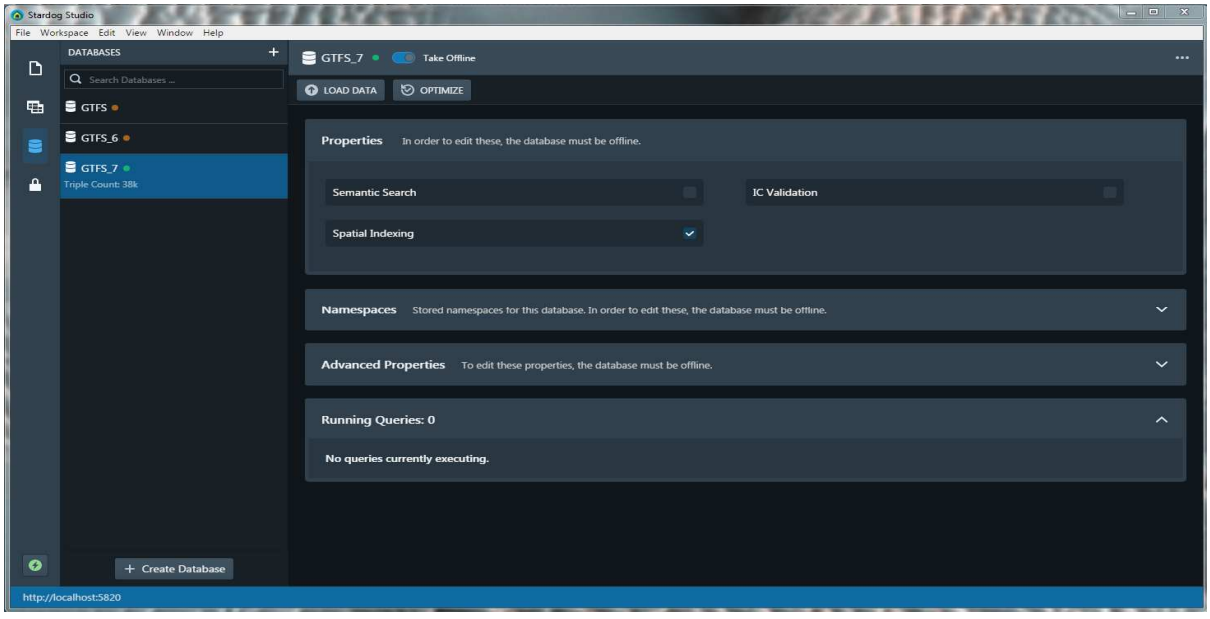


Figura 10: Parametrització de la BD en Stardog Studio

Una vegada connectats, la interfície d'Studio permet veure les BD, carregar les dades, activar o desactivar paràmetres i d'altres tasques. En el nostre cas, la utilitzarem per realitzar la càrrega de les dades a partir de l'arxiu Turtle que hem generat amb Jena i realitzar les consultes sobre la nostra BD.

6. Demostració

6.1. Consultes amb Stardog Studio

6.1.1. Parades d'una línia

La primera consulta és sobre les parades d'una línia, amb el nom de la línia, el de la parada, el codi de la parada i l'accessibilitat.

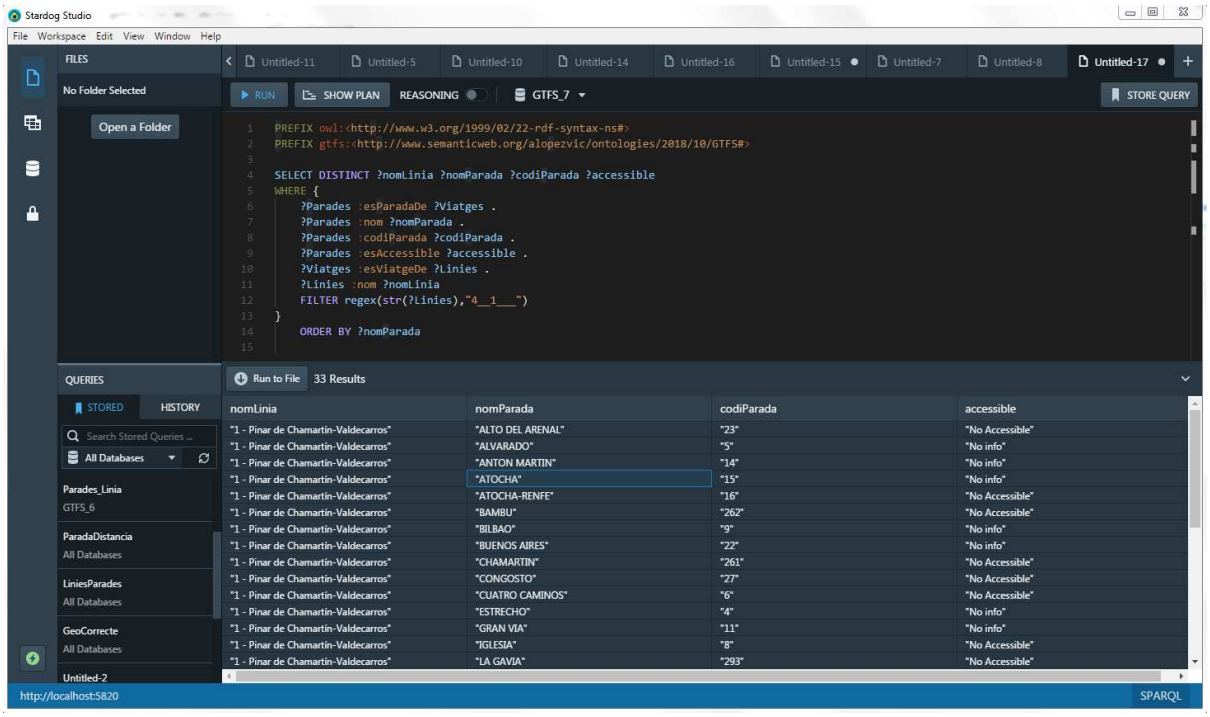


Figura 11: Consulta SPARQL sobre parades d'una línia

Consulta 1

```
PREFIX owl:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX gtfs:<http://www.semanticweb.org/alopezvic/ontologies/2018/10/GTFS#>

SELECT DISTINCT ?nomLinia ?nomParada ?codiParada ?accessible
WHERE {
    ?Parades :esParadaDe ?Viatges .
    ?Parades :nom ?nomParada .
    ?Parades :codiParada ?codiParada .
    ?Parades :esAccessible ?accessible .
    ?Viatges :esViatgeDe ?Linies .
    ?Linies :nom ?nomLinia
    FILTER regex(str(?Linies),"4__1___")
}
ORDER BY ?nomParada
```

6.1.2. Parades comunes a dues línies

La segona consulta ens dóna les parades de transbord entre dues línies.

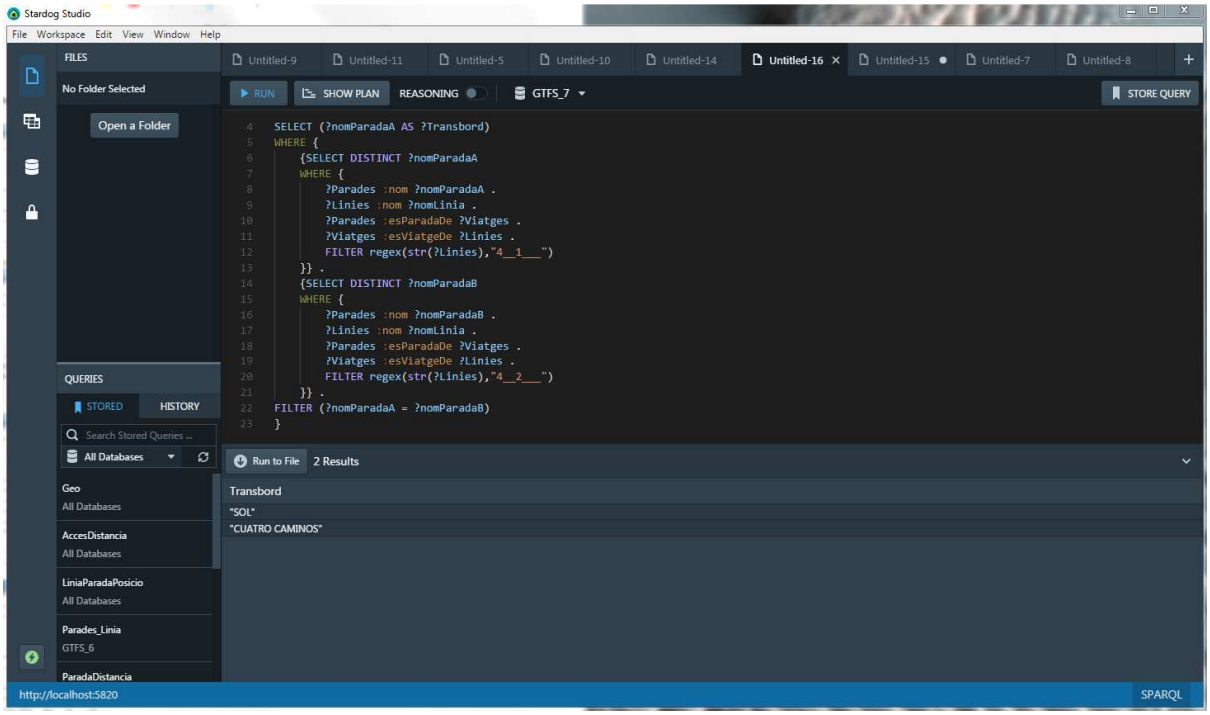


Figura 12: Consulta SPARQL sobre parades comunes

Consulta 2

```
PREFIX owl:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX gtfs:<http://www.semanticweb.org/alopezvic/ontologies/2018/10/GTFS#>

SELECT (?nomParadaA AS ?Transbord)
WHERE {
    {SELECT DISTINCT ?nomParadaA
    WHERE {
        ?Parades :nom ?nomParadaA .
        ?Linies :nom ?nomLinia .
        ?Parades :esParadaDe ?Viatges .
        ?Viatges :esViatgeDe ?Linies .
        FILTER regex(str(?Linies),"4__1___")
    }} .
    {SELECT DISTINCT ?nomParadaB
    WHERE {
        ?Parades :nom ?nomParadaB .
        ?Linies :nom ?nomLinia .
        ?Parades :esParadaDe ?Viatges .
        ?Viatges :esViatgeDe ?Linies .
        FILTER regex(str(?Linies),"4__2___")
    }} .
FILTER (?nomParadaA = ?nomParadaB)
}
```

6.1.3. Temps de pas per una parada

Aquesta consulta mostra el temps de pas per una parada, comptant des de l'inici del viatge així com el número d'ordre de la parada en la línia, tant en el sentit d'anada com el de tornada.

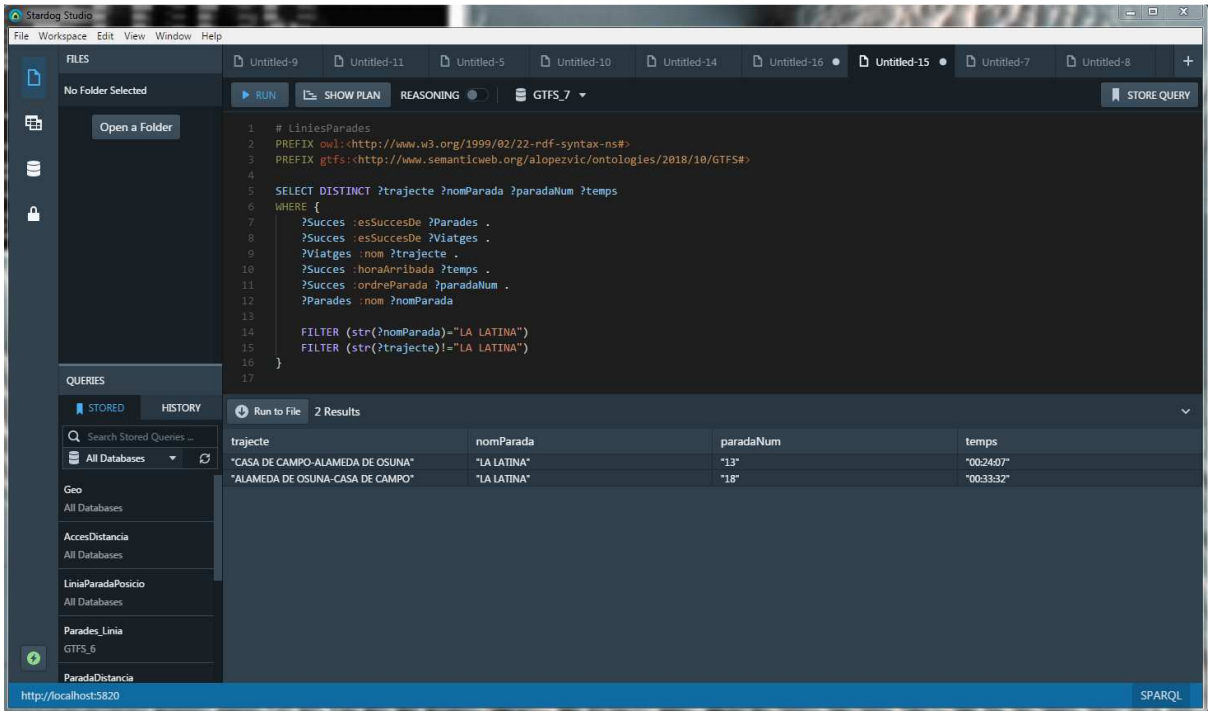


Figura 13: Consulta SPARQL sobre temps de pas

Consulta 3

```
PREFIX owl:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX gtfs:<http://www.semanticweb.org/alopezvic/ontologies/2018/10/GTFS#>

SELECT DISTINCT ?trajecte ?nomParada ?paradaNum ?temps
WHERE {
    ?Succes :esSuccesDe ?Parades .
    ?Succes :esSuccesDe ?Viatges .
    ?Viatges :nom ?trajecte .
    ?Succes :horaArribada ?temps .
    ?Succes :ordreParada ?paradaNum .
    ?Parades :nom ?nomParada

    FILTER (str(?nomParada)="LA LATINA")
    FILTER (str(?trajecte)!="LA LATINA")
}
```

6.1.4. Accessos dins un radi

La següent consulta ens dóna els tres accessos i la seva accessibilitat situats a menys de 400 metres d'un punt geogràfic utilitzant la funció *distance* de GeoSPARQL.

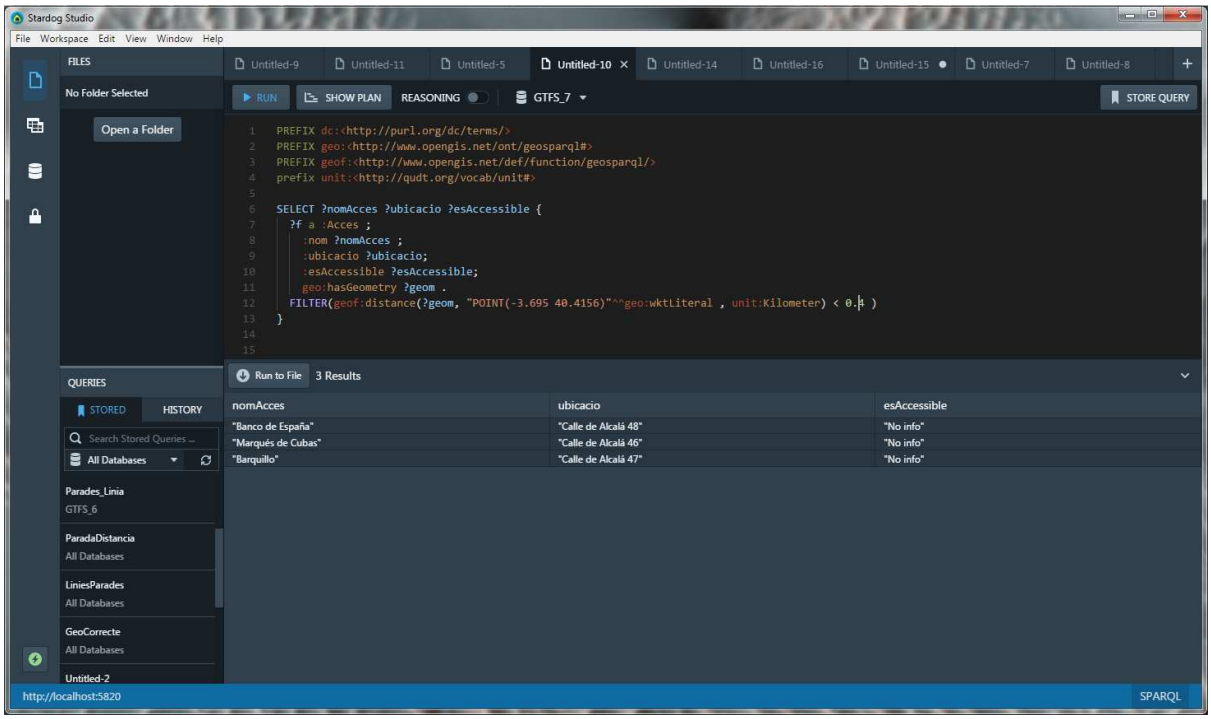


Figura 14: Consulta SPARQL sobre accessos propers

Consulta 4

```
PREFIX dc:<http://purl.org/dc/terms/>
PREFIX geo:<http://www.opengis.net/ont/geosparql#>
PREFIX geof:<http://www.opengis.net/def/function/geosparql/>
prefix unit:<http://qudt.org/vocab/unit#>

SELECT ?nomAcces ?ubicacio ?esAccessible {
  ?f a :Acces ;
    :nom ?nomAcces ;
    :ubicacio ?ubicacio;
    :esAccessible ?esAccessible;
    geo:hasGeometry ?geom .
  FILTER(geof:distance(?geom, "POINT(-3.695 40.4156)"^^geo:wktLiteral , unit:Kilometer) < 0.4 )
}
```

6.1.5. Parades dins un polígon

Aquesta consulta ens dona les parades i les línies que hi ha dintre d'un polígon, a partir de les propietats geo:lat i geo:long d'una Parada.

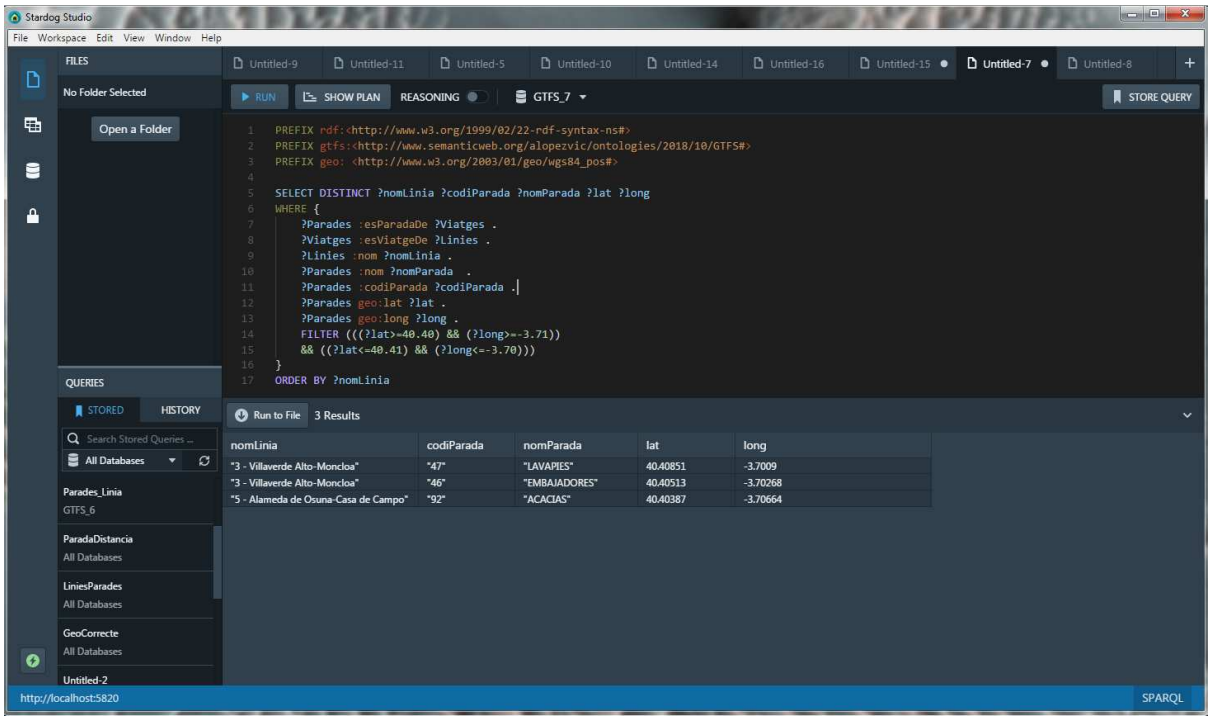


Figura 15: Consulta SPARQL sobre parades en un polígon

Consulta 5

```
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX gtfs:<http://www.semanticweb.org/alopezvic/ontologies/2018/10/GTFS#>
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>

SELECT DISTINCT ?nomLinia ?codiParada ?nomParada ?lat ?long
WHERE {
    ?Parades :esParadaDe ?Viatges .
    ?Viatges :esViatgeDe ?Linies .
    ?Linies :nom ?nomLinia .
    ?Parades :nom ?nomParada  .
    ?Parades :codiParada ?codiParada .
    ?Parades geo:lat ?lat .
    ?Parades geo:long ?long .
    FILTER (((?lat>=40.40) && (?long>=-3.71))
    && ((?lat<=40.41) && (?long<=-3.70)))
}
ORDER BY ?nomLinia
```

6.1.6. Parades properes a un punt d'interès

Finalment, en aquesta consulta intentava trobar la distància entre un punt, obtingut mitjançant un servei contra el "Punto SPARQL" de l'Instituto Geográfico Nacional i les parades més properes. Però no he aconseguit que la funció distance interpreti la geometria que li proporciona el servei. Stardog recomana fer servir el format BIND, però no he aconseguit que proporcioni valors. Com es pot veure, substituint la variable per un wktLiteral no hi ha problema.

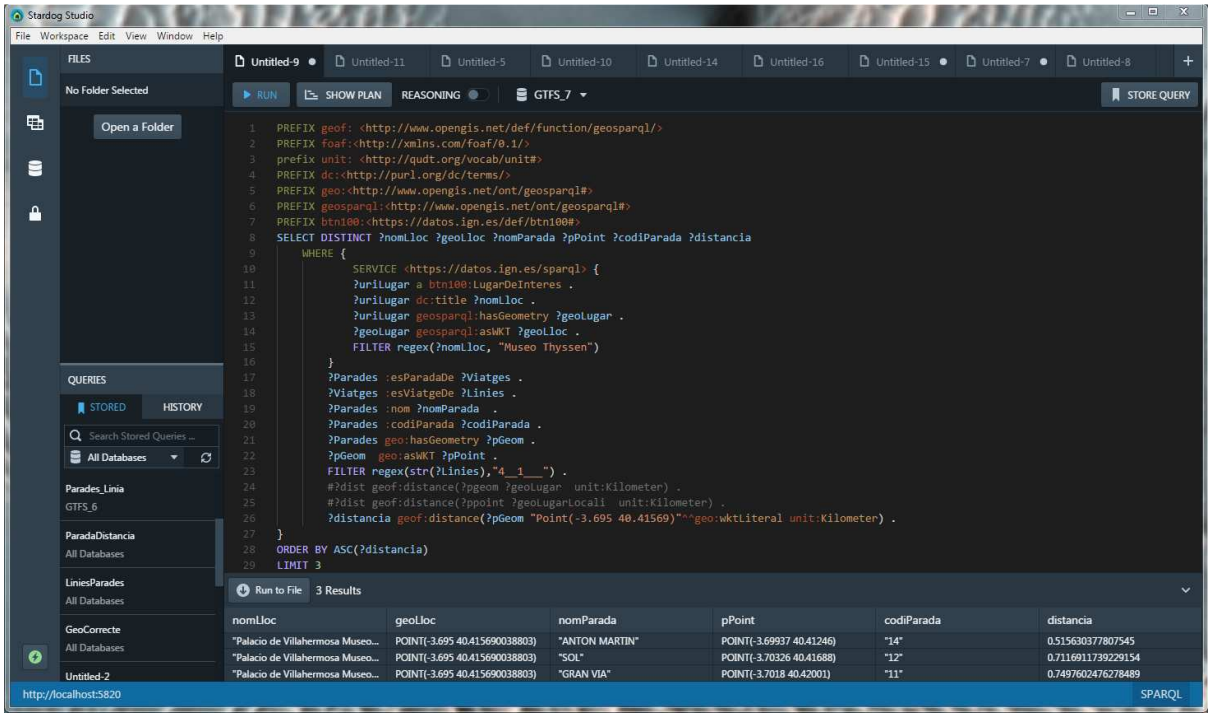


Figura 16: Consulta SPARQL sobre parades properes a un punt d'interés

Consulta 6

```
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX foaf:<http://xmlns.com/foaf/0.1/>
prefix unit: <http://qudt.org/vocab/unit#>
PREFIX dc:<http://purl.org/dc/terms/>
PREFIX geo:<http://www.opengis.net/ont/geosparql#>
PREFIX geosparql:<http://www.opengis.net/ont/geosparql#>
PREFIX btn100:<https://datos.ign.es/def/btn100#>
SELECT DISTINCT ?nombre ?geoLugarLocali ?nm ?ppoint ?cd ?dist
    WHERE {
            SERVICE <https://datos.ign.es/sparql> {
            ?uriLugar a btn100:LugarDeInteres .
            ?uriLugar dc:title ?nombre .
            ?uriLugar geosparql:hasGeometry ?geoLugar .
            ?geoLugar geosparql:asWKT ?geoLugarLocali .
            FILTER regex(?nombre, "Museo Thyssen")
        }
        ?Parades :esParadaDe ?Viatges .
        ?Viatges :esViatgeDe ?Linies .
        ?Parades :nom ?nm  .
        ?Parades :codiParada ?cd .
        ?Parades geo:hasGeometry ?pgeom .
        ?pgeom  geo:asWKT ?ppoint .
        FILTER regex(str(?Linies),"4__1___") .
        #?dist geof:distance(?pgeom ?geoLugar  unit:Kilometer) .
        ?dist geof:distance(?pgeom "Point(-3.695 40.41569)"^^geo:wktLiteral unit:Kilometer) .
}
ORDER BY ASC(?dist)
LIMIT 3
```

7. Conclusions i línies de futur

7.1. Conclusions

Aquest treball m'ha proporcionat la possibilitat de treballar en tot el procés de creació d'una ontologia pràcticament des de zero, amb els diferents problemes que comporta. Així mateix, m'ha ajudat a consolidar molts dels conceptes que vaig estudiar a l'assignatura de Gestió del Coneixement. He vist les possibilitats que té treballar en ontologies a l'hora de compartir informació i m'ha proporcionat un sistema de treball per crear-les.

Els objectius del treball crec que s'han aconseguit. Del contingut del paquet GTFS només ha faltat carregar els calendaris i les freqüències, cosa que estava prevista en les darreres iteracions.

En relació a la planificació, s'ha seguit, en termes generals. Crec que la metodologia és adequada. Com a propostes de millora, potser valdria la pena avançar una mica la creació de l'ontologia amb Protégé, per donar més de temps a la part final de consultes SPARQL. També caldria haver creat un paquet de proves que contingués les dades de només tres o quatre línies, de manera que, una vegada carregades les dades, s'hagués pogut treballar amb Protégé, creant regles i comprovant les inferències derivades d'elles.

També, amb més temps, es podrien haver provat fer les consultes en java amb l'ARQ de Jena i realitzar una interfície de càrrega i consulta de cara a proporcionar als usuaris una interfície amigable.

Com a reflexió personal, m'ha costat pensar de manera diferent a quan analitzo la informació en una BD clàssica, amb consultes SQL, potser per l'estructura relacional dels fitxers del GTFS.

7.2. Línies de futur

A partir del treball s'obre una sèrie de possibilitats molt interessants. A banda de carregar els calendaris, ja he comentat la possibilitat de realitzar una aplicació amb una interfície més amigable per la càrrega dels paquets GTFS, que permetés realitzar consultes SPARQL sobre les dades.

Un altre possibilitat seria establir regles que permetessin trobar camins mínims de temps entre dos punts geogràfics utilitzant les diferents xarxes de transport. Per això caldria crear uns paquets de proves amb un número d'individus controlat, carregar totes les xarxes, trobar els enllaços entre elles i establir regles, calculant els temps de recorregut i valorant altres variables.

I, finalment, també es podria plantejar utilitzar l'ontologia per crear un punt SPARQL per tal que es puguin realitzar consultes sense haver de descarregar fitxers, de manera que un servei pugués accedir a les dades.

7.3. Per acabar…

Com he comentat al principi, una de les coses que m'han sorprès al realitzar aquest treball és la poca "popularitat" del concepte Web Semàntica. Aquest gràfic mostra les visites a les pàgines de la Wikipèdia sobre la Web Semàntica i sobre la Intel·ligència Artificial, en el període màxim disponible.

Com es pot veure, la “curiositat” sobre la primera resta estable en un nivell de visites baix en relació al nombre de visites de la segona, que és molt superior i va creixent.

Figura 17: Comparativa de visites a la Wikipèdia

En un món en el que l’Internet de la Coses és cada cop més real, la comunicació semàntica entre màquines hauria de ser cada cop més important si no volem una intel·ligència artificial autista.

Bibliografia

General

Duran Cals, Jordi, Conesa i Caralt, Jordi i Clarisó Viladrosa, Robert. Ontologies i web semàntica. Materials UOC

Rodríguez, José Ramón (coord), Mariné Jové, Pere. Gestió de projectes. Materials UOC

Rubí Aguiló, Fernando. Ontologia de punts d'accés WIFI. Treball de final de Grau. UOC. Gener 2018. http://openaccess.uoc.edu/webapps/o2/bitstream/10609/73028/7/fernandorubiTFG0118mem%c3%b2ria.pdf (consultat octubre 2018)

Web Semàntica

https://www.w3.org/2001/sw/wiki/Main_Page (consultat octubre 2018)

https://en.wikipedia.org/wiki/Semantic_Web (consultat octubre 2018)

Ontologia

N. Guarino, editor. Formal Ontology in Information Systems: Proceedings of the First International Conference (FOIS'98). Ios Press Inc, 1998.

Gruber, Tom. Ontology. in the *Encyclopedia of Database Systems*, Ling Liu and M. Tamer Özsu (Eds.), Springer-Verlag, 2009

http://tomgruber.org/writing/ontology-definition-2007.htm (consultat octubre 2018)

https://www.w3.org/2001/sw/wiki/OWL (consultat octubre 2018)

https://en.wikipedia.org/wiki/Ontology_(information_science) (consultat octubre 2018)

W3C

https://en.wikipedia.org/wiki/World_Wide_Web_Consortium (consultat octubre 2018)

https://www.w3.org/Consortium/ (consultat octubre 2018)

URI

https://en.wikipedia.org/wiki/Uniform_Resource_Identifier (consultat octubre 2018)

https://www.w3.org/Addressing/#rfc3986 (consultat octubre 2018)

XML

https://www.w3.org/XML/#wgs (consultat octubre 2018)

https://en.wikipedia.org/wiki/XML (consultat octubre 2018)

RDF

https://www.w3.org/2001/sw/wiki/RDF (consultat octubre 2018)

https://en.wikipedia.org/wiki/Resource_Description_Framework (consultat octubre 2018)

OWL

https://www.w3.org/OWL/ (consultat octubre 2018)

https://en.wikipedia.org/wiki/Web_Ontology_Language (consultat octubre 2018)

SPARQL

https://www.w3.org/2001/sw/wiki/SPARQL (consultat octubre 2018)

https://en.wikipedia.org/wiki/SPARQL (consultat octubre 2018)

https://query.wikidata.org/ (consultat octubre 2018)

GeoSPARQL

https://www.opengeospatial.org/standards/geosparql (consultat octubre 2018)

http://www.geosparql.org/ (consultat 2018)

https://www.w3.org/2011/02/GeoSPARQL.pdf (consultat 2018)

Open data

http://opendefinition.org/od/1.1/ca/ (consultat octubre 2018)

https://en.wikipedia.org/wiki/Open_data (consultat octubre 2018)

http://data.europa.eu/euodp/en/data/ (consultat octubre 2018)

https://creativecommons.org/ (consultat octubre 2018)

https://theodi.org/ (consultat octubre 2018)

http://iniciativabarcelonaopendata.cat/ca/ (consultat octubre 2018)

https://5stardata.info/es/ (consultat octubre 2018)

GTFS

http://gtfs.org/es/ (consultat octubre 2018)

https://en.wikipedia.org/wiki/Transmodel (consultat octubre 2018)

http://www.transmodel-cen.eu/wp-content/uploads/sites/2/Use-of-UML-in-Tranmodel.pdf (consultat octubre 2018)

http://datos.crtm.es/ (consulta octubre 2018)

Recursos

https://protege.stanford.edu/ (consultat octubre 2018)

https://jena.apache.org/ (consultat octubre 2018)

https://www.stardog.com/ (consultat octubre 2018)

Disseny d'una ontologia

Noy, Natalya F.; McGuiness, Deborah L. Ontology Developement 101: A Guide to Creating Your First Ontology

https://protege.stanford.edu/publications/ontology_development/ontology101.pdf (consultat octubre 2018)

Allemang, Dean i Hendler, Jim. Semantic Web for the Working Ontologist 2n edition. Morgan Kaufmann 2011

Disseny

Houda, Mnasser i altres. A public transportaion ontology to support use travel plannig

https://www.researchgate.net/publication/224155014_A_public_transportation_ontology_to_support_user_travel_planning (consultat novembre 2018)

Keller, Christine i altres. Introducing the Public Transport Domain to the Web of Data
https://link.springer.com/chapter/10.1007/978-3-319-11746-1_38 (consultat novembre 2018)
Katsumi, Megan; Fox, Mark. Ontologies for transportation research: A survey
https://pdfs.semanticscholar.org/9ac6/8f205ab43995d88633fed1e6be9884f410e7.pdf?_ga=2.217749904.747794369.1544786473-180314185.1544786473 (consultat novembre 2018)

Construcció i poblament

https://old-homepages.abdn.ac.uk/jeff.z.pan/pages/teaching/CS5010ma/practicals/practical-onto-xslt-protege.html (consultat novembre 2018)
https://www.stardog.com/blog/geospatial-a-primer/ (consultat desembre 2018)
http://datos.ign.es/casos-de-uso.html (consultat desembre 2018)

Annexos

Annex A: Glossari

CSV – Arxiu de text separat per comes.

GTFS – Format obert per compartir informació sobre una xarxa de transport públic

OWL – Família de llenguatges de representació del coneixement per crear ontologies.

RDF – Model conceptual per definir dades en la Web Semàntica.

SPARQL – Llenguatge estandarditzat per la consulta d'arxius RDF.

TTL – Format que permet serialitzar RDF.

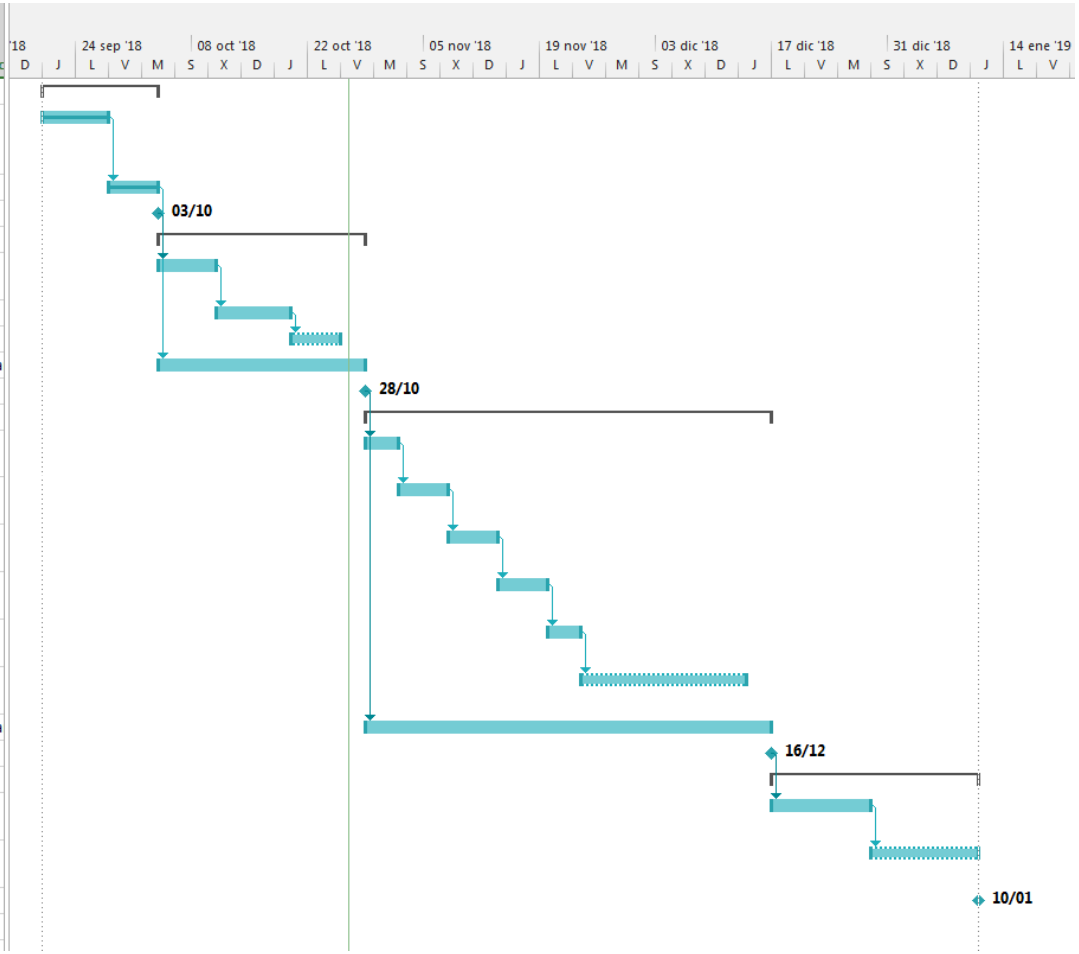
W3C – Consorci internacional que treballa per desenvolupar i promocionar estàndards pel WWW.

WWW – Espai d'informació on els documents i altres recursos estan enllaçats.

XML – Llenguatges d'etiquetes que pot ser llegit per màquines o humans.

Annex B: Diagrama de GANTT

	M de ta	Nom de la tasca	Durada	Inici	Final		Observac
1		**PAC 1**	**10 días**	**jue 20/09/18**	**mié 03/10/18**		**Fita**
2		Estudi del tema i planificació global del projecte	6 días	jue 20/09/18	jue 27/09/18		Prèvia
3		Redacció PAC 1	4 días	vie 28/09/18	mié 03/10/18	2	Prèvia
4		Entrega PAC1	1 día	mié 03/10/18	mié 03/10/18		
5		**PAC 2**	**17 días**	**jue 04/10/18**	**dom 28/10/18**		**Fita**
6		Estudi sobre la web semàntica, ontologies,...	5 días	jue 04/10/18	mié 10/10/18	4	Fase 1
7		Anàlisi dades GTFS	7 días	jue 11/10/18	vie 19/10/18	6	Fase 1
8		Estudi sobre les eines	5 días	sáb 20/10/18	jue 25/10/18	7	Fase 2
9		Redacció PAC 2	18 días	jue 04/10/18	dom 28/10/18	3	Tasca tra
10		Entrega PAC2	1 día	dom 28/10/18	dom 28/10/18		
11		**PAC 3**	**35 días**	**lun 29/10/18**	**dom 16/12/18**		**Fita**
12		Cerca d'ontologies existents	4 días	lun 29/10/18	jue 01/11/18	10	Fase 3
13		Enumerar els termes rellevants	4 días	vie 02/11/18	mié 07/11/18	12	
14		Definir classes i la jerarquia	4 días	jue 08/11/18	mar 13/11/18	13	
15		Definir les propietats de les classes	4 días	mié 14/11/18	lun 19/11/18	14	
16		Definir les restriccions de les propietats	4 días	mar 20/11/18	vie 23/11/18	15	
17		Població de l'ontologia, consultes i proves.	15 días	sáb 24/11/18	jue 13/12/18	16	Fase 4
18		Redacció PAC 3	36 días	lun 29/10/18	dom 16/12/18	10	Tasca tra
19		Entrega PAC3	1 día	dom 16/12/18	dom 16/12/18		
20		**Lliurament final**	**19 días**	**lun 17/12/18**	**jue 10/01/19**		**Fita**
21		Redacció final de la memòria	10 días	lun 17/12/18	vie 28/12/18	19	Fase 5
22		Preparació de la presentació	10 días	sáb 29/12/18	jue 10/01/19	21	Fase 5
23		Lliurament final	1 día	jue 10/01/19	jue 10/01/19		



Annex C: Especificacions dels fitxers utilitzats del paquet GTFS

agency.txt *[File: Required]*

Field Name	Required	Details
agency_id	Optional	The **agency_id** field is an ID that uniquely identifies a transit agency. A transit feed may represent data from more than one agency. The **agency_id** is dataset unique. This field is optional for transit feeds that only contain data for a single agency.
agency_name	Required	The **agency_name** field contains the full name of the transit agency. Google Maps will display this name.
agency_url	Required	The **agency_url** field contains the URL of the transit agency. The value must be a fully qualified URL that includes **http**:// or **https**://, and any special characters in the URL must be correctly escaped. See http://www.w3.org/Addressing/URL/4_URI_Recommentations.html for a description of how to create fully qualified URL values.
agency_timezone	Required	The **agency_timezone** field contains the timezone where the transit agency is located. Timezone names never contain the space character but may contain an underscore. Please refer to http://en.wikipedia.org/wiki/List_of_tz_zones for a list of valid values. If multiple agencies are specified in the feed, each must have the same agency_timezone.
agency_lang	Optional	The **agency_lang** field contains a IETF BCP 47 language code specifying the primary language used by this transit agency. This setting helps GTFS consumers choose capitalization rules and other language-specific settings for the feed. For an introduction to IETF BCP 47, please refer to http://www.rfc-editor.org/rfc/bcp/bcp47.txt and http://www.w3.org/International/articles/language-tags/.
agency_phone	Optional	The **agency_phone field** contains a single voice telephone number for the specified agency. This field is a string value that presents the telephone number as typical for the agency's service area. It can and should contain punctuation marks to group the digits of the number. Dialable text (for example, TriMet's "503-238-RIDE") is permitted, but the field must not contain any other descriptive text.
agency_fare_url	Optional	The **agency_fare_url** specifies the URL of a web page that allows a rider to purchase tickets or other fare instruments for that agency online. The value must be a fully qualified URL that includes **http**:// or **https**://, and any special characters in the URL must be correctly escaped. See http://www.w3.org/Addressing/URL/4_URI_Recommentations.html for a description of how to create fully qualified URL values.
agency_email	Optional	Contains a single valid email address actively monitored by the agency's customer service department. This email address will be considered a direct contact point where transit riders can reach a customer service representative at the agency.

routes.txt *[File: Required]*

Field Name	Required	Details
route_id	**Required**	The **route_id** field contains an ID that uniquely identifies a route. The route_id is dataset unique.
agency_id	Optional	The **agency_id** field defines an agency for the specified route. This value is referenced from the agency.txt file. Use this field when you are providing data for routes from more than one agency.
route_short_name	**Conditionally required**	The **route_short_name** contains the short name of a route. This will often be a short, abstract identifier like "32", "100X", or "Green" that riders use to identify a route, but which doesn't give any indication of what places the route serves. At least one of *route_short_name* or *route_long_name* must be specified, or potentially both if appropriate. If the route does not have a short name, please specify a *route_long_name* and use an empty string as the value for this field.
route_long_name	**Conditionally required**	The **route_long_name** contains the full name of a route. This name is generally more descriptive than the *route_short_name* and will often include the route's destination or stop. At least one of *route_short_name* or *route_long_name* must be specified, or potentially both if appropriate. If the route does not have a long name, please specify a *route_short_name* and use an empty string as the value for this field.
route_desc	Optional	The **route_desc** field contains a description of a route. Please provide useful, quality information. Do not simply duplicate the name of the route. For example, "**A** trains operate between Inwood-207 St, Manhattan and Far Rockaway-Mott Avenue, Queens at all times. Also from about 6AM until about midnight, additional **A** trains operate between Inwood-207 St and Lefferts Boulevard (trains typically alternate between Lefferts Blvd and Far Rockaway)."
route_type	**Required**	The **route_type** field describes the type of transportation used on a route. Valid values for this field are: * **0** - Tram, Streetcar, Light rail. Any light rail or street level system within a metropolitan area. * **1** - Subway, Metro. Any underground rail system within a metropolitan area. * **2** - Rail. Used for intercity or long-distance travel. * **3** - Bus. Used for short- and long-distance bus routes. * **4** - Ferry. Used for short- and long-distance boat service. * **5** - Cable car. Used for street-level cable cars where the cable runs beneath the car. * **6** - Gondola, Suspended cable car. Typically used for aerial cable cars where the car is suspended from the cable. * **7** - Funicular. Any rail system designed for steep inclines.
route_url	Optional	The **route_url** field contains the URL of a web page about that particular route. This should be different from the agency_url. The value must be a fully qualified URL that includes **http**:// or **https**://, and any special characters in the URL must be correctly escaped. See http://www.w3.org/Addressing/URL/4_URI_Recommentations.html for a description of how to create fully qualified URL values.
route_color	Optional	In systems that have colors assigned to routes, the **route_color** field defines a color that corresponds to a route. The color must be provided as a six-character hexadecimal number, for example, 00FFFF. If no color is specified, the default route color is white (FFFFFF). The color difference between **route_color** and **route_text_color** should provide sufficient contrast when viewed on a black and white screen. The W3C Techniques for Accessibility Evaluation And Repair Tools document offers a useful algorithm for evaluating color contrast. There are also helpful online tools for choosing contrasting colors, including the snook.ca Color Contrast Check application.
route_text_color	Optional	The route_text_color field can be used to specify a legible color to use for text drawn against a background of route_color. The color must be provided as a six-character hexadecimal number, for example, FFD700. If no color is specified, the default text color is black (000000). The color difference between **route_color** and **route_text_color** should provide sufficient contrast when viewed on a black and white screen.
route_sort_order	Optional	The **route_sort_order** field can be used to order the routes in a way which is ideal for presentation to customers. It must be a non-negative integer. Routes with smaller **route_sort_order** values should be displayed before routes with larger **route_sort_order** values.

stop_times.txt *[File: Required]*

<table>
<tr><th>Field Name</th><th>Required</th><th>Details</th></tr>
<tr><td>trip_id</td><td>Required</td><td>The trip_id field contains an ID that identifies a trip. This value is referenced from the trips.txt file.</td></tr>
<tr><td>arrival_time</td><td>Required</td><td>The arrival_time specifies the arrival time at a specific stop for a specific trip on a route. The time is measured from "noon minus 12h" (effectively midnight, except for days on which daylight savings time changes occur) at the beginning of the service day. For times occurring after midnight on the service day, enter the time as a value greater than 24:00:00 in HH:MM:SS local time for the day on which the trip schedule begins. If you don't have separate times for arrival and departure at a stop, enter the same value for arrival_time and departure_time.

Scheduled stops where the vehicle strictly adheres to the specified arrival and departure times are timepoints. For example, if a transit vehicle arrives at a stop before the scheduled departure time, it will hold until the departure time. If this stop is not a timepoint, use either an empty string value for the arrival_time field or provide an interpolated time. Further, indicate that interpolated times are provided via the timepoint field with a value of zero. If interpolated times are indicated with timepoint=0, then time points must be indicated with a value of 1 for the timepoint field. Provide arrival times for all stops that are time points.

An arrival time must be specified for the first and the last stop in a trip. Times must be eight digits in HH:MM:SS format (H:MM:SS is also accepted, if the hour begins with 0). Do not pad times with spaces. The following columns list stop times for a trip and the proper way to express those times in the arrival_time field:
<table>
<tr><th>Time</th><th>arrival_time value</th></tr>
<tr><td>08:10:00 A.M.</td><td>08:10:00 or 8:10:00</td></tr>
<tr><td>01:05:00 P.M.</td><td>13:05:00</td></tr>
<tr><td>07:40:00 P.M.</td><td>19:40:00</td></tr>
<tr><td>01:55:00 A.M.</td><td>25:55:00</td></tr>
</table>
Note: Trips that span multiple dates will have stop times greater than 24:00:00. For example, if a trip begins at 10:30:00 p.m. and ends at 2:15:00 a.m. on the following day, the stop times would be 22:30:00 and 26:15:00. Entering those stop times as 22:30:00 and 02:15:00 would not produce the desired results.</td></tr>
<tr><td>departure_time</td><td>Required</td><td>The departure_time specifies the departure time from a specific stop for a specific trip on a route. The time is measured from "noon minus 12h" (effectively midnight, except for days on which daylight savings time changes occur) at the beginning of the service day. For times occurring after midnight on the service day, enter the time as a value greater than 24:00:00 in HH:MM:SS local time for the day on which the trip schedule begins. If you don't have separate times for arrival and departure at a stop, enter the same value for arrival_time and departure_time.

Scheduled stops where the vehicle strictly adheres to the specified arrival and departure times are timepoints. For example, if a transit vehicle arrives at a stop before the scheduled departure time, it will hold until the departure time. If this stop is not a timepoint, use either an empty string value for the departure_time field or provide an interpolated time (further, indicate that interpolated times are provided via the timepoint field with a value of zero). If interpolated times are indicated with timepoint=0, then time points must be indicated with a value of 1 for the timepoint field. Provide departure times for all stops that are time points.

A departure time must be specified for the first and the last stop in a trip even if the vehicle does not allow boarding at the last stop. Times must be eight digits in HH:MM:SS format (H:MM:SS is also accepted, if the hour begins with 0). Do not pad times with spaces. The following columns list stop times for a trip and the proper way to express those times in the departure_time field:
<table>
<tr><th>Time</th><th>departure_time value</th></tr>
<tr><td>08:10:00 A.M.</td><td>08:10:00 or 8:10:00</td></tr>
<tr><td>01:05:00 P.M.</td><td>13:05:00</td></tr>
<tr><td>07:40:00 P.M.</td><td>19:40:00</td></tr>
<tr><td>01:55:00 A.M.</td><td>25:55:00</td></tr>
</table>
Note: Trips that span multiple dates will have stop times greater than 24:00:00. For example, if a trip begins at 10:30:00 p.m. and ends at 2:15:00 a.m. on the following day, the stop times would be 22:30:00 and 26:15:00. Entering those stop times as 22:30:00 and 02:15:00 would not produce the desired results.</td></tr>
<tr><td>stop_id</td><td>Required</td><td>The stop_id field contains an ID that uniquely identifies a stop. Multiple routes may use the same stop. The stop_id is referenced from the stops.txt file. If location_type is used in stops.txt, all stops referenced in stop_times.txt must have location_type of 0. Where possible, stop_id values should remain consistent between feed updates. In other words, stop A with stop_id 1 should have stop_id 1 in all subsequent data updates. If a stop is not a time point, enter blank values for arrival_time and departure_time.</td></tr>
<tr><td>stop_sequence</td><td>Required</td><td>The stop_sequence field identifies the order of the stops for a particular trip. The values for stop_sequence must be non-negative integers, and they must increase along the trip. For example, the first stop on the trip could have a stop_sequence of 1, the second stop on the trip could have a stop_sequence of 23, the third stop could have a stop_sequence of 40,</td></tr>
</table>

		and so on.
stop_headsign	Optional	The **stop_headsign** field contains the text that appears on a sign that identifies the trip's destination to passengers. Use this field to override the default **trip_headsign** when the headsign changes between stops. If this headsign is associated with an entire trip, use **trip_headsign** instead.
pickup_type	Optional	The **pickup_type** field indicates whether passengers are picked up at a stop as part of the normal schedule or whether a pickup at the stop is not available. This field also allows the transit agency to indicate that passengers must call the agency or notify the driver to arrange a pickup at a particular stop. Valid values for this field are: * **0** - Regularly scheduled pickup * **1** - No pickup available * **2** - Must phone agency to arrange pickup * **3** - Must coordinate with driver to arrange pickup The default value for this field is **0**.
drop_off_type	Optional	The **drop_off_type** field indicates whether passengers are dropped off at a stop as part of the normal schedule or whether a drop off at the stop is not available. This field also allows the transit agency to indicate that passengers must call the agency or notify the driver to arrange a drop off at a particular stop. Valid values for this field are: * **0** - Regularly scheduled drop off * **1** - No drop off available * **2** - Must phone agency to arrange drop off * **3** - Must coordinate with driver to arrange drop off The default value for this field is **0**.
shape_dist_traveled	Optional	When used in the stop_times.txt file, the **shape_dist_traveled** field positions a stop as a distance from the first shape point. The **shape_dist_traveled** field represents a real distance traveled along the route in units such as feet or kilometers. For example, if a bus travels a distance of 5.25 kilometers from the start of the shape to the stop, the **shape_dist_traveled** for the stop ID would be entered as "5.25". This information allows the trip planner to determine how much of the shape to draw when showing part of a trip on the map. The values used for **shape_dist_traveled** must increase along with **stop_sequence**: they cannot be used to show reverse travel along a route. The units used for **shape_dist_traveled** in the stop_times.txt file must match the units that are used for this field in the shapes.txt file.
timepoint	Optional	The timepoint field can be used to indicate if the specified arrival and departure times for a stop are strictly adhered to by the transit vehicle or if they are instead approximate and/or interpolated times. The field allows a GTFS producer to provide interpolated stop times that potentially incorporate local knowledge, but still indicate if the times are approximate. For stop-time entries with specified arrival and departure times, valid values for this field are: * **empty** - Times are considered exact. * **0** - Times are considered approximate. * **1** - Times are considered exact. For stop-time entries without specified arrival and departure times, feed consumers must interpolate arrival and departure times. Feed producers may optionally indicate that such an entry is not a timepoint (value=0) but it is an error to mark a entry as a timepoint (value=1) without specifying arrival and departure times.

stops.txt *[File: Required]*

<table>
<tr><th>Field Name</th><th>Required</th><th>Details</th></tr>
<tr><td>stop_id</td><td>Required</td><td>The stop_id field contains an ID that uniquely identifies a stop, station, or station entrance. Multiple routes may use the same stop. The stop_id is used by systems as an internal identifier of this record (e.g., primary key in database), and therefore the stop_id must be dataset unique.</td></tr>
<tr><td>stop_code</td><td>Optional</td><td>The stop_code field contains short text or a number that uniquely identifies the stop for passengers. Stop codes are often used in phone-based transit information systems or printed on stop signage to make it easier for riders to get a stop schedule or real-time arrival information for a particular stop. The stop_code field contains short text or a number that uniquely identifies the stop for passengers. The stop_code can be the same as stop_id if it is passenger-facing. This field should be left blank for stops without a code presented to passengers.</td></tr>
<tr><td>stop_name</td><td>Required</td><td>The stop_name field contains the name of a stop, station, or station entrance. Please use a name that people will understand in the local and tourist vernacular.</td></tr>
<tr><td>stop_desc</td><td>Optional</td><td>The stop_desc field contains a description of a stop. Please provide useful, quality information. Do not simply duplicate the name of the stop.</td></tr>
<tr><td>stop_lat</td><td>Required</td><td>The stop_lat field contains the latitude of a stop, station, or station entrance. The field value must be a valid WGS 84 latitude.</td></tr>
<tr><td>stop_lon</td><td>Required</td><td>The stop_lon field contains the longitude of a stop, station, or station entrance. The field value must be a valid WGS 84 longitude value from -180 to 180.</td></tr>
<tr><td>zone_id</td><td>Optional</td><td>The zone_id field defines the fare zone for a stop ID. Zone IDs are required if you want to provide fare information using fare_rules.txt. If this stop ID represents a station, the zone ID is ignored.</td></tr>
<tr><td>stop_url</td><td>Optional</td><td>The stop_url field contains the URL of a web page about a particular stop. This should be different from the agency_url and the route_url fields. The value must be a fully qualified URL that includes http:// or https://, and any special characters in the URL must be correctly escaped. See http://www.w3.org/Addressing/URL/4_URI_Recommentations.html for a description of how to create fully qualified URL values.</td></tr>
<tr><td>location_type</td><td>Optional</td><td>The location_type field identifies whether this stop ID represents a stop, station, or station entrance. If no location type is specified, or the location_type is blank, stop IDs are treated as stops. Stations may have different properties from stops when they are represented on a map or used in trip planning. The location type field can have the following values:
* 0 or blank - Stop. A location where passengers board or disembark from a transit vehicle.
* 1 - Station. A physical structure or area that contains one or more stop.
* 2 - Station Entrance/Exit. A location where passengers can enter or exit a station from the street. The stop entry must also specify a parent_station value referencing the stop ID of the parent station for the entrance.</td></tr>
<tr><td>parent_station</td><td>Optional</td><td>For stops that are physically located inside stations, the parent_station field identifies the station associated with the stop. To use this field, stops.txt must also contain a row where this stop ID is assigned location type=1.
<table>
<tr><th>This stop ID represents...</th><th>This entry's location type...</th><th>This entry's parent_station field contains...</th></tr>
<tr><td>A stop located inside a station.</td><td>0 or blank</td><td>The stop ID of the station where this stop is located. The stop referenced by parent_station must have location_type=1.</td></tr>
<tr><td>A stop located outside a station.</td><td>0 or blank</td><td>A blank value. The parent_station field doesn't apply to this stop.</td></tr>
<tr><td>A station.</td><td>1</td><td>A blank value. Stations can't contain other stations.</td></tr>
</table></td></tr>
<tr><td>stop_timezone</td><td>Optional</td><td>The stop_timezone field contains the timezone in which this stop, station, or station entrance is located. Please refer to Wikipedia List of Timezones for a list of valid values. If omitted, the stop should be assumed to be located in the timezone specified by agency_timezone in agency.txt. When a stop has a parent station, the stop is considered to be in the timezone specified by the parent station's stop_timezone value. If the parent has no stop_timezone value, the stops that belong to that station are assumed to be in the timezone specified by agency_timezone, even if the stops have their own stop_timezone values. In other words, if a given stop has a parent_station value, any stop_timezone value specified for that stop must be ignored. Even if stop_timezone values are provided in stops.txt, the times in stop_times.txt should continue to be specified as time since midnight in the timezone specified by agency_timezone in agency.txt. This ensures that the time values in a trip always increase over the course of a trip, regardless of which timezones the trip crosses.</td></tr>
</table>

wheelchair_boarding	Optional	The **wheelchair_boarding field** identifies whether wheelchair boardings are possible from the specified stop, station, or station entrance. The field can have the following values: * **0** (or empty) - indicates that there is no accessibility information for the stop * **1** - indicates that at least some vehicles at this stop can be boarded by a rider in a wheelchair * **2** - wheelchair boarding is not possible at this stop When a stop is part of a larger station complex, as indicated by a stop with a **parent_station** value, the stop's **wheelchair_boarding** field has the following additional semantics: * **0** (or empty) - the stop will inherit its **wheelchair_boarding** value from the parent station, if specified in the parent * **1** - there exists some accessible path from outside the station to the specific stop / platform * **2** - there exists no accessible path from outside the station to the specific stop / platform For station entrances, the **wheelchair_boarding** field has the following additional semantics: * **0** (or empty) - the station entrance will inherit its **wheelchair_boarding** value from the parent station, if specified in the parent * **1** - the station entrance is wheelchair accessible (e.g. an elevator is available to platforms if they are not at-grade) * **2** - there exists no accessible path from the entrance to station platforms

trips.txt *[File: Required]*

Field Name	Required	Details
route_id	**Required**	The **route_id** field contains an ID that uniquely identifies a route. This value is referenced from the routes.txt file.
service_id	**Required**	The **service_id** contains an ID that uniquely identifies a set of dates when service is available for one or more routes. This value is referenced from the calendar.txt or calendar_dates.txt file.
trip_id	**Required**	The **trip_id** field contains an ID that identifies a trip. The **trip_id** is dataset unique.
trip_headsign	Optional	The **trip_headsign** field contains the text that appears on a sign that identifies the trip's destination to passengers. Use this field to distinguish between different patterns of service in the same route. If the headsign changes during a trip, you can override the **trip_headsign** by specifying values for the **stop_headsign** field in stop_times.txt.
trip_short_name	Optional	The **trip_short_name** field contains the text that appears in schedules and sign boards to identify the trip to passengers, for example, to identify train numbers for commuter rail trips. If riders do not commonly rely on trip names, please leave this field blank. A **trip_short_name** value, if provided, should uniquely identify a trip within a service day; it should not be used for destination names or limited/express designations.
direction_id	Optional	The **direction_id** field contains a binary value that indicates the direction of travel for a trip. Use this field to distinguish between bi-directional trips with the same **route_id**. This field is not used in routing; it provides a way to separate trips by direction when publishing time tables. You can specify names for each direction with the **trip_headsign** field. * 0 - travel in one direction (e.g. outbound travel) * 1 - travel in the opposite direction (e.g. inbound travel) For example, you could use the *trip_headsign* and *direction_id* fields together to assign a name to travel in each direction for a set of trips. A trips.txt file could contain these rows for use in time tables: * `trip_id,...,trip_headsign,direction_id` * `1234,...,Airport,0` * `1505,...,Downtown,1`
block_id	Optional	The **block_id** field identifies the block to which the trip belongs. A block consists of a single trip or many sequential trips made using the same vehicle, defined by shared service day and block_id. A block_id can have trips with different service days, making distinct blocks. (See example below)
shape_id	Optional	The **shape_id** field contains an ID that defines a shape for the trip. This value is referenced from the shapes.txt file. The shapes.txt file allows you to define how a line should be drawn on the map to represent a trip.
wheelchair_accessible	Optional	* **0** (or empty) - indicates that there is no accessibility information for the trip * **1** - indicates that the vehicle being used on this particular trip can accommodate at least one rider in a wheelchair * **2** - indicates that no riders in wheelchairs can be accommodated on this trip
bikes_allowed	Optional	0 (or empty) - indicates that there is no bike information for the trip * **1** - indicates that the vehicle being used on this particular trip can accommodate at least one bicycle * **2** - indicates that no bicycles are allowed on this trip

Example showing blocks and service day

The example below is valid, with distinct blocks every day of the week.

route_id	**trip_id**	**service_id**	**block_id**	*(first stop time)*	*(last stop time)*
red	trip_1	mon-tues-wed-thurs-fri-sat-sun	red_loop	22:00:00	22:55:00
red	trip_2	fri-sat-sun	red_loop	23:00:00	23:55:00
red	trip_3	fri-sat	red_loop	24:00:00	24:55:00
red	trip_4	mon-tues-wed-thurs	red_loop	20:00:00	20:50:00
red	trip_5	mon-tues-wed-thurs	red_loop	21:00:00	21:50:00

Notes on above table: * On Friday into Saturday morning, for example, a single vehicle operates trip_1, trip_2, and trip_3 (10:00 PM through 12:55 AM). Note that the last trip occurs on Saturday, 12:00 AM to 12:55 AM, but is part of the Friday “service day” because the times are 24:00:00 to 24:55:00. * On Monday, Tuesday, Wednesday, and Thursday, a single vehicle operates trip_1, trip_4, and trip_5 in a block from 8:00 PM to 10:55 PM.