

Data flow control framework for indicator-based policy making: An urban planning example

Autor: Bernat Esquirol
Tutors: Javi Ortigosa (AMB)
Josep Vañó (UB)

Abstract	2
Introducció	3
Estat de l'art	4
Context	5
Objectius generals	5
Problemàtiques de dades al PDU	6
Metodologia	9
Rigidesa de les dades	9
Control de flux	10
Objectiu	11
Disseny	13
Diagrama de casos d'ús	13
Planificació	16
Implementació	17
Funcionament a nivell d'usuari	17
Primers passos	17
Interfície d'usuari	17
Funcionament	18
Aplicacions	19
Funcionament intern	19
Tecnologies descartades	19
Tecnologies triades	20
Funcionament intern	20
Estructures de dades	22
Exemple: Equipaments	24
Conclusió	28
Índex de figures	29
Bibliografia	30

Abstract

In this paper we develop a tool called *flowio* to control Urban Master Plan's data flow from the initial phase (external databases) up to the moment they are developed into indicators, going through every intermediate step (functions or studies). Is a tool based on diagrams and pretends to unify the documentation of processes in a consultable data structure.

Resum

En aquest treball es desenvolupa l'eina *flowio* per tal de controlar el flux de dades del Pla Director Urbanístic de l'Àrea Metropolitana de Barcelona, desde que arriben d'un agent extern (bases de dades) fins que es converteixen en indicadors, passant per cada etapa intermitja (estudis o funcions). És una eina basada en diagrames i que pretén unificar la forma de documentar els processos en una estructura de dades consultable.

Introducció

El món en què vivim és ple de dades. En l'era digital es generen, s'emmagatzemen i es processen milers de dades per segon que tenen a veure amb tots els aspectes de la nostra vida.

En el Sector Privat aquest fenomen ha suposat una revolució i un canvi radical en les estratègies operatives. Les dades s'han utilitzat per generar de valor i eficiència, per millorar l'experiència de l'usuari i el que resulta més important, com a eina indispensable per la presa de decisions (Einav and Levin 2014).

En el Sector Públic, cada cop més administracions públiques han evolucionat cap a un model on les dades són un dels pilars de la innovació en la creació de llei (Jetzek, Avital, and Bjorn-Andersen 2014). L'administració mai havia tingut tantes dades al seu abast i això dóna la oportunitat reformar-se i millorar l'eficiència dels processos públics (Davies 2010). La presa de decisions per parts dels ens públics encara té molt camí per recórrer en el món de les dades. Les dades proporcionen informació en diferents àmbits que pot ajudar al legislador a entendre millor quin és l'interès general i què es pot fer per assolir-lo. (“The Promise of Data-Driven Policymaking | Issues in Science and Technology” 1970)

A Catalunya existeixen alguns exemples de projectes públics innovadors, que aprofiten el poder de les dades i de la interconnectivitat per tenir impacte en la societat. És el cas de decidim plataforma oberta i de codi lliure promogut per l'ajuntament de Barcelona per engrescar la participació ciutadana per ciutats i organitzacions (“Decidim Whitepaper :: Decidim Docs” n.d.). O les diverses plataformes de dades obertes (Barcelona, AMB, Generalitat).

L'ajuntament de Barcelona en el seu Manifest a favor de la sobirania tecnològica (“Manifesto in Favour of Technological Sovereignty and Digital Rights for Cities :: Ethical Digital Standards” n.d.), on defensa els drets dels ciutadans i la importància de desenvolupar software gratuït (punt 4), que ofereix més estabilitat, eficiència i interoperabilitat; dades obertes (punt 5), un element necessari per la sobirania tecnològica; i protocols de comunicació oberts (punt 6), que aporten millora a la transparència i comunicació entre entitats públiques i sector privat.

Seguint aquest fil, el treball aquí presentat és un intent per millorar l'eficiència, transparència i claredat dels estudis fets amb dades a l'administració pública, a través d'un nou llenguatge proposat. Les dades però són una matèria complexa, poden contenir biaixos i qui les tracta ha d'evitar amplificar-los o crear-ne de nous. En concret s'analitzarà en l'àmbit de l'urbanisme i les *smart cities*.

Estat de l'art

S'espera que el 2050, 70% de la població viurà en ciutats (actualment ens trobem al 54.5%). Degut a la dinàmica del planejament urbanístic, els canvis tarden anys a materialitzar-se, s'ha de tenir una visió de futur i encarar els problemes presents i futurs. En l'àmbit d'urbanisme la crecscuda de les poblacions urbanes aporta molta complexitat en molts àmbits: energia, medi ambient, transport, etc... el camp de les dades és indispensable per tal d'entendre tots aquests fenòmens (Psyllidis et al. 2015). Aquesta publicació, anomenada 'A Platform for Urban Analytics and Semantic Data Integration in City Planning', explica un mètode per tal de barrejar dades de moltes fonts, incloent sensors i l'anàlisi de les xarxes socials, i creen una plataforma web 'SocialGlass' per integrar-ho i posa exemples d'implementació. És un paper que té el mateix problema que aquest treball: utilitzar les dades heterogènies de la ciutat eficientment, però amb un altre propòsit: analitzar esdeveniments puntuals a la ciutat. El planejament urbanístic és a més llarg termini.

La publicació 'Spatial urban data system: A cloud-enabled big data infrastructure for social and economic urban analytics' (Anejionu et al. 2019) presenta una infraestructura de dades 'SUDS' que és una plataforma de dades urbanes que facilita la recerca i el descobriment social per el Regne Unit. A més fa un recull de polítiques i lleis, per àrea, i també analitza com aplicar els indicadors a zones reduïdes de l'àrea de treball. Aquesta última publicació fa referència a 'Towards cloud based big data analytics for smart future cities' (Khan et al. 2015) que proposa un enfocament similar a través de la plataforma Hadoop. Aquestes dues publicacions dissenyen una estructura de dades global on encabir-hi tots les fonts actives de dades. En el cas d'aquest treball, l'enfocament és menys global i més descentralitzat (s'implementa a nivell d'equip de treball), que malgrat tenir menys consistència, té més facilitat d'implementació.

També el llibre 'Information Systems for Urban Planning' (Laurini 2014) que el capítol 2 'Design methodologies for information systems' ens parla del model d'entitat relació aplicat a urbanisme, i models orientats a objectes.

Per altra banda, hi ha enfocaments diferents com el de (Wilson et al., n.d.) que apliquen tècniques generatives i disseny computacional al planejament urbanístic. Però utilitzen optimització basada en un sistema de dades rígid per intentar adaptar els seus dissenys a la realitat.

Context

L'AMB és l'administració que s'encarrega de planificar i gestionar l'àrea metropolitana de Barcelona, formada per 36 municipis del voltant de Barcelona amb una població de 3.239.337 habitants i una extensió de 636 km quadrats. Té competències en els àmbits de la cohesió social, la planificació territorial i l'urbanisme, la mobilitat, el transport, la gestió de residus, l'abastament d'aigua, la protecció del medi ambient, l'habitatge social, les infraestructures i la promoció econòmica del territori metropolità.

El Pla Director Urbanístic (PDU) de l'AMB (<http://urbanisme.amb.cat/>) és el pla director metropolità que pretén substituir el Pla General Metropolità (PGM) que és l'eina que fa més de 40 anys que ordena els usos del territori. Pretén crear unes regles generals i unes prioritats pel desenvolupament urbanístic i territorial.

L'autor d'aquest treball ha treballat al PDU com a desenvolupador desde el juliol del 2018 fins el maig del 2019, temps durant el qual s'ha analitzat les pràctiques amb les dades i creat aquest document. Al PDU s'hi fan estudis dades GIS per entendre la complexitat de la metròpolis barcelonina, i per tal de crear un marc legal per regular-la.

Objectius generals

El que es creen rutinariament al PDU, com a molts altres departaments de l'administració, són indicadors. Els indicadors no són res més que el resultat final d'un estudi: dades que ens ajuden a treure conclusions i coneixements sobre alguna part del territori (o persones, o empreses, o qualsevol altre unitat de treball). Pretenen conèixer la unitat de treball i estan basats en dades de diverses fonts. És fonamental garantir l'eficiència, la consistència, la transparència de la creació i el manteniment d'aquests indicadors.

Per poder acomplir aquests propòsits hi ha certs objectius genèrics útils que el PDU es marca en la qüestió de les dades:

- Automatitzar processos: si bé automatitzar ajuda molt a l'eficiència, cal assegurar que aquest procés sigui prou flexible, és a dir, petites modificacions a l'estudi no poden reflectir-se en grans canvis en el codi. Per a evitar aquesta situació, resulta útil encarar la creació de l'estudi d'una forma modulars.
- Establir criteris objectius i quantificables per cada indicador: arribar a la veritat darrere les dades i evitar biaixos.
- Establir mecanismes dinàmics d'actualització d'indicadors: la realitat és canviant i per tant també ho són les dades que es recullen. Això ha de poder manifestar-se en els indicadors que s'han d'adaptar-se a la realitat canviant.

- Compartir dades entre entitats públiques: un dels pilars bàsics és la interoperabilitat entre administracions per poder crear un teixit de dades el màxim extens possible.

Problemàtiques de dades al PDU

Al PDU es fan estudis amb moltes dades. La majoria són geodades, és a dir dades amb una geometria associada. La manera d'emmagatzemar és a través de fitxers de geodades (.shp / .gdb). Aquests fitxers s'emmagatzemen a un disc remot comú (J:, G:, K:). Són dades heterogènies i no segueixen cap estructura lògica en concret. Les dades que es reben són provinents de moltes fonts diferents, els criteris del format d'aquestes dades canvien sovint i es reben periòdicament.

Exemples d'aquestes geodades són: el cadastre, que mostra informacions relacionades amb cada parcel·la de l'àrea; el planejament, que es divideix en diferents capes i que té informació del que es construirà; el carrer, que té com a geometries tots els carrers de l'àrea i informació sobre aquests (flux de cotxes,...).

No només hi ha el problema del manteniment de tot de bases de dades massives, sinó que aquestes no encaixen geomètricament entre elles, no parlen de les mateixes unitats d'estudi. Això fa encara més difícil l'estudi amb geodades, ja que vol dir barrejar-les, decidir en quines unitats geomètriques s'ha de treballar, i per tant diferents opinions en la forma d'operar. La primera problemàtica és doncs la falta de rigidesa en les dades entrants.

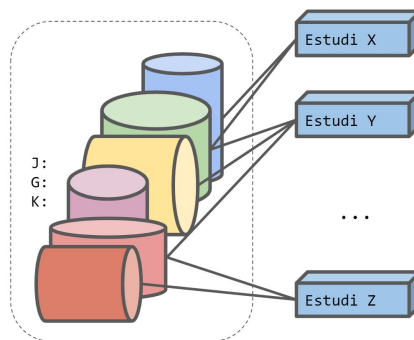


Figure 1. Creació d'estudis amb dades heterogènies

La segona problemàtica és la complexitat en el flux de dades. Per treure coneixement de les dades, es fan estudis. Aquests estudis combinen, modifiquen, corregeixen i deriven aquestes dades. Les dades es modifiquen amb el software de modificació i visualització de dades ArcGIS. Les opcions de modificar dades que dóna aquest software són similar a les que podria donar un Excel, afegir un camp, calcular un camp a través d'operacions de camps anteriors; així com operacions geomètriques: creuar geometries, un join espacial o una superposició de capes geomètriques. falta de un mecanisme de control del flux.

Normalment per modificar les dades el que es fa es crear nous camps en aquestes bases de dades originals i s'omplen de dades derivades de les dades

originals. Que vol dir que cada base de dades està formada per unes certes columnes originals (vermell), i unes quantes columnes derivades (gris) de les originals, que s'han anat creant a l'hora d'analitzar aquestes dades.

DESCRIPCIO	US	ELECTRIFIC	SUPRASERVE	NUMVIES	Shape Length	COBR	TAV	AMPLIEVIA	CO	ROB	F	ACTUACIO	PI	TR	METRO
Bifurcació Pla de Vilanova. Tram ADF 2005	Viatgers	1,5 KV (DC)	FGC Barcelona - Valles	2	536,27459	0	0	Estandard	0	1	0	Sense_Actuacio	0	0	0
TGV Arago - Barcelona -Figures Vilamala. Tram ADF 0160	Viatgers	25 KV 50Hz (CA)	Linia Alta Velocitat	2	77708,249708	0	1	Estandard	0	0	0	Sense_Actuacio	0	0	0
TGV Arago - Barcelona -Figures Vilamala. Tram ADF 0190	Viatgers	25 KV 50Hz (CA)	Linia Alta Velocitat	2	5827,822811	0	1	Estandard	1	0	0	Sense_Actuacio	0	0	0
TGV Arago - Barcelona -Figures Vilamala. Tram ADF 0200	Viatgers	25 KV 50Hz (CA)	Linia Alta Velocitat	2	7961,789225	0	1	Estandard	1	0	0	Sense_Actuacio	0	0	0
TGV Arago - Barcelona -Figures Vilamala. Tram ADF 0220	Mixt	25 KV 50Hz (CA)	Linia Alta Velocitat	2	7906,063972	0	1	Estandard	1	0	0	Sense_Actuacio	0	0	0
TGV Arago - Barcelona -Figures Vilamala. Tram ADF 0300	Mixt	25 KV 50Hz (CA)	Linia Alta Velocitat	2	34890,293998	0	1	Estandard	0	0	0	Sense_Actuacio	0	0	0
Madrid Chamartin-Barcelona Estacin de Franra. Tram ADF 0415	Mixt	3 KV (DC)	Regionals	1	11059,18551	0	0	Iberic	0	1	0	Sense_Actuacio	0	0	0
Madrid Chamartin-Barcelona Estacin de Franra. Tram ADF 0430	Mixt	3 KV (DC)	Regionals	1	28081,466038	0	0	Iberic	0	1	0	Sense_Actuacio	0	0	0

Figure 2. Aspecte de base de dades del PDU

Això cal acompanyar-ho doncs, d'una explicació de com s'han calculat aquests camps, es creen documents Word per explicar aquestes noves columnes com s'han calculat i amb quin objectiu.

```
XarxaFerroviaria.gdb conté el shp XarxaFerroviariaAMB.shp on apareixen les vies de tren de l'AMB segons la base de Geofiscal (Departament de territori i Sostenibilitat, Generalitat de Catalunya) classificades segons diversos paràmetres.

Aquests paràmetres/informació que conté el shp són:
TIPOLOGIA
OPERADOR
AMPLE_VIA
NUM_VIES
US_ ("Mercaderies", "Mixt", "Viatgers")
COBERT ("0" = Descobert, "1" = Cobert)
SUPRASERVE
ELECTRIFIC
DESCRIPCIO
Shape_Length
**A partir d'aquí camps d'informació nous, útils per representar individualment**
CORR_MED_ ("1" = Corredor Mediterrani)
TAV_ ("1" = Alta Velocitat)
RODALIES_ ("1" = Xarxa de Rodalies per viatgers)
FGC_ ("1" = Ferrocarrils de la Generalitat) Es pot filtrar per L'Ibregat-Anoia Barcelona-Vallès amb el camp SUPRASERVE
ACTUACIO_ ("Sense_Actuacio", "Nova_Infra" = Vies proposades pel PTMB que s'executaran els proxims anys, "En_Desús" = Vies que deixaran d'existir)
PLATJA_VIES_ ("1" = Pertany a una platja de vies).

Cal esmentar que és un shp. complet en quant a informació però simplificat en quant a línies perquè a l'hora de fer plànols a escala metropolitana es llegeixi millor la informació. Per això, a trams on diferents tipus de vies van paral·leles, s'ha simplificat a un únic traç i s'ha afegit tota la informació corresponent per la seva correcta representació (us...). Excepte amplex de via, us, operador, etc. que evidentment al fulsonar-los, s'ha seggat informació.
```

Figure 3. Documentació de la base de dades de la figura 2.

Això mètode de treball amb les dades genera certs problemes:

- Centralització:** Pràcticament només el creador d'aquesta part de l'estudi entendrà perfectament com és cada un d'aquests camps. La documentació és heterogènia i extensa. L'exemple mostrat és relativament senzill, però al fer-ho sovint i amb bases de dades més grans i complicades, provoca una dependència personal a l'autor de l'estudi.
- El mètode de treball és molt **seqüencial**, es van agafant dades i modificant fins a obtenir noves dades. Això fa que:
 - Sigui poc modular, cada cop que tinguem una nova dada tornarem a calcular coses que potser ja hem calculat abans amb altres dades.
 - Sigui difícil preveure els colls d'ampolla, i gestionar el temps proporcionalment a la importància dels processos. A vegades les coses més complicades amb una aproximació més naïf es podrien resoldre ja que pràcticament no afecten al resultat final.
- Poc rígid:** No hi ha un marc estricte de documentació, i aquesta és llarga i poc jeràrquica.

4. **Duplicació:** Provoca duplicació de dades originals. Si es vol modificar una base de dades per dos fins diferents, o bé se'n creen dues d'iguals (repetint dades) o bé s'utilitza la mateixa (moltes columnes que tenen poc a veure).
5. És **poc automatitzable**, ja que tots aquests processos són molt encarats a solucionar el problema d'un estudi en concret i no solucions generals que es puguin aplicar a altres llocs. Per tant dona lloc a automatitzacions massa concretes. Petits canvis a l'estudi costaran gran esforç al programador.
6. La **dependència del software** utilitzat és elevada, ja que totes aquestes noves columnes estan creades utilitzant ArcGIS. Això fa que a vegades hi hagin alternatives que no ofereix l'ArcGIS, però sí a més baix nivell python, per exemple, que no es tenen en compte.

Metodologia

En aquest apartat es proposen les solucions a les problemàtiques comentades a l'apartat context

Rigidesa de les dades

La solució proposada en aquest treball gira entorn la segona problemàtica de la complexitat de dades; es basa més en el control del flux que no en la rigidesa de les dades d'entrada.

Tot i això també es proposa una possible solució a aquesta primera problemàtica es proposa un preprocessament de les dades inicials (les donades per diferents agents) i la creació d'una base de dades PDU amb estructura lògica pròpia. Seguint un esquema com el de la següent imatge.

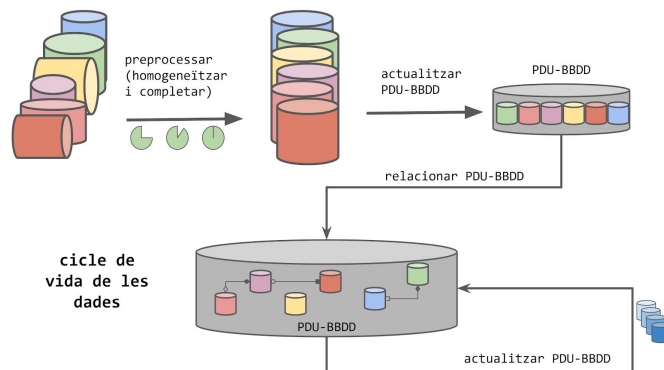


Figure 4. Esquema de la base centralitzada PDU

Aquesta base de dades hauria de contenir les relacions rígides entre diferents unitats a través d'un model entitat relació. Un esquema lògic que relacioni les diferents dades donades i cada cop que arribi una nova dada s'ha d'entendre com una part d'aquesta gran base de dades.

L'existència d'aquesta base de dades global permetria que ja no calgués preguntar-se com es relacionen les diferents unitats de treball geomètriques (parcel·la amb polígon de planejament) cada cop que es fa un estudi, sinó que s'unificaria el criteri en global per tot el PDU.

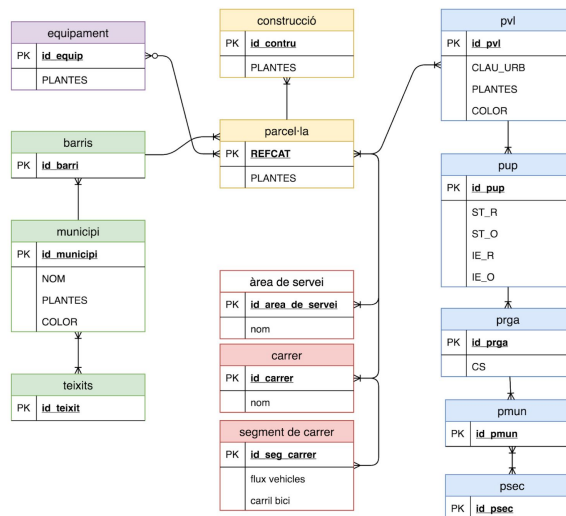


Figure 5. Exemple de model E-R amb les unitats d'estudi del PDU

Aquest enfocament és similar al vist en el mètode 'SUDS' a l'estat de l'art. Cal dir que aquesta solució requeriria molt d'esforç tècnic és ja que no és una tasca fàcil decidir aquests criteris unificadors de relació entre dades. En dos estudis es poden voler utilitzar dos criteris diferents i si tenim rigidesa absoluta tindriem un problema.

Control de flux

Pel que fa la segona problemàtica del control de flux és la temàtica principal d'aquest treball. La solució proposada es basa en crear un llenguatge comú per poder controlar el flux de dades i solucionar les dificultats presentades. Aquest llenguatge pretén canviar com es documenten els processos al PDU. Veiem les necessitats d'aquest llenguatge.

1. Ha de generar una documentació **fàcil de veure i comprendre**, per qualsevol persona. Ha de ser fàcil de crear, al ser una capa més d'informació a mantenir cal que sigui molt intuïtiva. Per tal que sigui fàcil de comprendre, cal rigidesa i que tothom ho faci de la mateixa manera. És una eina que pot anar sent modificada col·lectivament en funció de les necessitats.
2. Ha de reduir en parts més petites un gran estudi, de manera que aquests paquets siguin útils per altres estudis, ajudant així a la **modularització**, i això portarà a més, facilitat en l'automatització de processos. Ha d'ajudar també a la planificació d'un estudi, s'ha de poder crear de dalt a baix (planificació) i de baix a dalt (estudis ja fets).
3. Ha de treure importància a les dades derivades i centrar-se en el procés que crea aquestes dades. De manera que qualsevol estudi sigui fàcilment **replicable** amb dades diferents.
4. Ha de poder **jerarquitzar** la informació: poder incorporar diferents visualitzacions, dalt i baix nivell: els que executen l'estudi els hi fa falta

saber cada una de les funcions utilitzades per replicar-lo, però a l'hora d'explicar-ho no cal baixar a nivell tant baix.

5. Ha d'**unificar** la forma de documentar, establir uns standards de documentació.

La solució proposada es basa en la creació de diagrames de blocs. S'ha creat una eina amb la qual es poden crear diferents tipus de diagrames, amb diferents tipus de blocs que representen cada element del flux de les dades: funcions, bases de dades i estudis. Aquestes representacions es poden fer referència entre elles, i aquestes relació és guarden lògicament, de manera que desde cada un dels diagrames podem accedir als diagrames dels components que el formen. Estudis es fan servir en altres estudis i les bases de dades es reutilitzen. Es poden establir uns blocs standard a ser utilitzats per tal que el llenguatge sigui rígid.

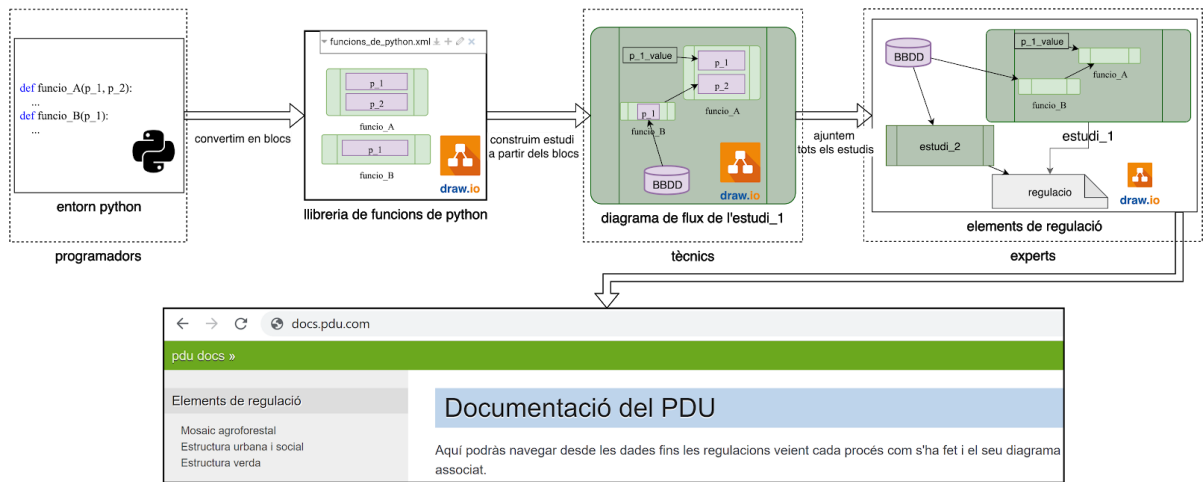
És una solució que per ser funcional ha de ser incorporada en el workflow dels tècnics. Els tècnics en la fase de planejament de l'estudi han de poder utilitzar els diagrames per comunicar les diferents etapes de l'estudi. Mentre fan l'estudi han de poder completar aquests diagrames amb el procés real que s'utilitza i finalment quan s'acabi l'estudi han de poder gaudir d'una documentació standard que expliqui el seu procés d'una forma senzilla i directa. Per tal d'analitzar millor els usos que volem d'aquesta eina creem el diagrama de casos d'ús.

Objectiu

Per tal de solucionar el problema del control de flux, l'objectiu principal d'aquest treball és crear una estructura d'informació que permeti la creació d'un vocabulari unificat, que representi els canvis que se'ls fa a les dades, a través d'un mateix canal: el dels diagrames.

És a dir no es pretén crear un vocabulari únic per fer referència a totes les funcions utilitzades o estudis, sinó l'eina amb el qual aquest vocabulari es pot anar creant i millorant.

D'aquesta manera els tècnics i experts prototipen les línies generals de l'estudi. Posteriorment els tècnics busquen les dades, fan uns primers esquemes per detectar els colls d'ampolla de l'estudi i contacten als programadors per aquelles funcionalitats més específiques que no es podran fer amb les funcions que tenen disponibles. Els programadors creen les noves funcions i aquestes funcions s'afegeixen a la llibreria col·lectiva de funcions. Els tècnics i els experts executen finalment l'estudi i en treuen conclusions. I tot aquest procés es documenta amb els diagrames, de manera que es té una única manera de documentar, i per tant en pot sortir una documentació "oficial" i unitaria de tots els estudis.



documentació

Figure 6. Flux de dades

Disseny

Diagrama de casos d'ús

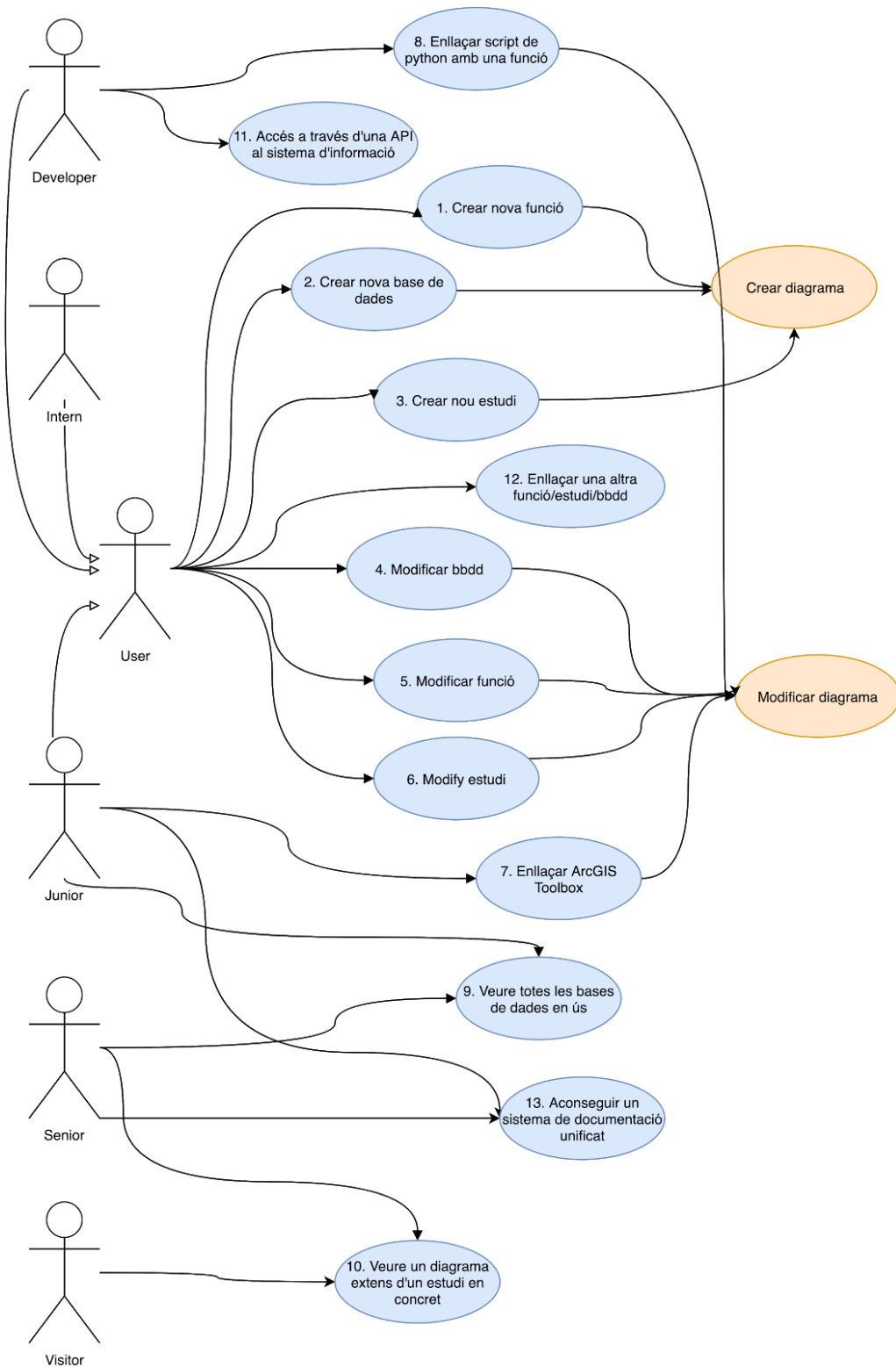


Figure 7. Diagrama de casos d'ús

Aquest diagrama de casos d'ús és més aviat encarat a les aplicacions d'aquesta estructura d'informació i no tant al seu disseny.

Descripció	1. Crear nova funció
Actors	User (Junior, Intern & Developer)
Precondicions	Programa obert
Flux bàsic	System: Ha d'oferir el diagrama standard d'una funció com a llibreria User: Arrossega el diagrama standard i modifica els continguts
Postcondicions	La funció creada estarà disponible com a bloc de diagrama a adjuntar

Descripció	2. Crear nova base de dades
Actors	User (Junior, Intern & Developer)
Precondicions	Programa obert
Flux bàsic	System: Ha d'oferir el diagrama standard d'una base de dades com a llibreria-drawio User: Arrossega el diagrama standard i modifica els continguts
Postcondicions	La base de dades creada estarà disponible com a bloc de diagrama a adjuntar

Descripció	3. Crear nou estudi
Actors	User (Junior, Intern & Developer)
Precondicions	Programa obert
Flux bàsic	System: Ha d'oferir el diagrama standard d'un estudi com a llibreria-drawio User: Arrossega el diagrama standard i modifica els continguts
Postcondicions	L'estudi creat estarà disponible com a bloc de diagrama a adjuntar

Descripció	4./5./6. Modificar diagrama
Actors	User (Senior, Junior, Intern & Developer)

Precondicions	Ha d'existir el diagrama.
Flux bàsic	User: obra el diagrama User: el modifica
Postcondicions	El bloc que representa aquella funció/bbdd/estudi és actualitzat seguint les noves modificacions

Descripció	7. Enllaçar una ArcGIS Toolbox amb una funció
Actors	Junior
Precondicions	Tenir una toolbox que executi la funció del diagrama
Flux bàsic	User: copia la ruta de la toolbox al diagrama funció

Descripció	8. Enllaçar un python script amb una funció
Actors	Developer
Precondicions	Tenir un script de python que executi la funció del diagrama
Flux bàsic	User: copia la ruta de l'script al diagrama funció

Descripció	9. Veure totes les bases de dades en ús
Actors	Senior, Junior
Precondicions	Haver estat creades
Flux bàsic	User: opció al menú

Descripció	10. Veure tots els diagrames subjugats a un estudi concret
Actors	Senior, Junior
Precondicions	Els diagrames d'un estudi han estat creats
Flux bàsic	User: es crea un nou diagrama buit User: es selecciona desde el programa la creació del diagrama complet (opció del menú) i es selecciona l'estudi que es vulgui.

Descripció	11. API del sistema d'informació
Actors	Developer
Precondicions	None

Flux bàsic	Developer: crear aplicacions sobre el sistema
------------	--

Descripció	12. Enllaçar diagrames
Actors	User (Junior, Intern & Developer)
Precondicions	Haver creat el diagrama de què depèn i estar modificant el dependent
Flux bàsic	User: arrossega el bloc minimitzat que representa el diagrama (present a la llibreria del sistema) al nou diagrama

Descripció	13. Aconseguir un sistema de documentació unificat
Actors	Senior, Junior
Precondicions	Tenir diagrames creats
Flux bàsic	User: obra una aplicació de documentació i pot navegar a través de tots els estudis

Planificació

Aquí es mostra la planificació per aquest projecte. Aquest projecte com a tal va començar el febrer de 2019 i s'ha acabat el juny del 2019, dedicant aproximadament 4 hores al dia. És cert que l'anàlisi del context s'havia fet prèviament, ja que al treballar automatitzant l'estudi del sostre potencial s'estava fent l'anàlisi del context i de la dinàmica de les dades.

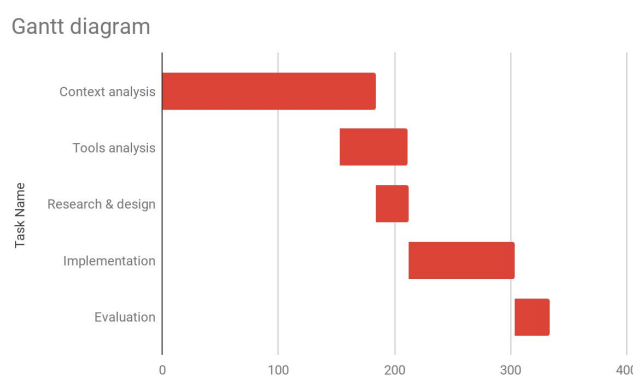


Figure 8. Diagrama de Gantt del projecte

Implementació

S'han implementat per aquest treball dos projectes:

- **flowio**: una aplicació d'escriptori que recull tot el que s'ha escrit en l'apartat de disseny. Pretén controlar el flux de dades a través de diagrames de draw.io. La versió portable de l'aplicació es troba a:
<https://sourceforge.net/projects/flowio/files/flowio/>
- **flowio-docs**: una aplicació d'escriptori que crea una documentació standard a partir dels diagrames creats a flowio. La versió portable de l'aplicació es troba a:
<https://sourceforge.net/projects/flowio/files/flowio-docs/>

Funcionament a nivell d'usuari

També hi ha una guia d'usuari a l'enllaç:

<https://bernatesquirol.github.io/flowio-desktop/>

Primers passos

Cal descarregar-se el programa desde aquest [enllaç](#) i executar-lo en un ordinador Windows.

El primer cop que s'obra el programa, cal decidir una carpeta on es guardaran tots els diagrames que formin part de l'entorn flowio. Per fer això cal anar a flow.io > Change diagrams folder i seleccionar la carpeta desitjada.

Interfície d'usuari

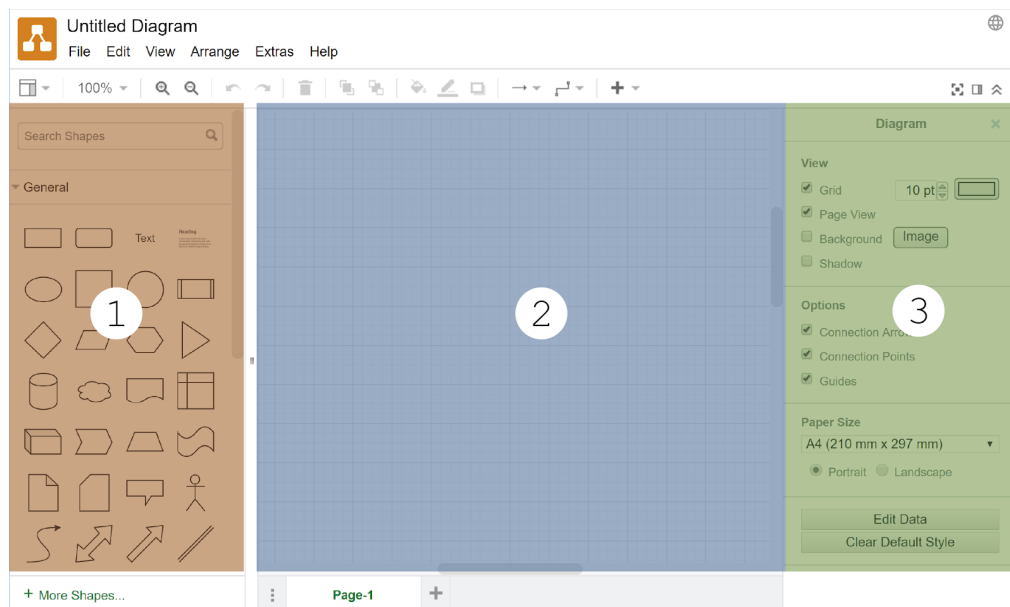


Figure 9. UI de drawio

La UI de drawio és senzilla, consta de 3 parts:

1. Llibreries de drawio: d'aquí apareixen les formes (blocs) útils per crear diagrames.
2. Espai de treball: Aquí és on es treballa, s'arrosseguen els blocs de 1 i es canvia el text i es connecten amb altres elements mitjançant fletxes.
3. Propietats de la forma: aquesta vista ens permet editar les propietats d'una forma seleccionada a 2

La UI de flowio és exactament igual però a 1 només hi trobem certes llibreries:

- flowio: aquesta llibreria conté els diagrames standard que explicarem a continuació.
- diagrames ja creats: cada diagrama guardat es converteix en un bloc que podem enllaçar als nous diagrames

Funcionament

L'usuari ha de crear diagrames que representin els estudis, bases de dades o funcions que s'han executat.

Totes les dades es generen a partir dels diagrames. Hi ha uns diagrames standard a la llibreria flowio creats com a base d'on crear diagrames.

Hi ha tres tipus de diagrames standard que podem fer:

- Bases de dades: representa el concepte de d'una base de dades original, donada per algú, i no en som responsables del manteniment.
- Funcions: representa el concepte de d'una funció, amb uns paràmetres d'entrada i uns de sortida. Podria ser una funció de Python o una funció de l'ArcGIS
- Estudis: agafen bases de dades i les connecten amb funcions per tal de crear noves dades derivades, que seran el resultat de l'estudi.

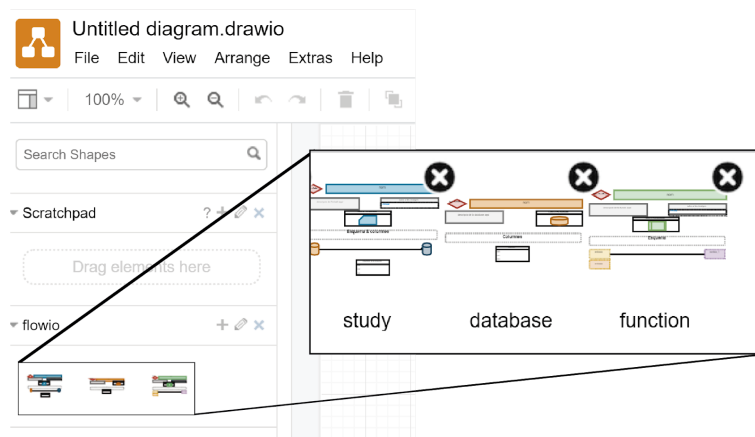


Figure 10. Diagrames standard de la llibreria flowio

Els tres tipus de diagrames tenen una sèrie de blocs que son purament informatius, on hi va un text: títol, descripció,... Tenen els tres també una caixa on hi posa 'Forma versió minificada'. Si es substitueix la forma de dins d'aquest requadre per qualsevol altra forma de les llibreries (estan amagades es pot fer click

a 'More shapes...'), la versió minificada d'aquella funció, base de dades o estudi. D'aquesta manera el vocabulari del llenguatge és molt modificable, una funció pot tenir un color o una forma que la distingeixi.

Els estudis i les funcions tenen una part esquemàtica, que és on s'explica mitjançant diagrames quin procés intern té aquell estudi o funció. Allà és on es pot fer referència a funcions, diagrames o bases de dades de la llibreria.

Les funcions només poden fer referència a altres funcions, ja que una funció és independent al valor dels seus paràmetres. Els estudis en canvi poden fer referència a bases de dades o altres estudis, d'on surt la informació, i funcions que la transformin.

Aplicacions

El fet de tenir tots els estudis interrelacionat físicament (els diagrames fan referència explícita als altres diagrames associats) permet crear aplicacions a sobre útils:

- Extensió d'un diagrama
Aquesta aplicació està incorporada a dins de flow.io i és per visualitzar tota aquesta informació d'una forma més compacta. El que es pot fer és crear un nou diagrama amb tota la informació d'una funció/estudi i la informació dels estudis, funcions o bases de dades que fa referència i així recursivament. Això ens deixa un mapa genèric d'un estudi en concret
Si es va a flow.io > Expand diagram
- Flowio-docs
Aquesta aplicació es tracta d'una aplicació externa de flowio. Flowio-docs crea una documentació basada en els diagrames amb totes les funcions i estudis que hem anat creant.

Funcionament intern

Hi han guies de desenvolupador de cada part als enllaços:

<https://bernatesquirol.github.io/flowio-core/>

<https://bernatesquirol.github.io/flowio/>

<https://bernatesquirol.github.io/flowio-docs/>

Tecnologies descartades

Degut a que el llenguatge de programació més comú a l'entorn és python (ArcGIS té una API per a python i quan s'han fet projectes a més baix nivell s'ha utilitzat geopandas amb python) al principi el mètode triat per l'implementació d'aquest projecte era python. Juntament amb l'entorn JupyterLab es va pensar que els diagrames es podrien implementar amb drawio desde l'entorn Jupyter, deixant així oberta la funcionalitat de poder executar aquests diagrames amb python.

Finalment al veure que l'entorn de Jupyter és complicat de programar i tot i haver-hi una extensió ja creada pel JupyterLab de draw.io, no era suficient ni amb una interfície gaire intuïtiva. Per tant es va prioritzar utilitzar alguna cosa més desenvolupada i més intuïtiva.

Tot i això es va crear el principi d'una api per python dels arxius de draw.io. Aquesta etapa va estar lligada al prototyping i al disseny de l'aplicació.

Tecnologies triades

Era clar que l'eina de fer diagrames era el draw.io, ja que una de les més obertes (open-source), amb més suport i més fàcil d'usar. És un projecte open-source liderat per Gaudenz Alder i David Benson basat en JGraph.

Aquesta eina és molt àmplia i desenvolupada desde fa molts anys, per tant té un nivell de complexitat al codi molt alt. També és cert que no hi ha una documentació extensa per tal d'entendre com funciona internament ja que tot i ser codi obert, no està creat amb la intenció que sigui modificat. Per evitar haver de descobrir com funciona tota l'aplicació web, no s'ha canviat directament la aplicació web del draw.io sinó només l'aplicació d'escriptori. Draw.io utilitza un framework anomenat electron.js per convertir la seva aplicació web en una aplicació d'escriptori: drawio-desktop. Aquest framework ens dona en entorn de Node.js entorn de programació de javascript que serveix per executar codi. Per tant es treballarà amb duplicats (forks) del codi de drawio, on només es tocaran els arxius que fan referència a la compilació per escriptori.

Funcionament intern

Aquest apartat és un resum del les guies per desenvolupadors que hi ha als repositoris de github. Veiem un esquema general del projecte:

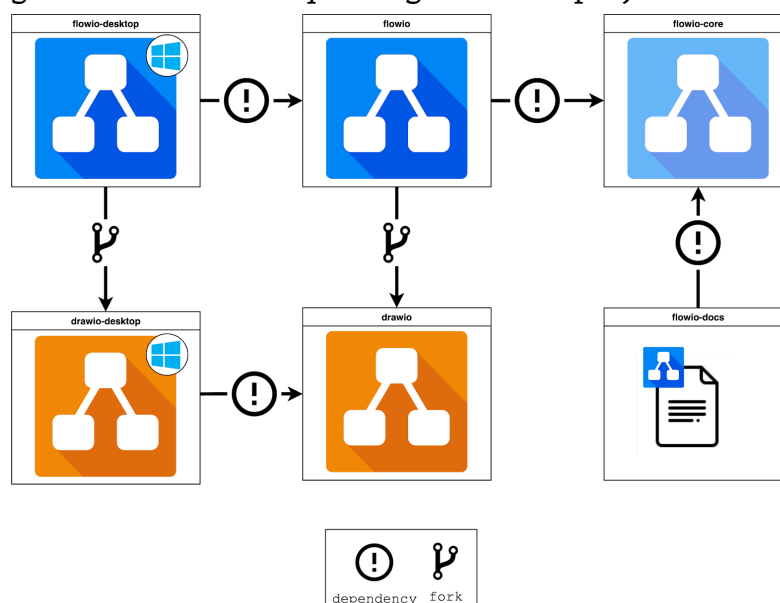


Figure 11. Estructura interna del programa

El codi és més aviat funcional que no orientat a objectes, ja que la tendència de javascript és cada cop tendir més cap a una programació funcional i reactiva. Flowio i flowio-docs depenen els dos d'electron.js, una llibreria de javascript per compilar aplicacions web per entorns d'Escriptori.

Repassem els principals arxius i les seves funcionalitats:

Flowio-core: On es troba el codi nou de tot el projecte

<https://github.com/bernatesquirol/flowio-core>

`drawio-node.js`: És bàsicament una API per els fitxers de drawio, tal com drawio els guarda. Aquesta API ens donarà facilitat al llegir i escriure diagrames i llibreries de drawio. Esta desenvolupada quasi desde zero.

`flowio.js`: És el principal punt d'accés al projecte. Utilitza les funcions de 'drawio-node.js' per fer tot el I/O. Bàsicament s'encarrega de implementar la lògica pròpia del flowio: base de dades - funció - estudi. Això ho fa confiant que l'usuari fa servir uns diagrames especials. Les figures d'aquests diagrames que interessin aniran marcades amb una clau 'flowio_key'.

Flowio: és un fork de drawio modificant alguns comportaments.

<https://github.com/bernatesquirol/flowio>

`electron.js`: Desde aquí es controla l'start-up de l'aplicació que és on quasi tot es canvia modifica. Fa el paper de servidor en una aplicació web, però amb la possibilitat d'executar codi al client.

1. L'aplicació draw.io permet canviar alguns dels seus paràmetres a través de la query 'GET' a la URL que es faci al iniciar l'aplicació, però drawio-desktop no ho permet. Amb flowio es poden canviar aquests paràmetres a través de l'arxiu 'urlParams.json' situat a '\$APPDATA\$/urlParams.json' (es pot trobar també fent click al menú *flow.io > Change diagram location*). Els paràmetres extres que incorpora flow.io (a més dels de draw.io):
 - 'flowio_path' [\triangle REQUIRED!!]: és la ruta on guardarem tots els diagrames, perquè funcioni l'aplicació és necessari omplir-la, per defecte és null.
 - 'clibs': són llibreries de draw.io que es volen importar cada cop que s'obra l'aplicació, es pot modificar amb un String o una array
2. electron.js controla a més l'estat de l'aplicació. Els esdeveniments que emet el 'BrowserWindow' (la finestra de navegador que fa servir electron per a mostrar la webapp) ens són útils per controlar el flux. En el cas de flowio ens interessa usar l'esdeveniment on 'dom-ready' del 'BrowserWindow.webcontents' per tal de carregar les llibreries, quan l'aplicació estigui preparada. Quan 'dom-ready' carreguem totes les llibreries que necessitem: s'envia un objecte 'args-obj' al client (ElectronApp.js) amb la llibreria preparada per ser carregada.

ElectronApp.js: Arxiu ja existent a drawio que actua de client en l'aplicació, reb missatges del servidor i administra els recursos de diagramly i mxgraph (dependències de drawio). S'encarrega per exemple d'obrir un diagrama de l'ordinador. S'ha extès aquestes funcionalitats incorporant que quan es rebí un missatge desde servidor amb 'args-obj' amb la clau 'llib' (local library) es carregui aquesta llibreria. Aquesta llibreria ha d'estar en el format standard de 'mxLibrary'. El valor de xml ha d'estar comprimit utilitzant la llibreria 'pako' tal com es fa a drawio.

```
<mxlibrary>
  [{"xml":"dVFBcoMWDHYN...", "title":"title"},...]
</mxlibrary>
```

Figure 12. Llibreria de flowio

Flowio-docs: Una de les aplicacions construïdes sobre flowio, basada en docsify, crea una documentació a partir dels diagrames creats

<https://github.com/bernatesquirol/flowio-docs>

main.js: És el punt d'entrada de l'aplicació flowio-docs, agafa els paràmetres de l'urlParams.json de flowio-desktop, i crea una vista utilitzant docsify.

flowio-docs.js: Converteix els diagrames en el format de docsify, tot creant les sidebars i els README's de cada carpeta. Es crea tot en una carpeta _docs allà on tenim tots els diagrames (flowio_path). Preserva l'estructura de carpetes de la carpeta on es guarda tot.

Flowio-desktop: és un fork del repositori de JGraph

Estructures de dades

Cada diagrama de drawio es guarda en format xml i comprimit de la següent manera:

```
<mxfile modified="2019-06-01T06:55:24.253Z">
  <diagram id="aI8DmJniFy8GYhdvDRHG" name="Page-1">
    VrLVuM4EP2aLOH4nWRJAnQv0j1MM3NmyVF...
  </diagram>
</mxfile>
```

Figure 13. Diagrama de drawio

Quan es descomprimeix el contingut té aquest format:

```
<mxGraphModel dx="171" ...>
  <root>
    <mxCell id="0"/>
    <mxCell id="1" parent="0"/>
    <mxCell id="kRyRl8HqRjgSbQ_-zm33-1" style="..." parent="1">
      <mxGeometry x="70" y="180" width="80" height="80" as="geometry"/>
    </mxCell>
    ...
  </root>
</mxGraphModel>
```

Figure 14. Diagrama descomprimit de drawio

Es poden veure les diferents formes que componen el diagrama i les relacions que tenen. Relacions parentals entre formes, impliquen que una forma conté l'altre. La clau per la intrusió a drawio desde flowio es basa en el fet que podem editar propietats d'aquestes formes. Podem fer que una forma (per exemple el que representa el títol d'un diagrama):

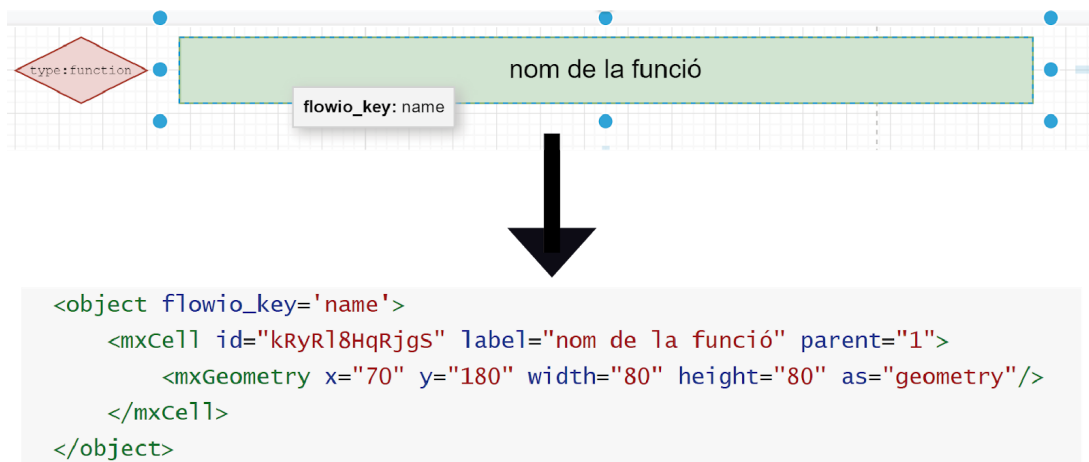


Figure 15. Bloc del diagrama amb clau `flowio_key`

D'aquesta manera, establint uns blocs amb unes `flowio_key` desde codi podrem accedir al nom de la funció i a tots els seus detalls. Així també és com s'estableixen els blocs de descripció, els que detecten si un diagrama és una funció, una base de dades o un estudi, o qualsevol dels que s'utilitzen per extreure informació del diagrama. Utilitzant la llibreria `xml2js` es tradueix en objectes javascript, amb `$` indicant les propietats i cadascuna de les altres claus fills del node. Per ajudar a tractar aquests objectes s'ha creat la classe `mXObject`, que reuneix algunes funcions comunes per tot node dels diagrames.

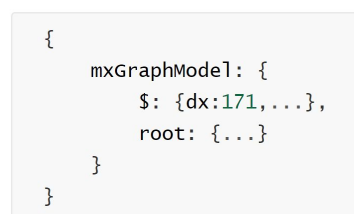


Figure 16. Conversió de l'`xml` en objecte javascript

Un altre punt important de l'estructura de dades és els fitxers. Tot el programa es basa en els diagrames guardats en una de les unitats de l'ordinador amb l'extensió `'drawio'`. Això genera un parell de conflictes: necessitem un identificador per cada fitxer, ja que els diagrames són les dades de flowio i es necessiten identificar. Primer es feia servir la propietat `ino` dels fitxers, que és un identificador únic per fitxer. Ràpidament es va veure que això era un error ja que no es pot moure l'arxiu entre unitats (canvia el seu `ino`). Si volguéssim compartir una sèrie de diagrames a un altre ordinador no podríem ja que es substituirien els identificadors per uns de nous. Es va optar per utilitzar l'identificador de cada diagrama. Això tampoc és una solució gaire més bona, ja que tot i que soluciona el problema anterior, hi ha dos comportaments que posen en perill la integritat del

sistema: quan es borren els blocs un diagrama ja guardat (desde l'aplicació draw.io) i es guarda amb un nom nou, la id del diagrama no canvia. Per tant es pot tenir dos diagrames amb la mateixa id. Quan importem els diagrames ens assegurem no importar-ne de repetits, però tot i això és un comportament problemàtic.

Exemple: Equipaments

Els equipaments són edificis públics que donen un servei: CAPs, biblioteques, centres cívics, escoles bressol. L'annex 8 de l'avanç del PDU (Equipaments 2019) tracta d'analitzar la infradotació o superdotació dels equipaments de l'àrea. S'ha seleccionat aquest estudi per provar l'eina ja que és un exemple d'estudi que combina dades heterogènies, funcions de ArcGIS i funcions de python. Tot i que aquesta eina està feta per ser modificada mentre es crea l'estudi, també podem documentar a posteriori.

Ens fixarem en el cas dels Centres d'Atenció Primària, però veurem que els estudis dels altres tipus d'equipaments són de fet utilitzant les mateixes funcions. Aquest estudi consisteix en creuar les dades de població per parcel·la (un altre estudi fet pel PDU) amb les dades de capacitat per equipament, obtenint al final una relació entre equipaments i parcel·les que correspon a quin equipament aniria una parcel·la en concret. Així es pot concloure quines parcel·les no estan ben ateses i també quins CAPs estan sobrecarregats. Cal seguir desde les dades originals fins a la relació entre equipaments i parcel·les.



Figure 17. A l'esquerra l'estructura de fitxers, a la dreta els blocs de les bdd

Abans de començar s'ha de decidir l'estructura de carpetes amb què es vol estructurar els diagrames: aquest cas com que només analitzem un estudi es fa segons tipus de diagrama i no segons temàtica, però això és decisió de l'usuari.

Seguidament es comencen a crear els esquemes corresponents a les bases de dades inicials, en aquest cas creem el de les dades de població per secció censal (que provenen de l'IDESCAT) i el dels equipaments que ja té la capacitat associada (prové de l'IERMB).

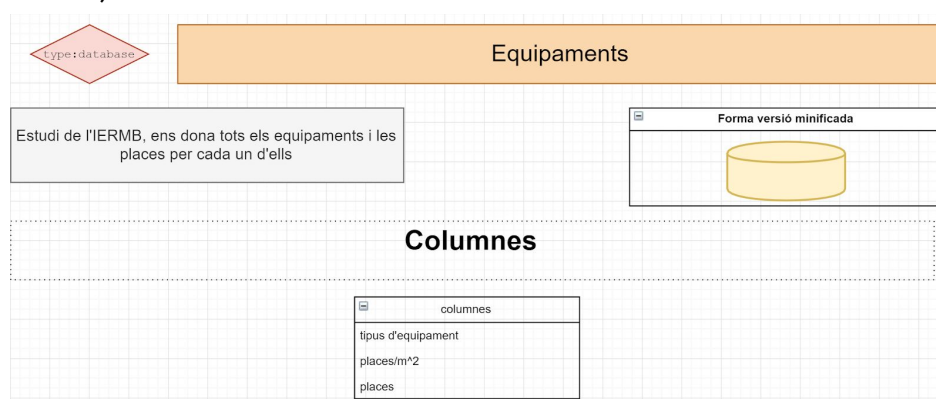


Figure 18. Creem un diagrama a partir l'standard base de dades

Es veu que ja apareixen com a figures a la llibreria bdd (nom de la carpeta). El PDU va fer un estudi per passar aquesta població de l'IDESCAT a població per parcel·la. Això es va fer utilitzant, la base de dades de les seccions censals i un altre estudi, el dels teixits residencials. Cal crear l'estudi de teixits residencials i el de població per parcel·la. Aquí, per exemple, no explicitem quin procediment hi ha dins d'aquest càlcul. Si bé s'hauria d'especificar per tenir una documentació exhaustiva, no cal per mostrar aquest exemple.

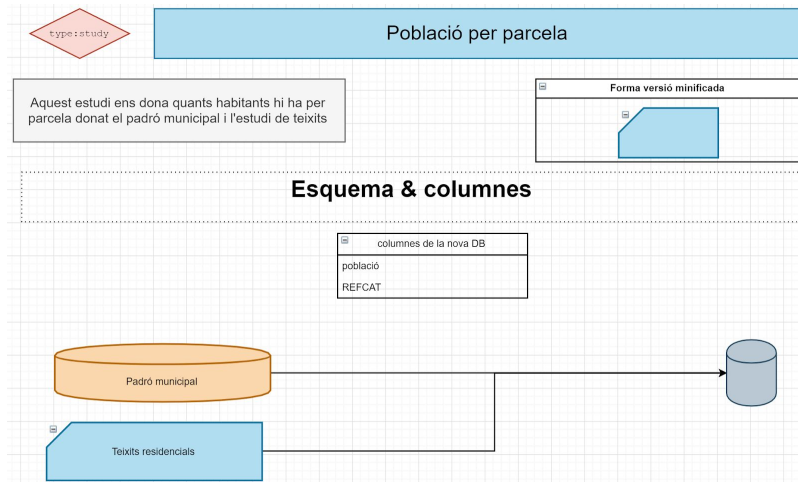


Figure 19. Diagrama de l'estudi Població per parcel·la

Per tal de relacionar estudis i parcel·les s'utilitza una funció de ArcGIS que s'anomena MatriuOD, és la primera funció que s'ha de crear. Aquesta funció genera una matriu de costos entre uns orígens i uns destins. Els costos són el temps que es tarda d'anar de l'origen al destí. Per tant la MatriuOD té tres camps: la clau de l'origen, la clau del destí i el temps que es tarda caminant (utilitzant un carrer).

Ara es tenen tots els ingredients per tal de distribuir les parcel·les als equipaments. Això es fa amb un algorisme de Python i està trencat en diverses funcions. Bàsicament el que fa aquest algorisme és mantenir una llista ordenada de parcel·les més properes per cada equipament i una llista d'equipaments més propers a cada parcel·la (funció: trobar preferits). Si una parcel·la P1 té com a equipament més proper un E1 tal que té la parcel·la és entre les seves més properes llavors es fa el match (funció: trobar parelles). Si no, s'espera a la següent ronda per veure si entra. És un algorisme similar a les carreres i estudiants de la selectivitat.

S'ha d'explicar aquest algorisme amb blocs, cosa que no és fàcil. Per fer-ho ens cal el concepte de loop, ja que al passar d'una ronda a la següent es torna a executar la funció que distribueix. El que es fa és crear la funció que s'executa a cada ronda: filtra els equipaments i parcel·les ja aparellats, calcula els tops de cada parcel·la i de cada equipament, i fa el matching. Si el matching no ha canviat vol dir que hem acabat. Si ha canviat vol dir que podem fer una altra ronda.

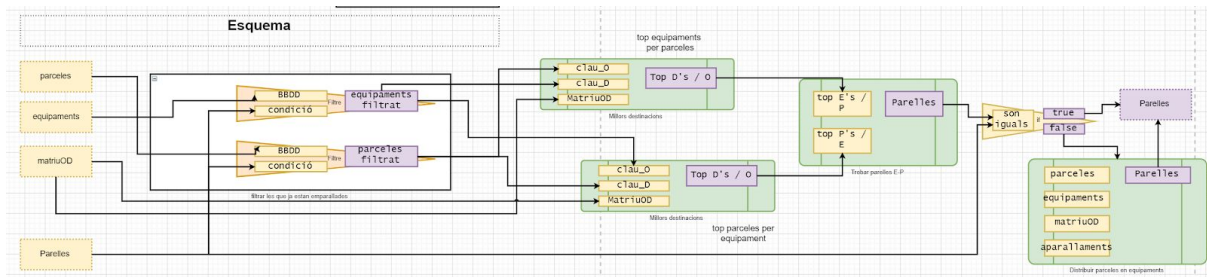


Figure 20. Esquema del funcionament de la distribució de les parcel·les per equipaments

Cal dir que l'algorisme és una mica més complex, ja que també és té una llista de parcel·les que no es pot fer match ja perquè no tenen cap equipament disponible. Aquesta llista també s'inclou a la condició d'aturada però no a l'esquema per intentar-ho fer més digerible.

Finalment tenim l'estudi dels Centres d'Atenció Primària (CAP). Agafa les els equipaments selecciona els del tipus CAP, calcula la MatriuOD d'aquestes i totes les parcel·les de l'àrea i les distribueix.

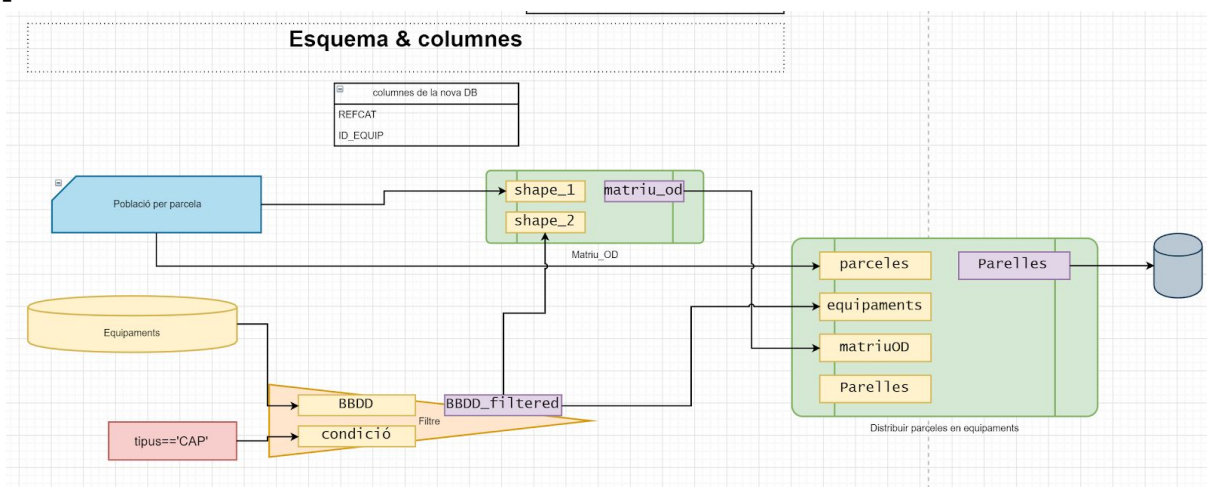


Figure 21. Estudi CAPs

Així ja s'ha completat els diagrames referents a aquest estudi. Per crear l'estudi de les biblioteques o dels centres de dia, l'esquema seria el mateix. Les funcions serien les mateixes que ja s'han creat (no caldria crear-les de nou) i simplement creant un nou estudi amb un diagrama similar a aquest, canviant la condició de tipus=='CAP' a tipus=='biblioteca'.

Ara es pot veure què generen les aplicacions creades sobre flowio:

1. Extensió de l'esquema de l'estudi: crea automàticament un diagrama global de tots els estudis i funcions dins d'un cert estudi. El nivell 1 tenim els blocs minificats de 3 funcions: MatriuOD, filtre i distribució i d'un estudi: població per parcel·la. Nivell 2 els esquemes d'aquests 4 estesos i el nivell 3 els de la funció distribució estesos.

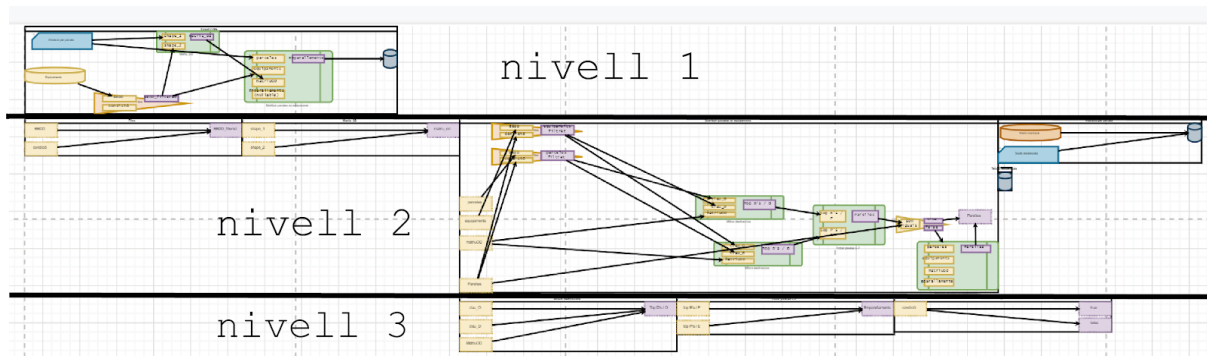


Figure 22. Diagrama expandit de l'estudi dels CAPs

És evident que no és la millor manera d'ensenyar tota aquesta informació ja que el diagrama és molt extens i descompensat, però és una proof-of-concept del que es podria arribar a ser si es té aquesta estructura creada.

2. Flowio-docs: la documentació que s'ha generat automàticament posa per cada carpeta quins estudis hi ha a dins, els detalls d'aquest estudi o funció, i també una visualització interactiva dels esquemes de cada una de les funcions i dels estudis.

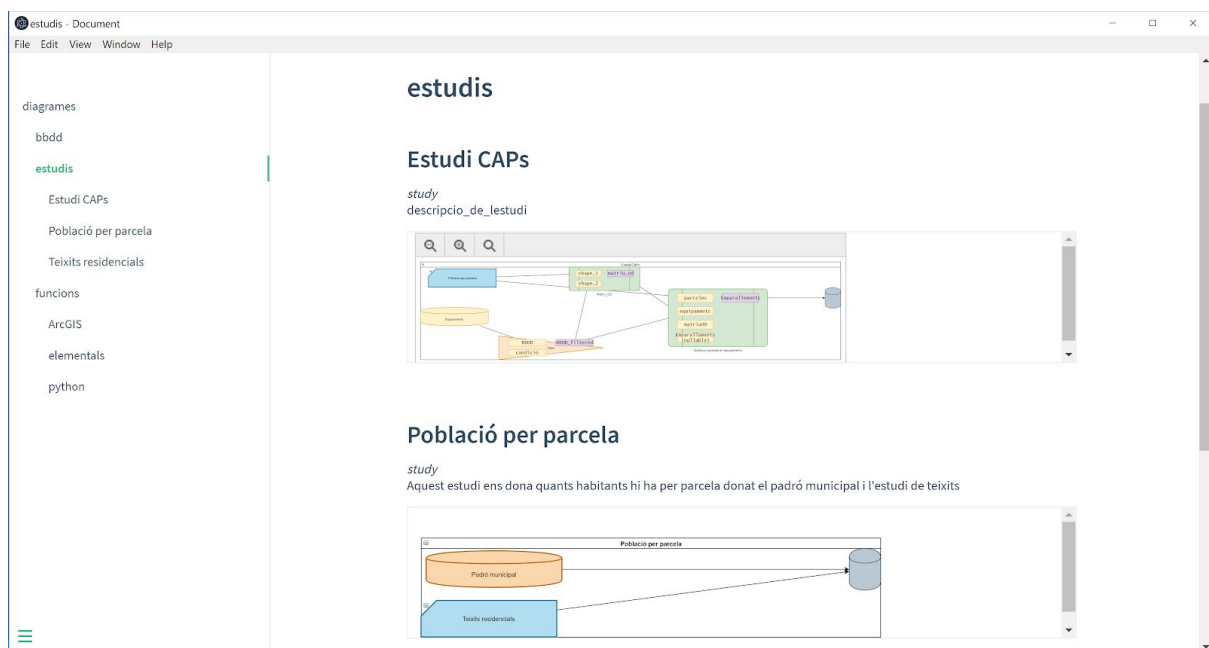


Figure 23. Vista de flowio-docs dels diagrames creats

Conclusió

Un dels objectius de l'administració és entendre el funcionament de la societat per poder oferir serveis més acurats i que persegueixin l'interès públic. . Les dades són una base fonamental per entendre com s'estructura una comunitat i quines són les seves necessitats. Però les dades són heterogènies, provenen de diferents agents amb diferents perspectives i no són exhaustives, per tant sovint s'han de combinar per obtenir coneixement real. Això genera una gran complexitat.

Al Pla Director Urbanístic de l'AMB s'han de fer estudis amb moltes dades diverses. Per tal de dominar aquesta complexitat cal tenir mecanismes de control. Molta recerca es basa en com crear un mecanisme global de dades públiques.

Aquest treball intenta crear una eina (flowio) per unificar la documentació d'estudis generada per un equip de treball per tal de tendir a un enfocament més modular i organitzat de la feina. La proposta d'aquest treball rau en la creació de diagrames que representin el procés que passen les dades d'origen (base de dades donada) a final (indicador).

Aquesta eina s'incorpora a la feina de tècnics, experts i desenvolupadors i crea una estructura d'informació que s'hi pot accedir mitjançant aplicacions programades sobre aquest esquema. Una de les aplicacions més útils és la de crear una documentació unitaria i oficial de tots els estudis d'un grup de treball.

És cert que segueix sent una capa d'informació mantinguda i per tant una de les continuacions d'aquest treball podria ser l'execució d'aquests diagrames, fent així tot part del mateix cicle plantejament-execució-documentació.

Índex de figures

- [Figure 1. Creació d'estudis amb dades heterogènies](#)
- [Figure 2. Aspecte de base de dades del PDU](#)
- [Figure 3. Documentació de la base de dades de la figura 2.](#)
- [Figure 4. Esquema de la base centralitzada PDU](#)
- [Figure 5. Exemple de model E-R amb les unitats d'estudi del PDU](#)
- [Figure 6. Flux de dades](#)
- [Figure 7. Diagrama de casos d'ús](#)
- [Figure 8. Diagrama de Gantt del projecte](#)
- [Figure 9. UI de drawio](#)
- [Figure 10. Diagrames standard de la llibreria flowio](#)
- [Figure 11. Estructura interna del programa](#)
- [Figure 12. Llibreria de flowio](#)
- [Figure 13. Diagrama de drawio](#)
- [Figure 14. Diagrama descomprimit de drawio](#)
- [Figure 15. Bloc del diagrama amb clau flowio_key](#)
- [Figure 16. Conversió de l'xml en objecte javascript](#)
- [Figure 17. A l'esquerra l'estructura de fitxers, a la dreta els blocs de les bdd](#)
- [Figure 18. Creem un diagrama a partir l'standard base de dades](#)
- [Figure 19. Diagrama de l'estudi Població per parcela](#)
- [Figure 20. Esquema del funcionament de la distribució de les parcel·les per equipaments](#)
- [Figure 21. Estudi CAPs](#)
- [Figure 22. Diagrama expandit de l'estudi dels CAPs](#)
- [Figure 23. Vista de flowio-docs dels diagrames creats](#)

Bibliografia

- Anejionu, Obinna C. D., Piyushimita (vonu) Thakuriah, Andrew McHugh, Yeran Sun, David McArthur, Phil Mason, and Rod Walpole. 2019. "Spatial Urban Data System: A Cloud-Enabled Big Data Infrastructure for Social and Economic Urban Analytics." *Future Generations Computer Systems: FGCS 98* (September): 456–73. <https://doi.org/10.1016/j.future.2019.03.052>.
- Davies, Tim. 2010. "Open Data, Democracy and Public Sector Reform." *A Look at Open Government Data Use from Data. Gov. UK*. <http://www.opendataimpacts.net/report/wp-content/uploads/2010/08/How-is-open-government-data-being-used-in-practice.pdf>.
- "Decidim Whitepaper :: Decidim Docs." n.d. Accessed June 15, 2019. <https://docs.decidim.org/whitepaper/en/decidim-a-brief-overview/>.
- Einav, Liran, and Jonathan Levin. 2014. "The Data Revolution and Economic Analysis." *Innovation Policy and the Economy* 14 (January): 1–24. <https://doi.org/10.1086/674019>.
- Equipaments. 2019. "Annex 8: Estudi D'equipaments de l'AMB." A8: Equipaments. March 2019. https://docs.amb.cat/alfresco/api/-default-/public/alfresco/versions/1/nodes/of4fd24c-e2cb-4033-96af-azeod818c127/content/A8_EquipamentsAMB_Avan%C3%A7%20PDU_6_Annexos.pdf?attachment=false&mimeType=application/pdf&sizeInBytes=14846962.
- Jetzek, Thorhildur, Michel Avital, and Niels Bjorn-Andersen. 2014. "Data-Driven Innovation through Open Government Data." *Journal of Theoretical and Applied Electronic Commerce Research* 9 (2): 100–120. <https://doi.org/10.4067/So718-18762014000200008>.
- Khan, Zaheer, Ashiq Anjum, Kamran Soomro, and Muhammad Atif Tahir. 2015. "Towards Cloud Based Big Data Analytics for Smart Future Cities." *Journal of Cloud Computing* 4 (1): 2. <https://doi.org/10.1186/s13677-015-0026-8>.
- Laurini, Robert. 2014. *Information Systems for Urban Planning: A Hypermedia Cooperative Approach*. CRC Press. <https://www.taylorfrancis.com/books/9781482268355>.
- "Manifesto in Favour of Technological Sovereignty and Digital Rights for Cities :: Ethical Digital Standards." n.d. Accessed June 17, 2019. <https://www.barcelona.cat/digitalstandards/manifesto/o.2/>.
- Psyllidis, Achilleas, Alessandro Bozzon, Stefano Bocconi, and Christiaan Titos Bolivar. 2015. "A Platform for Urban Analytics and Semantic Data Integration in City Planning." In *Computer-Aided Architectural Design Futures. The Next City - New Technologies and the Future of the Built Environment*, 21–36. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-47386-3_2.
- "The Promise of Data-Driven Policymaking | Issues in Science and Technology." 1970. *Issues in Science and Technology*. January 1, 1970. <https://issues.org/esty-2/>.
- Wilson, Luc, Jason Danforth, Carlos Cerezo Davila, and Dee Harvey. n.d. "How to Generate a Thousand Master Plans: A Framework for Computational Urban Design." https://static1.squarespace.com/static/568ee2b4cbced6419694f1fc/t/5c78636b1905f46a4b109aa3/1551393648032/63_How+to+Generate+a+Thousand+Master+Plans-A+Framework.pdf.