
This is the **published version** of the bachelor thesis:

Castellanos Sancho, Josep; Freire Bastidas, Diego Mauricio, dir. Front-end aplicació per comparar algorismes d'encaminament en xarxes oportunistes. 2021. (958 Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/264153>

under the terms of the  license

Front-end aplicació per comparar algorismes d'encaminament en xarxes oportunistes

Josep Castellanos Sancho

Resum– En les xarxes oportunistes els nodes estan en constant moviment. Per aquesta raó, per poder comunicar-se aprofiten les oportunitats de connexió. A més, s'utilitzen diferents algorismes per a l'encaminament de paquets, algorismes que cal avaluar mitjançant simuladors. Aquests simuladors configuren les simulacions a través de fitxers de text, per aquesta raó, en aquest treball s'ofereix un front-end per a la definició dels escenaris i l'avaluació dels resultats. D'aquesta forma, ja no és necessari conèixer prèviament el funcionament dels simuladors. En altres paraules, la incorporació d'aquest front-end facilitarà l'estudi de les xarxes oportunistes, estalviant temps i incorporant altres funcionalitats.

Paraules clau– Xarxes oportunistes, Encaminament, The One simulador, Front-end, Vue.js.

Abstract– In opportunistic networks nodes are in constant movement, so in order to communicate they take advantage of connection opportunities. In addition, different algorithms are used for packet routing, algorithms that must be evaluated through simulators. These simulators simulations are configured through text files. For this reason, in this work it is offered a front-end for the definition of the scenarios and the evaluation of the results. This way, it is no longer necessary to know previously how the simulators work. In other words, the incorporation of this front-end will facilitate the study of opportunistic networks, saving time and incorporating other functionalities.

Keywords– Opportunistic networks, Routing, The One simulator, Front-end, Vue.js.



per aquesta raó s'usen simuladors que permeten provar l'efectivitat d'aquests algorismes en diferents entorns.

1 INTRODUCCIÓ

A diferència d'Internet, que es basa en la commutació de paquets, les xarxes oportunistes es basen en la commutació de missatges que s'emmagatzemen en els nodes, fins que s'entreguen al node final, aprofitant les oportunitats de connexió. D'aquesta forma es pot mantenir la comunicació en entorns on no es pot assegurar la connexió continua [1].

Per aconseguir aquesta commutació de paquets, les xarxes oportunistes usen protocols d'encaminament, que generalment es basen en decidir quan emmagatzemar, transportar i encaminar (store-carry-and-forward) els paquets [2]. L'estudi d'aquests protocols es complex,

En aquest projecte s'utilitza el simulador The One. El problema és que, encara que disposi d'una interfície per visualitzar les simulacions, la configuració i posterior visualització dels resultats, es fa a través de documents de text. El que significa que per poder generar i interpretar aquests documents, s'ha d'entendre com estan estructurats.

Aquest document seguirà la següent estructura: en primer lloc, es contextualitzarà el treball en l'apartat Estat de l'art, començant amb una explicació sobre les xarxes oportunistes, un segon subapartat presentant els Protocols d'encaminament i una introducció al front-end per The One. En segon lloc, es troba l'apartat corresponent als Objectius, la Metodologia i el Pla de projecte, on es veurà el Cronograma i la Planificació. Llavors es passarà a explicar el Desenvolupament del projecte amb els subapartats d'Arquitectura, Diagrama de flux i el Tipus de simulacions (Simulacions parametritzades i Simulacions bulk). Una vegada donades totes

- E-mail de contacte: 1526535@uab.cat
- Menció realitzada: Tecnologies de la Informació
- Treball tutoritzat per: Diego Mauricio Freire Bastidas (DEIC)
- Curs 2021/22

les explicacions prèvies es veuran els Resultats del projecte, repassant cadascuna de les parts front-end desenvolupat, amb el seu corresponent apartat. Finalment, les Conclusions del projecte, el Treball futur, els Agraïments i les Referències bibliogràfiques.

2 ESTAT DE L'ART

Actualment, les xarxes oportunistes encara són terme d'estudi, per la qual cosa cal avaluar mitjançant simulació els diferents mecanismes amb els quals funcionen, principalment, els protocols d'encaminament.

En aquest apartat es començarà contextualitzant que són les xarxes oportunistes, a continuació es veuran les principals filosofies per l'encaminament de missatges, quins simuladors existeixen i per acabar, l'estat actual de front-ends pel simulador The One.

2.1 Xarxes oportunistes

Les xarxes oportunistes ofereixen una solució per la comunicació en entorns on no es pot mantenir una infraestructura fixa [3]. En aquest tipus d'entorns els nodes no disposen d'una organització predeterminada dels enllaços, i, generalment, estan en constant moviment [4]. El que proposen les xarxes oportunistes és aprofitar les oportunitats de connexió pel reencaminament, és a dir, dinàmicament, els nodes poden emmagatzemar els missatges mentre es desplacen i reenvien el missatge a altres nodes fins a arribar al seu destí.

Per reencaminar els missatges, les xarxes oportunistes utilitzen diferents algorismes que s'encarreguen de prendre decisions relatives a aquest reencaminament, és a dir, de quina forma s'aprofiten les oportunitats de connexió. Per exemple, quan un node haurà de reencaminar un missatge, quant temps l'haurà d'emmagatzemar, etc.

2.2 Protocols d'encaminament

Encara que hi ha un gran ventall de possibilitats a l'hora de dissenyar algorismes encarregats de l'encaminament en xarxes oportunistes [5], principalment tots es basen en algun dels dos esquemes següents:

- **Inundació:** es basa en la capacitat dels nodes per emmagatzemar i retransmetre els missatges, generant d'aquesta forma duplicats dels missatges, cosa que provoca un alt consum de recursos de xarxa.
- **Reexpedició:** només es manté una còpia de cada missatge, que es va encaminant fins al seu destí.

2.3 Simuladors de xarxes oportunistes

Les xarxes oportunistes encara estan en desenvolupament i constant millora, per tant, encara és necessari crear i provar els diferents protocols d'encaminament. És a dir, les xarxes oportunistes encara són terme d'estudi. Per aquesta raó, i per l'alta complexitat, és necessari simular per provar l'efectivitat dels protocols d'encaminament en diferents

situacions controlades.

Simular xarxes oportunistes és una tasca complexa, per aquesta raó existeixen diferents simuladors per l'estudi i anàlisi de les xarxes oportunistes. Les principals són ns-3, OmNet++ i The One, encara que per raons que es donaran més endavant, The One ha estat l'elecció final per aquest projecte.

La primera, ns-3 [6], és un simulador d'esdeveniments discrets de sistemes d'Internet, que és capaç de simular xarxes oportunistes. Alternativament, OMNet++ [7], també és un simulador d'esdeveniments discrets que, a través d'un framework, pot simular models de moviment de les xarxes oportunistes.

Però, l'opció escollida ha estat The One [8], ja que ofereix una solució específica per xarxes oportunistes que és capaç d'incorporar models de mobilitat externa i interna, diferents protocols d'encaminament, a més de postprocessament en el resultat de les simulacions.

2.4 Front-end per The One

The One pot executar simulacions de dues maneres diferents, a través d'interfície gràfica o sense. El problema és que aquesta interfície gràfica només permet visualitzar la simulació en qüestió, en cap cas permet veure els resultats ni configurar les simulacions, el que significa que són necessaris coneixements previs en l'estructura i funcionament del simulador. A més, l'execució de simulacions està limitada, ja que si es vol executar múltiples simulacions amb diferents configuracions, és necessari, encara més temps i domini del simulador The One.

Si bé, és veritat que es pot utilitzar i configurar l'entorn de simulació de The One a través d'un IDE com Eclipse, per exemple, actualment no existeix una capa entre el simulador The One i el sistema d'arxius que fa servir per a les configuracions. Com es veu a la Figura 1, per estudiar les xarxes oportunistes (nivell més alt) s'empren simuladors i per configurar aquests simuladors, de forma tradicional, es fan servir arxius de text. Entre aquestes dues últimes capes, s'ha afegit una nova capa que no existia prèviament i que permet a l'usuari no haver d'entendre a tan baix nivell el funcionament del simulador.

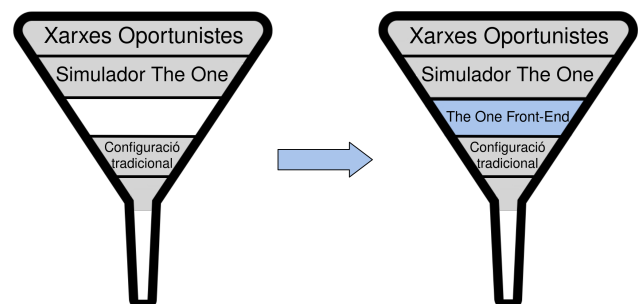


Figura 1: Capa front-end entre el simulador i la configuració tradicional.

3 OBJECTIUS

L'objectiu principal del projecte és afegir una capa al simulador The One, un front-end, que permeti comparar els diferents algorismes d'encaminament en xarxes oportunistes. Això és pel fet que la forma tradicional a través de documents de text és limitada i requereix coneixements previs en l'estructura d'aquests documents.

Per aconseguir aquest objectiu principal es plantegen els següents objectius funcionals:

- Desenvolupar un front-end que sigui capaç d'executar simulacions bàsiques pel simulador The One.
- Dissenyar un formulari al front-end que permeti a l'usuari configurar diferents paràmetres de simulació.
- Presentar els resultats de les simulacions a l'usuari de forma gràfica.
- Permetre a l'usuari exportar els arxius per simular de forma tradicional.

De cara a futur, com que el disseny del projecte és sostenible i escalable, es plantegen els següents objectius de disseny:

- Adaptar el front-end per permetre el seu allotjament en un servidor.
- Emmagatzemar els arxius de configuració i de resultats en una base de dades.
- Oferir el front-end com a servei web a través d'Internet.

Durant la realització del projecte es va plantejar un nou objectiu principal, l'opció d'executar simulacions bulk, cosa que es tractarà en l'apartat de planificació, amb la corresponent justificació a l'apartat de desenvolupament. En resum, les simulacions bulk consisteixen a executar simulacions provant paràmetres específics per veure quin impacte tenen en els resultats.

4 MOTIVACIÓ

El problema de The One és l'enujós sistema amb el qual funciona, a través de documents de text, per al que es requereix coneixement previ en l'estructura i en el funcionament d'aquest mecanisme [9]. Amb el desenvolupament d'aquest front-end es permetrà centralitzar els esforços únicament en l'estudi de les xarxes oportunistes, estalviant temps en la preparació dels entorns de les simulacions i fent més amigable la comparació de les simulacions executades, de forma visual amb gràfiques.

Finalment, una de les problemàtiques a l'hora d'executar múltiples simulacions és la falta de recursos, cosa que es soluciona amb la mateixa naturalesa del projecte. Al tractar-se d'un front-end escalable, amb l'opció d'allotjar-se en un servidor es podrà optimitzar el procés de simulació, traslladant l'execució i consum dels recursos de l'usuari a un servidor.

5 PLA DE PROJECTE

5.1 Metodologia

La metodologia que s'ha seguit al llarg del projecte és una metodologia àgil. S'ha dividit el projecte en diferents fases que estan definides segons les entregues parcials marcades al calendari del Treball de Fi de Grau. Els esprints són de curta duració i tenen uns objectius molt definits. D'aquesta manera s'aconsegueix flexibilitat i seguretat, ja que en tot moment es coneix de forma detallada que s'ha fet del projecte i que es pot continuar fent. A més, en tractar-se d'una metodologia àgil, no hi ha hagut cap problema en eliminar o afegir esprints en funció de les necessitats del projecte.

Bàsicament, el seguiment s'ha realitzat a través d'un document en llenguatge markdown que es troba a l'arrel del projecte en github, on s'ha redactat una llista amb totes les tasques a completar. Cada vegada que es completava una tasca (esprint) es ratllava de la llista, s'apuntava la data en la qual s'acomplia la tasca i s'escrivía un resum de com es va fer. D'aquesta forma s'obté un resum ordenat cronològicament de com s'ha completat tot el projecte.

5.2 Planificació

El pla inicial divideix en tres fases el projecte, la fase inicial, el sistema de simulació i el sistema de resultats. El pla de projecte complert està disponible en l'apèndix Subsecció A.1, tot i que, com s'està treballant amb una metodologia àgil, és de caràcter orientatiu, ja que possiblement els esprints canviïn o es desglossin en altres més petits per a una millor organització, o per si les prioritats canvien.

Finalment, el pla de projecte només ha sofert un canvi important, la incorporació d'una nova forma de generar les simulacions, les simulacions bulk (aquesta filosofia s'explicarà més endavant). Tot i això, no va suposar un gran canvi en la planificació, gràcies a la reactivitat de la metodologia àgil i l'aprofitament del temps guanyat en altres tasques.

5.3 Cronograma

Per veure de forma més clara, ordenadament en el temps se seguirà el cronograma disponible en l'apèndix Subsecció A.2. Tot i que, finalment, només va patir una modificació, com s'ha comentat al pla de projecte, ja que, per cobrir les possibles necessitats d'estudi de les simulacions, es va afegir durant el desenvolupament una nova filosofia per executar les simulacions.

Es considera que s'ha seguit el cronograma satisfactòriament, no ha quedat cap tasca pendent i s'han complert en el temps previst. Fins i tot s'ha pogut reaccionar als canvis apareguts. Es considera responsable, en gran part, de l'èxit del projecte a la metodologia àgil escollida en la planificació.

6 DESENVOLUPAMENT

El desenvolupament del front-end s'ha dut a terme a través de l'editor de codi Atom [10], en llenguatge Javascript. Tot i això, el projecte no s'ha desenvolupat purament en Javascript. En la Figura 2 es pot veure com s'ha combinat l'ús dels frameworks: vue.js, frappe.js, node.js, express.js i bash script.

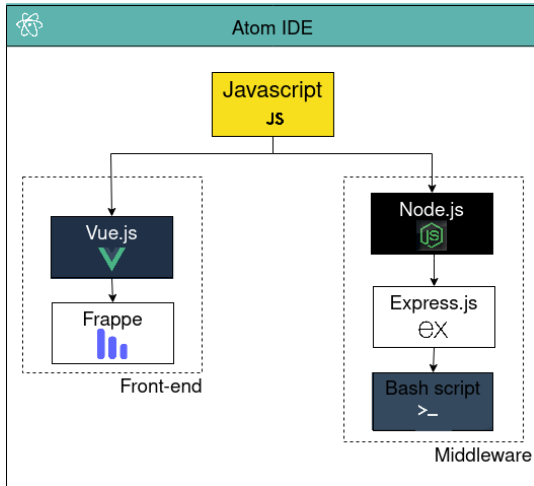


Figura 2: Entorn utilitzat en el projecte.

En primer lloc, pel desenvolupament del front-end, tot allò que veu l'usuari i amb el que pot interactuar, s'ha desenvolupat en vue.js [11], un framework de codi obert que permet crear interfícies en una sola pàgina. La raó pel qual s'ha escollit vue.js és perquè l'estudiant ja estava prèviament familiaritzat amb el framework. D'igual manera, en tractar-se d'un framework basat en la creació d'interfícies en una sola pàgina, permet executar les simulacions sense necessitat de recarregar l'app web.

Adicionalment, per la visualització dels resultats interactiva i dinàmicament, s'ha escollit el component de vue.js, frappe [12], el qual permet crear gràfiques dinàmiques, a més d'adaptar-se sense gaire dificultats a la forma en la qual es gestionen les dades en el projecte.

En segon lloc, pel middle-ware s'ha utilitzat express.js, una infraestructura per aplicacions web en node.js [13], que, a la vegada, farà de traductor entre el simulador The One i el front-end desenvolupat. Secundàriament, també s'ha fet servir bash script per enviar l'ordre d'executar les simulacions amb les configuracions seleccionades per l'usuari.

6.1 Arquitectura

L'arquitectura completa del front-end consta de dues parts principals: la mateixa part desenvolupada en el projecte, el front-end, corresponent a la part blava de la Figura 3 i el nucli, el simulador The One, que correspon a la part negra de la Figura 3. Així mateix, el front-end consta de dues seccions. En primer lloc, es troba la part més superficial que es presenta a l'usuari i amb la qual interactua directa-

ment, la interfície d'usuari.

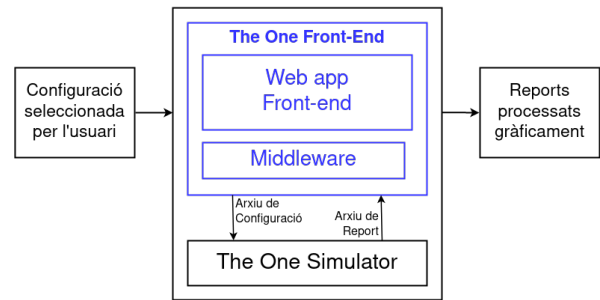


Figura 3: Arquitectura del sistema front-end desenvolupat.

També és la capa encarregada d'emmagatzemar temporalment la configuració, abans d'enviar-se a la següent capa. A l'inrevés, també rep els resultats de les simulacions per processar-los en gràfiques que es presenten a l'usuari a través del front-end.

En segon lloc, encara en la part superficial, es troba el middle-ware. S'encarrega de mantenir la infraestructura web i, bidireccionalment, de la comunicació entre el simulador i el front-end. O sigui, per una banda, tradueix els paràmetres seleccionats a través de la interfície en arxius de configuració perquè el simulador The One pugui entendre i executar, com l'usuari indica, totes les simulacions. I, per altra banda, també tradueix els reports que The One genera de les simulacions executades, de forma que el front-end pugui processar aquestes dades en les gràfiques que es mostraran a l'usuari.

Gràcies a la capa middle-ware, el projecte s'ha pogut desenvolupar sense modificar el simulador The One. Aquesta decisió s'ha pres de cara a futur. Per exemple, si es volgués migrar la web app a un servidor, aquest procés de migració no dependria de canvis importants en el simulador, gràcies al disseny modular de l'arquitectura.

En resum, l'usuari només interactua amb la capa més externa, la web app, per configurar i analitzar gràficament els resultats. Això significa que, per començar a estudiar els diferents algorismes d'encaminament de xarxes oportunistes, l'usuari no necessita entendre ni l'estructura dels arxius de configuració ni l'estructura dels reports.

6.2 Diagrama de flux

Inicialment, seguint la Figura 4, el flux de l'aplicació comença quan un usuari entra a l'aplicació web. L'usuari tindrà dues opcions per executar les simulacions: de forma parametrizada o bulk (al següent apartat s'explicarà en què consisteixen aquest tipus de simulacions).

Una vegada s'hagi escollit la configuració desitjada, en enviar l'ordre d'executar la simulació, l'usuari tindrà tres opcions a manera de confirmació. En primer lloc, només descarregar la configuració que s'acaba d'escollir, sense executar la simulació; en segon lloc, executar la simulació, i per últim, cancel·lar-la per continuar configurant la

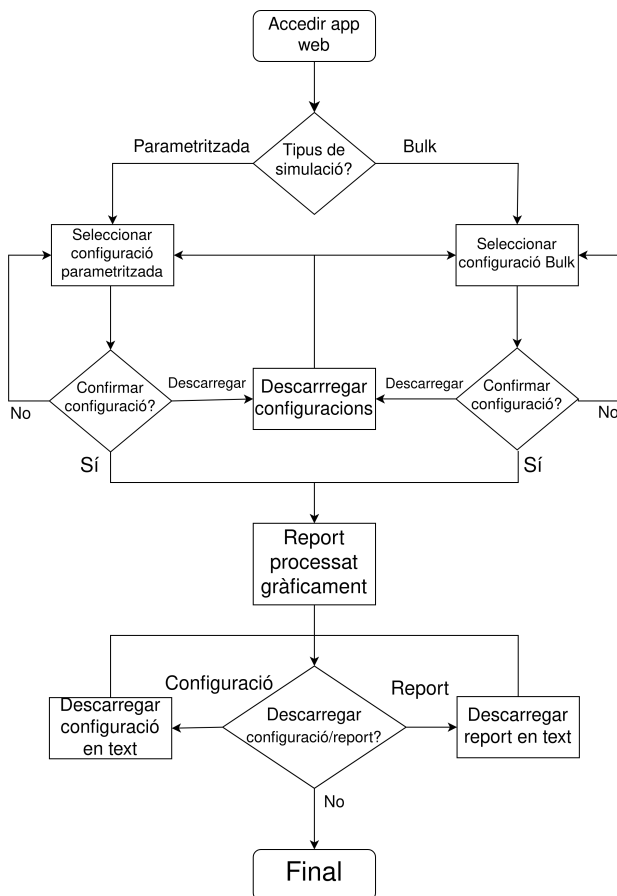


Figura 4: Diagrama de flux del front-end.

simulació.

Llavors les simulacions s'executaran i, quan s'hagin acabat d'executar, es mostraran a l'usuari de forma gràfica. En aquest punt, l'usuari pot descarregar l'arxiu de configuració, o de report, que desitgi. Si no desitja descarregar cap document, el flux de l'aplicació hauria acabat.

6.3 Tipus de simulacions

La forma d'executar les simulacions pot canviar segons les necessitats de l'usuari. Amb l'objectiu de cobrir aquesta necessitat, s'han dissenyat dues filosofies a l'hora de generar i executar simulacions, les simulacions parametritzades i les simulacions bulk.

6.3.1 Simulacions parametritzades

Es tracta del model més personalitzat a l'hora d'executar simulacions. En essència ofereix una solució per als usuaris que volen executar un nombre determinat de simulacions amb unes configuracions específiques.

Per tant, si s'escull executar les simulacions parametritzades, es podrà escollir el nombre exacte de simulacions a executar i personalitzar els paràmetres de cadascuna de les simulacions seleccionades.

6.3.2 Simulacions bulk

Alternativament, aquest model es presenta pels usuaris que desitgen executar un gran nombre de simulacions per comparar la influència d'alguns dels paràmetres de les xarxes oportunistes en l'eficàcia dels algorismes d'encaminament.

L'usuari selecciona quins paràmetres vol provar i el sistema de generació de configuracions genera totes les combinacions possibles amb aquests paràmetres seleccionats per, posteriorment executar-les. D'aquesta forma es permet estudiar com afecten certs paràmetres en les simulacions.

7 RESULTATS

Una vegada contextualitzat el projecte en l'anterior apartat de desenvolupament, es pot passar a presentar els resultats del desenvolupament realitzat. Per tindre una idea general del projecte desenvolupat, es presenta a l'apèndix Subsecció A.3 un mapa de la web desenvolupada.

Els resultats obtinguts, essencialment la web app, s'expliquen per pàgina. Aquestes pàgines són: la pàgina d'inici, la configuració de simulacions parametritzades i de simulacions bulk, el menú de confirmació i els resultats de simulació. Encara que abans es presentarà el middle-ware, ja que totes aquestes pàgines interaccionen d'alguna forma o altra amb el middle-ware.

7.1 Middle-ware

Com s'ha anat comentant al llarg de l'article, el middle-ware és la capa intermèdia, encarregada de la comunicació entre el nucli del simulador i la interfície. Encara que no només fa aquesta funció, també fa d'infraestructura web del front-end.

La infraestructura web, a més d'aixecar la web app, també s'encarrega de la funció d'encaminament. És a dir, que quan l'usuari interactua amb la interfície, sigui reencaminat a la pàgina correcta si la seva acció implica un canvi de pàgina. Aquest encaminament es pot veure a al mapa de l'apèndix Subsecció A.3, encara que, només el primer i segon nivell de l'arbre s'han considerat un canvi de context. Això significa que només es recarregarà la web app en aquests casos.

Per altra banda, per explicar la comunicació de la qual s'encarrega el middle-ware, a continuació es presenten els tres casos en els quals el middle-ware tradueix la comunicació entre el front-end i el The One:

- **Executar simulació.**

Quan s'executa una simulació s'envien les dades seleccionades a través del formulari al The One, per executar les simulacions amb la corresponent configuració. Per aconseguir-ho, s'ha creat una estructura de dades al front-end, que emmagatzema les dades del formulari de forma reactiva. Una vegada l'usuari decideix executar una simulació, les dades es remeten al middle-ware en format JSON. Llavors el middle-ware desempaqueta el missatge JSON i escriu un arxiu de configuració per cadascuna de les

simulacions a executar, traduint l'estructura de dades a l'estructura de l'arxiu que el simulador accepta. Per acabar, a partir d'un script en bash, s'executa el The One per cadascuna de les configuracions creades. Resumidament, aquest és el procés quan s'executa una simulació, encara que pel cas de les simulacions bulk s'afegeixen un parell de passes, ja que no és tan senzill crear els arxius de configuració.

- **Rebre resultats de simulació.**

Una vegada s'acaben d'executar les simulacions, el middle-ware ha de fer la traducció inversa a l'anterior. És a dir, convertir l'arxiu dels resultats de simulació en una estructura de dades, que com anteriorment, es formatarà en un missatge JSON que el front-end desempaquetarà i processarà en una gràfica.

- **Descarregar arxiu de configuració/simulació.**

Aquest darrer cas, és similar al primer però amb menys passes, ja que, com no s'executarà cap simulació, el middle-ware només necessita fer la corresponent traducció i enviar-la al front per descarregar.

A l'hora de la implementació del middle-ware, s'han presentat dos principals problemes, fer la traducció d'estructura de dades a arxiu en format The One, i la sincronització de processos. Ja que, en el primer cas, dependent de la configuració seleccionada, l'estructura de l'arxiu de configuració canviava, i per això s'havien de considerar aquests casos excepcionals. Pel cas invers, traduir els resultats generats per The One a estructura de dades era més senzill, pel fet que el processament de les gràfiques es realitza al front-end.

Per altra banda, es va complicar la sincronització de processos a causa del funcionament de Javascript i la pila de processos. Finalment, es va solucionar executant el The One de forma asíncrona, a través de time outs, un mètode de Javascript, que permet executar codi de forma temporitzada per poder controlar ordenadament la execució i posterior lectura dels resultats.

7.2 Pàgina principal

Es tracta de la primera pàgina amb la qual es troba l'usuari quan accedeix a la web app. En la Figura 5, es pot comprovar que consisteix en un menú per seleccionar quina de les dues modalitats de simulació es vol executar. En qualsevol cas, quan l'usuari esculli qualsevol de les dues opcions, serà redirigit a la pàgina de configuració de la corresponent modalitat.

7.3 Configuració de simulacions parametritzades

Es tracta de la primera modalitat desenvolupada, en la qual es poden seleccionar fins a 8 simulacions configurables a través d'un formulari. Per exemple, en la Figura 6, només hi hauria una simulació seleccionada. Això permet executar simulacions de forma controlada i específica.

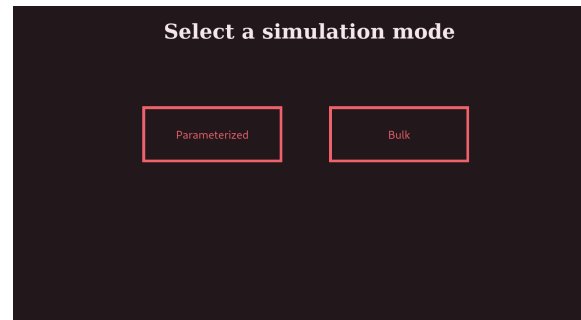


Figura 5: Pàgina principal per seleccionar la modalitat de simulació.

En total, s'han utilitzat tres components: el component pare, el component de configuració, que es crida fins a 8 vegades, una per configuració seleccionada i el component encarregat dels resultats de simulació. D'aquesta manera es facilita l'escalabilitat per la implementació en servidor.

En cas que es volgués augmentar el nombre de simulacions a executar, amb un parell de modificacions al codi, es podrien afegir de forma senzilla. Gràcies al disseny per components que permet el framework vue.js.

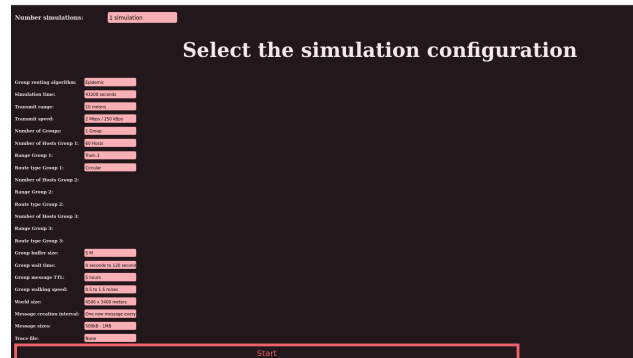


Figura 6: Pàgina de configuració de simulacions parametritzades.

Per executar les simulacions, es prem el botó "Start" de la part inferior de la Figura 6, i llavors s'obre el menú de confirmació.

Els principals problemes han estat dissenyar el formulari per tal que segons les opcions escollides, es canviïn els valors dependents en el formulari, ja que alguns valors només accepten certes configuracions. Cosa que s'ha aconseguit gràcies al mètode v-if (que renderitza o no condicionalment) de vue.js, i un mètode creat que s'encarrega de canviar el renderitzat de cada opció.

En últim lloc, la gestió de l'estructura de dades també ha estat un repte. Aquesta estructura de dades consisteix en un array d'objectes amb cadascuna de les opcions seleccionades com a propietat amb el corresponent valor. D'aquesta forma es pot operar fàcilment amb les dades seleccionades.

7.4 Configuració de simulacions bulk

Les simulacions bulk, de forma similar a la simulació parametritzada, consten d'un formulari però, en aquest cas, en format taula, com es veu a la Figura 7, en la qual se seleccionen quins paràmetres es volen provar a través de checkbox. Cada opció té tres possibles valors i depenent de les caselles marcades, es construiran totes les possibles combinacions.



Figura 7: Pàgina de configuració de simulacions bulk.

En canviar la modalitat de simulació, no ha sigut possible reutilitzar l'estructura de dades dissenyada per les simulacions parametritzades, ja que es complicava moltíssim la traducció i creació de les configuracions, perquè, com s'explicarà a continuació, és necessari que aquesta estructura de dades sigui dissenyada per poder fer backtracking, pel fet que es farà combinatòria per obtenir totes les configuracions amb les opcions marcades.

Aquesta estructura de dades està dividida en dos: un diccionari i un array d'objectes que emmagatzemarà les opcions escollides per l'usuari. Per una banda, s'ha creat un objecte que s'utilitza a manera de diccionari per obtenir les opcions i els seus valors possibles, a partir de la qual es genera la taula del formulari, cosa que augmenta l'escalabilitat, ja que si es vol afegir opcions, només caldrà afegir l'opció al diccionari, seguint el format.

Per altra banda, s'ha creat la mateixa estructura de dades a partir de la qual es farà el backtracking per la combinatòria. Es tracta d'un array que emmagatzema objectes de forma ordenada, on cada objecte correspon a una propietat i el valor seleccionat. És a dir, cada checkbox marcat tindrà el seu corresponent objecte dins d'aquest array.

El backtracking, el qual es realitza a la banda del middle-ware, és necessari per realitzar la combinatòria. S'encarregarà de recórrer l'array ordenat amb cada valor seleccionat i generar totes les combinacions d'aquests valors, ja que només pot haver-hi un valor per opció en cada combinació.

En la Figura 8 es pot veure, amb un exemple, com es portaria a cap la combinatòria. Si un usuari selecciona en la taula dos valors de temps de simulació, dos valors per rang de transmissió i l'algorisme d'encaminament epidèmic, es generaran un total de quatre combinacions. Recorrent els arbres com es mostra a la Figura 8, s'aconsegueix que cada combinació només tingui un valor per opció, com es

buscava inicialment.

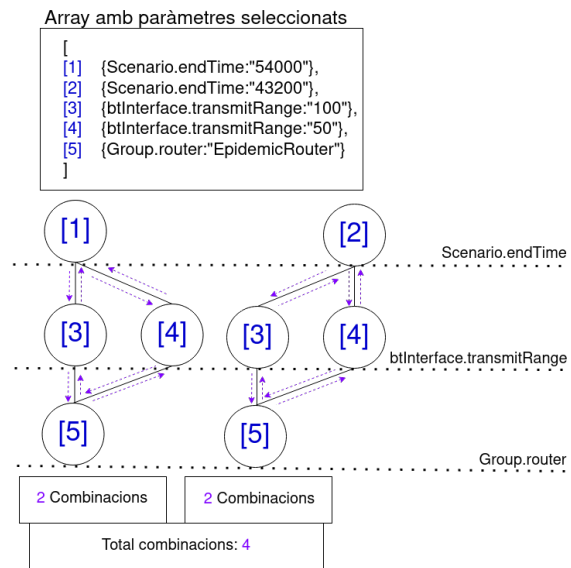


Figura 8: Funcionament del backtracking.

Un cop generades totes les combinacions i de forma similar al cas de les parametritzades, s'envien a executar però amb menys passes, ja que l'estructura de dades és més propera a l'estructura final que accepta el The One. Tot i això, la programació en el middleware no va ser senzilla, pel fet que l'addició de la funció backtracking va complicar la sincronització dels processos, però es va poder gestionar a través de funcions asíncrones i callbacks.

El principal problema en aquest tipus de simulacions ha estat gestionar la gran quantitat de combinacions que es poden arribar a generar i la falta de recursos per executar-les. Per aquesta raó, s'ha limitat el nombre de combinacions a la banda del front i del middle-ware. Per una banda, al front hi ha un límit de valors a escollir, i per l'altra, al middle-ware, se selecciona una mostra de 5 configuracions, les quals s'enviaran per executar. Aquestes limitacions es poden treure fàcilment i es facilitaran a l'usuari les passes per desactivar-les, si l'usuari ho desitja.

7.5 Menú de confirmació

Una vegada escollida la configuració desitjada, apareix un menú de confirmació com el de la Figura 9. En aquest menú de confirmació es pot pitjar el botó de "Cancel" si encara es volen fer canvis en la configuració, pitjar el botó de "Download config files" per només descarregar les configuracions, o polsar el botó de "Start simulations" per continuar amb l'execució de les simulacions.

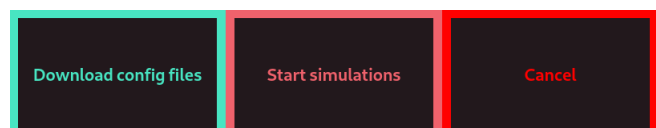


Figura 9: Menú de confirmació de simulació.

7.6 Resultats de simulació

Finalment, els resultats de la simulació es poden visualitzar de forma gràfica com es mostra en la part superior de la Figura 10, i, a la vegada, en la part inferior de la mateixa figura, també es pot descarregar la configuració o report de cadascuna de les simulacions executades.

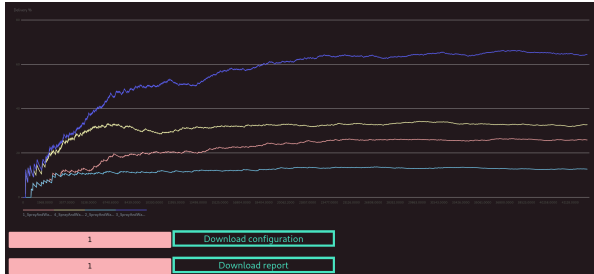


Figura 10: Pàgina de resultats de simulació.

Els principals problemes en aquesta part han estat la construcció de gràfiques, ja que la majoria de frameworks de vue.js per construir gràfiques no eren compatibles amb aquest projecte i s'hauria d'haver canviat tota l'estructura. Però, finalment, amb el framework utilitzat i frappe.js, consultant el codi del framework, es va poder importar de forma manual.

Pel que fa a la descàrrega, era més senzill, només s'havia de consultar localment al document desitjat, ja que una vegada executades les simulacions, ja estan generats tots els documents necessaris.

8 CONCLUSIONS

Durant la realització del projecte, s'ha pogut complir el principal objectiu del treball. S'ha pogut agregar una capa intermèdia entre el simulador i la configuració tradicional de les simulacions, a través de documents de text.

S'ha pogut comprovar que desenvolupar un front-end que es comuniqui amb un programa, en aquest cas el simulador The One, no és una feina trivial; cal implementar un middle-ware que transmeti tota la informació seleccionada a través del front-end fins al middle-ware, The One. A més, tota aquesta informació ha de ser formatada correctament, per tal que totes les capes siguin capaces de processar la informació, ja sigui perquè es mostri a l'usuari o que el middle-ware pugui processar-la.

Per altra banda, tot i haver-hi canvis durant el desenvolupament, com l'addició d'una nova forma d'executar les simulacions, les simulacions bulk, la cronologia s'ha pogut seguir sense gaire problema desplaçant la tasca del processament dels resultats. Això ha estat permès gràcies a la metodologia àgil que es va establir en les fases prèvies del projecte.

Finalment, durant les fases finals del projecte, van sorgir bastants problemes per trobar un framework capaç de generar gràfiques dinàmicament i que fos compatible amb el projecte. Afortunadament, es va poder exportar

el framework frappe, de forma que fos compatible amb l'estructura del projecte.

Personalment, aquest projecte ha sigut bastant profitós tant en l'àmbit tècnic com en l'àmbit transversal. És a dir, a escala tècnica he millorat moltíssim les meves aptituds en el desenvolupament front-end a través del framework vue.js. Com es comunica un front-end amb un middle-ware, en aquest cas a través d'express.js. A més, transversalment, s'ha millorat l'adaptació als canvis imprevistos i l'administració del temps.

9 FUTUR TREBALL

Una vegada s'ha realitzat el front-end del simulador per les simulacions executades en bach, és a dir, sense interfície gràfica, el següent pas és desenvolupar un front-end integrat en el simulador executat amb interfície gràfica. És a dir, una eina unificada que permeti la configuració i execució de les simulacions amb la interfície gràfica en temps real i, posteriorment, la possibilitat de visualitzar, en la mateixa eina, els resultats formatats com en el front-end desenvolupat.

Pel que fa a la funcionalitat, es podria allotjar el sistema de fitxers en una base de dades per aprofitar avantatges com alliberar el sistema local de l'emmagatzemament dels fitxers i disposar d'un historial consultable amb totes les simulacions realitzades. També es podrien afegir altres opcions per augmentar la configuració de les simulacions, encara que gràcies al sistema implementat, l'addició d'altres paràmetres és força senzilla. Finalment, es podria utilitzar aquest sistema en altres simuladors, només caldria fer la traducció adequada en el middle-ware.

AGRAÏMENTS

Primerament, agrair al tutor Diego Mauricio Freire Bastidas, pel suport ofert durant el projecte. Sense la seva alta disponibilitat i predisposició per revisar la feina, el treball final de grau hauria estat més feixuc, a més, els seus consells i punts de vista han estat molt profitosos pel desenvolupament del projecte. Finalment, també agrair a la família i amics pel seu suport constant, que tot i oferir un altre tipus d'ajuda, també ha estat molt important.

REFERÈNCIES

- [1] F. Warthman *et al.*, "Delay-and disruption-tolerant networks (dtns)," *A Tutorial. V. 0, Interplanetary Internet Special Interest Group*, pp. 5–9, 2012.
- [2] C. Glacet, M. Fiore, and M. Gramaglia, "Temporal connectivity of vehicular networks: The power of store-carry-and-forward," in *2015 IEEE Vehicular Networking Conference (VNC)*. IEEE, 2015, pp. 52–59.
- [3] R. Ramanathan and J. Redi, "A brief overview of ad hoc networks: challenges and directions," *IEEE communications Magazine*, vol. 40, no. 5, pp. 20–22, 2002.

- [4] C. Boldrini, K. Lee, M. Önen, J. Ott, and E. Pagani, "Opportunistic networks." *Comput. Commun.*, vol. 48, no. 14, pp. 1–4, 2014.
- [5] M. Liu, Y. Yang, and Z. Qin, "A survey of routing protocols and simulations in delay-tolerant networks," in *International Conference on Wireless Algorithms, Systems, and Applications*. Springer, 2011, pp. 243–253.
- [6] S. Rampfl, "Network simulation and its limitations," in *Proceeding zum seminar future internet (FI), Innovative Internet Technologien und Mobilkommunikation (IITM) und autonomous communication networks (ACN)*, vol. 57. Citeseer, 2013.
- [7] A. Petz, J. Enderle, and C. Julien, "A framework for evaluating dtn mobility models," in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, 2009, pp. 1–8.
- [8] A. Keränen, T. Kärkkäinen, and J. Ott, "Simulating mobility and dtms with the one," *J. Commun.*, vol. 5, no. 2, pp. 92–105, 2010.
- [9] X. Liu and Y. Chen, "Report of a dtn simulator-the one," 2013.
- [10] Atom documentation. [En línia]. Disponible: <https://atom.io/docs>. [Últim accés: 1 de Abril de 2022].
- [11] Vue.js documentation. [En línia]. Disponible: <https://vuejs.org/guide/introduction.html>. [Últim accés: 26 de Abril de 2022].
- [12] Frappe documentation. [En línia]. Disponible: <https://frappeframework.com/docs/v13/user/en/introduction>. [Últim accés: 13 de Mayo de 2022].
- [13] Node.js documentation. [En línia]. Disponible: <https://nodejs.org/es/docs/>. [Últim accés: 25 de Abril de 2022].

APÈNDIX

A.1 Planificació

El projecte va començar amb una reunió per entendre els requisits funcionals del projecte, per tal de poder determinar els objectius d'aquest i poder escollir l'entorn de desenvolupament adequat a les necessitats i coneixements de l'estudiant.

Pel lliurament de l'informe inicial, durant la setmana quatre, els esforços es van centrar en la redacció de l'informe i en una implementació bàsica del front-end.

El front-end comptava amb un formulari de selecció de paràmetres de simulació (que s'emmagatzemaven en una estructura de dades al front-end). A més, era capaç d'executar una simulació bàsica, per després mostrar el report resultant a l'usuari, sense cap format, en cru.

Durant les següents setmanes, es va desenvolupar, i millorar, la comunicació entre el front-end y el middle-ware. Primerament, es va formatar la comunicació entre el front-end i el middle-ware en JSON, per tal de poder crear les configuracions i visualitzar els reports.

En segon lloc, es va afegir la possibilitat d'executar més d'una simulació per petició, adaptant el formulari i optimitzant tant el procés de generació de configuracions com el d'execució d'aquestes per fer-ho possible.

Un cop s'apropava l'entrega del primer Informe de progrés (Setmana 9), per tal de complir el Cronograma previst, es va començar a enllestir l'execució de simulacions parametritzades, ja que, un nou requisit imprevist havia entrat en joc: les simulacions per bulk.

Per tant, fins a la Setmana 9 es van afegir les opcions que quedaven al formulari de paràmetres, la selecció de grups i els paquets de traces. Encara que no estaven testeja-des, sobretot la darrera, que segons amb quina configuració no funcionava.

A continuació, es va testejar i fer les modificacions oportunes pel correcte funcionament de les simulacions parametritzades. Finalment, amb això, la funcionalitat de les simulacions parametritzades quedava tancada.

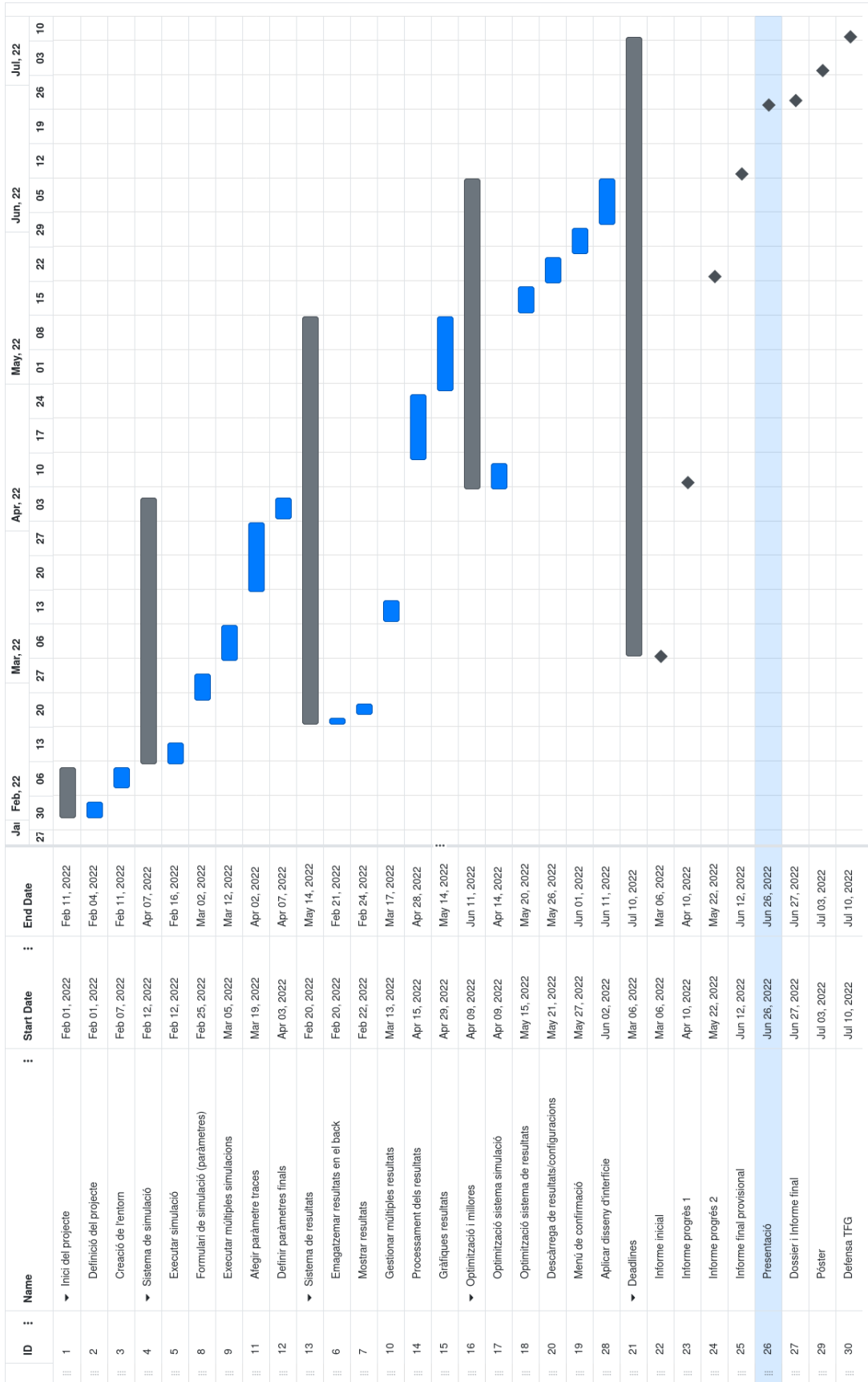
Les posteriors setmanes 10 i 11 es van centrar en el resultat de les simulacions, quins resultats interessava mostrar a l'usuari, quines opcions facilitaven l'ús de l'eina, etc.

Tanmateix, es va decidir el format de les gràfiques i implementar la possibilitat de descarregar tant les configuracions com els reports, una vegada s'acabés l'execució de les simulacions.

Durant les dues darreres setmanes a la segona entrega de l'informe de progrés, per una banda, es va implementar el menú de confirmació d'execució de simulacions, i per l'altra, l'execució de simulacions bulk amb la corresponent combinatòria a través del backtracking.

Per acabar, durant la mateixa setmana 14, corresponent a l'entrega del segon i últim informe, previ a l'informe final, es van implementar les gràfiques, adaptant els reports a l'entrada de dades de frappe, el framework utilitzat per la generació de gràfiques.

A.2 Cronograma



A.3 Mapa web

