Títol: Sistema de recomanació de continguts Big Data al Cloud
Autor: Jaume Jané Besora

Data: 1 de Desembre de 2014

Director: Daniel Giribet Farre
Institució del director: Televisió de Catalunya (TVC)

Ponent: Maria Carme Quer Bosor
Departament del ponent: Departament d'Enginyeria de Serveis i Sistemes d'Informació


Centre: Facultat d'Informàtica de Barcelona (FIB)
Universitat: Universitat Politècnica de Catalunya (UPC) BarcelonaTech
DADES DEL PROJECTE

Títol del projecte: Sistema de recomanació de continguts Big Data al Cloud

Nom de l’estudiant: Jaume Jané Besora


Crèdits: 37,5

Director: Daniel Giribet Farre

Institució del director: Televisió de Catalunya (TVC)

Ponent: Maria Carme Quer Bosor

Departament del ponent: Departament d’Enginyeria de Serveis i Sistemes d’Informació

MEMBRES DEL TRIBUNAL (nom i signatura)

President: Oscar Romero Moral

Vocal: Josep Ginebra Molins

Secretari/a: Maria Carme Quer Bosor

QUALIFICACIÓ

Qualificació numèrica:

Qualificació descriptiva:

Data: 1 de Desembre de 2014
1 Introducció ................................................................................................................. 6
  1.1 Context i motivació del projecte ............................................................................... 6
  1.2 Descripció del projecte ............................................................................................... 6
  1.3 Objectius del projecte .................................................................................................. 7
  1.4 Requisits ...................................................................................................................... 8
    1.4.1 Requisits Funcionals .............................................................................................. 8
    1.4.2 Requisits No Funcionals ........................................................................................ 8
  1.5 Decisió sobre l’arquitectura de Lupa ......................................................................... 9
  1.6 Selecció de tecnologies al mercat per cada subsistema ........................................... 10
    1.6.1 Recomanadors ....................................................................................................... 10
      1.6.1.1 Selecció de Criteris Tècnics ........................................................................... 11
      1.6.1.2 Selecció de Criteris No-Tècnics .................................................................... 12
      1.6.1.3 Ponderació dels criteris ................................................................................... 12
      1.6.1.4 Selecció Recomanadors Mercat ..................................................................... 14
        Mahout ....................................................................................................................... 14
        Easyrec ....................................................................................................................... 14
        Lenskit ....................................................................................................................... 15
        MyMediaLite ............................................................................................................... 15
        Rumblelabs .............................................................................................................. 15
      1.6.1.5 Avaluació Recomanadors ............................................................................... 16
      1.6.1.6 Selecció del Recomanador .............................................................................. 17
    1.6.2 Algoritme Similitud Notícies (KLD Classifier) ..................................................... 18
    1.6.3 Big Data ............................................................................................................... 19
    1.6.4 Programació distribuïda ....................................................................................... 20
      1.6.4.1 Amazon Elastic Cloud Compute (EC2) ............................................................ 21
  1.7 Decisió de la tecnologia ............................................................................................... 25
  1.8 Anàlisi de riscos ........................................................................................................ 26
2 Especificació .................................................................................................................. 28
  2.1 Especificació casos d’ús ............................................................................................ 28
    2.1.1 Inserir notícia al recomanador ............................................................................. 28
    2.1.2 Generar recomanacions ....................................................................................... 29
    2.1.3 Mostrar recomanacions ....................................................................................... 29
  2.2 Definició dels actors .................................................................................................... 30
  2.3 Diagrames de casos d’ús .......................................................................................... 31
  2.4 Diagrames de seqüència ............................................................................................ 31
    2.4.1 Inserir notícia ....................................................................................................... 32
    2.4.2 Mostrar recomanacions ....................................................................................... 32
    2.4.3 Generar recomanacions ....................................................................................... 33
  2.5 Contracte de les operacions ....................................................................................... 33
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5.1 Contractes Inserir Notícia</td>
<td>33</td>
</tr>
<tr>
<td>2.5.2 Contractes Mostrar recomanacions</td>
<td>34</td>
</tr>
<tr>
<td>2.5.3 Contractes Generar recomanacions</td>
<td>34</td>
</tr>
<tr>
<td>2.6 Model Conceptual</td>
<td>35</td>
</tr>
<tr>
<td>2.6.1 Diagrama de classes</td>
<td>35</td>
</tr>
<tr>
<td>2.6.2 Atributs de les classes</td>
<td>35</td>
</tr>
<tr>
<td>3 Disseny</td>
<td>37</td>
</tr>
<tr>
<td>3.1 Patró de disseny</td>
<td>37</td>
</tr>
<tr>
<td>3.1.1 Capa de Presentació</td>
<td>39</td>
</tr>
<tr>
<td>3.1.2 Capa de domini</td>
<td>39</td>
</tr>
<tr>
<td>3.1.3 Capa de gestió de dades</td>
<td>39</td>
</tr>
<tr>
<td>3.2 Diagrames de seqüència</td>
<td>40</td>
</tr>
<tr>
<td>3.2.1 Inserir notícia</td>
<td>40</td>
</tr>
<tr>
<td>3.2.2 Mostrar recomanacions</td>
<td>41</td>
</tr>
<tr>
<td>3.2.3 Generar recomanacions</td>
<td>42</td>
</tr>
<tr>
<td>4 Implementació</td>
<td>44</td>
</tr>
<tr>
<td>4.1 Aspectes importants</td>
<td>44</td>
</tr>
<tr>
<td>4.1.1 Importància de la claredat del codi</td>
<td>44</td>
</tr>
<tr>
<td>4.1.2 Clustering Jeràrquic</td>
<td>44</td>
</tr>
<tr>
<td>4.1.2.1 Clustering Jeràrquic</td>
<td>44</td>
</tr>
<tr>
<td>4.1.3 Topologia</td>
<td>47</td>
</tr>
<tr>
<td>4.1.3.1 Topologia sense clustering jeràrquic</td>
<td>47</td>
</tr>
<tr>
<td>4.1.3.1 Topologia amb clustering jeràrquic</td>
<td>48</td>
</tr>
<tr>
<td>4.1.4 Procés de recomanació</td>
<td>48</td>
</tr>
<tr>
<td>4.2 Desplegament en un clúster remot</td>
<td>49</td>
</tr>
<tr>
<td>4.3 Principals problemes trobats</td>
<td>50</td>
</tr>
<tr>
<td>4.3.1 Implementació pròpia recomanador</td>
<td>50</td>
</tr>
<tr>
<td>4.3.2 Implementació pròpia Clustering Jeràrquic</td>
<td>50</td>
</tr>
<tr>
<td>4.3.3 Connexió Freeling</td>
<td>50</td>
</tr>
<tr>
<td>4.3.4 Problemes amb la visualització dels resultats</td>
<td>51</td>
</tr>
<tr>
<td>4.4 Proves i correcció d’errors</td>
<td>51</td>
</tr>
<tr>
<td>5 Experimentació</td>
<td>53</td>
</tr>
<tr>
<td>5.1 Experimentació espai</td>
<td>53</td>
</tr>
<tr>
<td>5.2 Experimentació temps de resposta</td>
<td>53</td>
</tr>
<tr>
<td>5.2.1 Temps d’execució total</td>
<td>53</td>
</tr>
<tr>
<td>5.2.2 Temps d’execució per inserció</td>
<td>54</td>
</tr>
<tr>
<td>5.3 Experimentació recomanaci</td>
<td>54</td>
</tr>
<tr>
<td>5.3.1 Projecte Gutenberg</td>
<td>54</td>
</tr>
<tr>
<td>5.3.1.1 Experiment Projecte Gutenberg (I)</td>
<td>55</td>
</tr>
</tbody>
</table>
1 Introducció

En aquest primer capítol de la documentació del projecte s'explica en què consisteix el projecte, les motivacions que hi ha darrere d’aquest i les primeres decisions preses per tal de començar-lo a desenvolupar.

1.1 Context i motivació del projecte

A finals de Juliol de 2013 vaig començar unes pràctiques a Televisió de Catalunya (TVC) al departament de Recerca. La meva incorporació va ser en el projecte de Social Media on s’estudiaven temes relacionats amb els grans volums de dades generats per les Xarxes Socials i possibles usos mitjançant Big Data al Cloud.

Un cop finalitzat el meu conveni al Setembre de 2013, em van proposar la possibilitat de realitzar el Projecte de Final de Carrera enfocat en les tecnologies utilitzades en la meva estada prèvia. El projecte consistiria en recomanar continguts mitjançant Big Data per obtenir-los en temps real i obtenir la màxima escalabilitat desplegant-ho als serveis d’Amazon EC2.

La TVC ja disposa d’un recomanador de notícies, però que està basat en els temes introduïts pels redactors, per tant, l’objectiu principal del projecte, de cara a l’usuari final i als redactors, consisteix en millorar aquesta recomanació basant-la també en el contingut de la notícia.

El fet que cada cop es generin moltes més dades ja sigui per l’empremta dels propis usuaris o la inserció de continguts al web, fa que aparegui la necessitat de tractar aquestes dades amb els conceptes de Big Data si volem resultats en temps real. Això va fer que m’interessés en el projecte. A més, si observem els sistemes Web més populars on s’ofereixen continguts veiem que la recomanació dels propis continguts és una part important: Amazon, Youtube, Netflix, etc. Si a això s’hi afegeix el fet de poder treballar amb tecnologies poc (o gens) utilitzades durant la carrera, fet que suposa l’ampliació dels meus coneixements, obtenim un projecte molt interessant.

El projecte ha estat supervisat pel cap de Recerca de TVC Daniel Giribet.

1.2 Descripció del projecte

El projecte consisteix en la implementació d’un sistema software de recomanació de notícies Big Data al Cloud, el qual s’anomena Lupa.

El software ha d’estar enfocat tant a usuaris finals com a redactors de la pròpia casa. Per tant, tindrà dues utilitzacions diferenciades: per una banda quan els redactors estan cercant informació relacionada per redactar la seva notícia i per una altra, de cara l’usuari final (via Web, mòbil, etc.) li recomanem notícies que poden ser del seu interès.

La idea principal de cara a tant als redactors com a usuaris finals, com ja hem mencionat anteriorment, és millorar el resultats de la recomanació de notícies. Actualment, la recomanació es basa en les paraules claus associades pels redactors a cada notícia, que fan referència al tema de la notícia (Societat, Política, Europa, etc.).
Lupa basarà la recomanació en la similitud entre noticies mitjançant la diferència entre la distribució de les paraules.

Com a primera etapa del projecte, cal analitzar el mercat de productes software i seleccionar un software que ens serveixi de guia per tal d’implementar el nostre propi subsistema recomanador.

Un cop seleccionat el software recomanador, en una segona etapa cal implementar el nostre subsistema de recomanació i integrar-lo amb Storm. Storm és un sistema de computació distribuïda en temps real, que permetrà que el sistema resultant funcioni amb grans volums de dades (Big Data), i sigui escalable fàcilment (Cloud Computing). Per garantir l’escalabilitat un cop tingui la integració amb Storm, el desplegarem al Cloud (Amazon).

Finalment, la darrera etapa serà la integració de tot plegat amb Freeling (aplicació creada pel grup de recerca de llenguatge natural de l’UPC) per millorar la qualitat de la recomanació de les notícies. Freeling permetrà millorar l’anàlisi de similitud entre les notícies ja que permet reduir cada paraula a la seva base comuna. Per exemple aquells dos textos si no els passem per Freeling són totalment diferents:

1. El cotxe verd
2. Els cotxes verds

En canvi, el resultat de Freeling per cada frase és el següent:

1. El cotxe verd
2. El cotxe verd

**1.3 Objectius del projecte**

A partir del context, la motivació del projecte i la pròpia descripció definim els objectius del projecte, presentats a continuació:

- Desenvolupar Lupa per tal d’ofereir als redactors i als usuaris finals una recomanació més acotada que l’actual mitjançant la distribució de les paraules en una notícia, per exemple, si estan redactant/llegint una notícia relacionada amb la Política d’Ucraïna volem que recomani altres notícies, com per exemple, la situació social d’aquell país. Actualment només es recomanarien notícies sobre el tema Política, per tant, podríem estar recomanant notícies relacionades amb la Política Catalana que no serien rellevants pel cas.

- Programar Lupa per obtenir els resultats en temps real.

- Programar Lupa de manera que sigui continuament escalable, ja que cada cop tindrem més notícies en el data set.

- Programar Lupa de manera adequada per facilitar possibles ampliacions o contribucions futures.
1.4 Requisits

Durant l’anàlisi de requisits s’obtenen requisits de dos tipus: funcionals, que defineixen els serveis o funcions que ha de proveir un software, i no funcionals, que defineixen unes certes propietats de qualitat sobre com el software ha d’oferir els serveis o funcions.

1.4.1 Requisits Funcionals

Els requisits funcionals es definiran en detall a l’apartat d’especificació ja que es corresponen als casos d’ús del sistema. Bàsicament tindrem els següents requisits funcionals:

- Inserir notícia/es al recomanador
- Generar recomanacions
- Obtenir recomanacions

1.4.2 Requisits No Funcionals

Requisit: Exactitud

Descripció: Volem que els resultats de Lupa tinguin un nivell de precisió acceptable. És a dir, la recomanació millori els resultats del sistema de recomanació actual.

Requisit: Recursos

Descripció: Lupa haurà de treballar d’entrada amb 20.786 notícies, i el creixement del nombre de notícies anual és de 20.000 aproximadament, amb una mitjana de 1.666 notícies per mes.

Requisit: Eficiència

Descripció: Lupa haurà de ser eficient tant en temps com en recursos, ja que volem els resultats en temps real. Per tant, no voldrem un temps major a 1 minut (temps aproximat que tarda un redactor a publicar la notícia).

Requisit: Escalabilitat

Descripció: Lupa haurà de ser escalable (al augmentar les dades no es vol que sigui un problema en temps ni espai), ja que el volum de dades anirà creixent al llarg del temps.

Requisit: Extensibilitat

Descripció: Durant la implementació de Lupa s’haurà de tenir en compte que aquest haurà d’estar orientat a possibles extensions futures.
**Requisit:** Tolerància a errors

**Descripció:** En cas d'infringir els seus estàndards d'inputs volem que Lupa segueixi oferint un mínim de prestacions, ja que el nostre sistema no és podrà aturar per una simple excepció en el format de les dades.

**Requisit:** Capacitat de Recuperació

**Descripció:** En cas d'error, volem que el software pugi recuperar les dades ja processades, ja que el nostre sistema reenviarà les dades perdudes.

### 1.5 Decisió sobre l’arquitectura de Lupa

L’arquitectura global consistirà en un subsistema programat amb Storm que es comunicarà amb Redis (Base de dades) per gestionar les notícies/recomanacions i amb Freeling per reduir les paraules de les notícies a la base comuna:

![Il·lustració 1. Arquitectura global del sistema](image)

Per garantir els objectius d’escalabilitat i eficiència a nivell d’arquitectura dividirem el subsistema de Storm en vàries màquines on podrem afegir tantes màquines Slaves com vulguem (mitjançant Cloud Computing EC2 d’Amazon).
D’aquesta manera tenim l’escalabilitat de Slaves Storm assegurada, és a dir, si els Slaves s’encarreguen de la capacitat de càlcul de Storm llavors també podem millorar l’eficiència fàcilment afegint més nodes, per tant, més capacitat de càlcul.

Els conceptes de Storm i Amazon EC2 seran explicats en detall al següent capítol, el procés de recomanació s’integrarà dintre dels nodes de Storm com veurem al capítol d’implementació.

1.6 Selecció de tecnologies al mercat per cada subsistema

1.6.1 Recomanadors

Com s’ha explicat anteriorment una de les primeres etapes del projecte consisteix en la selecció d’un subsistema recomanador per entendre la lògica de funcionament i poder implementar un recomanador adaptat a les nostres necessitats.

Hi ha molts recomanadors al mercat, per seleccionar el mes adient hem realitzat un estudi amb la metodologia següent:

1. Selecció de criteris per avaluar els productes existents al mercat.
   Una cosa que cal fer en el projecte és l’avaluació dels recomanadors que existeixen al mercat. Per tal de fer aquesta avaluació cal definir uns criteris. Cal destacar, que tenim dos grups de criteris:
   - Tècnics: Són els que estan relacionats amb la funcionalitat del projecte i amb aspectes no funcionals.
   - No-Tècnics: Són els que estan més relacionats amb el productor, producte i el seu cost.
2. Ponderació d’aquests criteris.
Dels diferents criteris no tots tenen un mateix pes en la decisió de quin recomanador seleccionar. Caldrà d’alguna manera prioritzar quins són més importants en la decisió.

3. Selecció de diversos recomanadors del mercat.
Es tractarà de fer un estudi de mercat que ha consistit en una cerca de recomanadors existents a través de totes les fonts que es considerin adients.

4. Avaluació dels recomanadors mitjançant els criteris ponderats prèviament.

5. Selecció del recomanador que compleixi millor els criteris establerts

1.6.1.1 Selecció de Criteris Tècnics
La majoria dels criteris tècnics són els propis requisits no funcionals del projecte aplicats al subsistema de recomanació, els altres, estan més enfocats a nivell d’aprenentatge e integració amb el nostre sistema.

El ISO 25010 (International Organization for Standardization) [1] és un estàndard que proposa un model de qualitat dels productes software, que defineix les característiques de qualitat que ha de tenir qualsevol producte software. Per a cadascuna de les característiques de qualitat de l’estàndard, he descrit com hauria de ser el subsistema recomanador que interessaria per al present projecte:

- **Adequació**: Es vol que el subsistema recomanador aporti un conjunt de funcions que es divideixen en dues tasques independents, la generació de les recomanacions, i l’obtenció d’un conjunt de recomanacions.
- **Exactitud**: Es vol que els resultats del recomanador tinguin un nivell de precisió acceptable. És a dir, la recomanació estigui dins uns marges acceptables.
- **Tolerància a errors**: En cas d’infringir els seus estàndards d'inputs cal que el recomanador segueixi oferint un mínim de prestacions, ja que el sistema no es podrà aturar per una simple excepció en el format de les dades.
- **Capacitat de Recuperació**: En cas d’error, cal que el software pugi recuperar les dades ja processades, ja que el sistema reenviarà les dades perdudes.
- **Comportament temporal**: Cal que el software proporcioni temps de resposta apropriats al sistema. Ja que cal processar molta informació, i si els temps de resposta són molt grans limitarà els nostres objectius d’eficiència.
- **Utilització de recursos**: És necessari que el software sota unes condicions determinades consumeixi una quantitat de recursos adequats. Un dels objectius és aconseguir l’escalabilitat i el temps real sense haver de gastar molts recursos econòmics, per tant la utilització de recursos incideix directament amb un dels objectius.
- **Facilitat d’aprenentatge**: Capacitat del software per ser aprèn (Wiki, JavaDocs, GitHub-Wiki, etc.). És important que el software tingui una bona documentació com hem vist en l’anàlisi de riscos.
- **Facilitat d’ús**: Capacitat del software per ser utilitzat i controlat. Una característica important del software és la interacció que ens permet mitjançant les seves funcions, per poder visualitzar els resultats i la flexibilitat que ens ofereix per adaptar-ho al nostre problema.
- **Grau d’ajuda**: Nivell d’ajuda que ofereix el software quan l’usuari necessita assistència. En cas de necessitar consell, és important que els desenvolupadors pugin guiar-nos (via contacte, web, issues, etc.).
• **Autenticitat**: Els accessos a aquest software amb les dades només els poden dur a terme els subjectes autoritzats. Si el software és un servei web, ha de ser segur ja que sinó un subjecte no autoritzat podria accedir a les nostres dades.

• **Interoperabilitat**: El software ha de ser capaç d’interactuar/integrar-se amb el sistema distribuït, per poder desplegar el recomanador al cloud.

• **Canviabilitat**: Capacitat per poder modificar/implementar funcionalitats del software, en el nostre cas volem implementar nosaltres la funció que relaciona els continguts (distribució de paraules).

• **Adaptabilitat**: Capacitat del software per ser adaptat a diferents entorns específics (Linux, Ubuntu, Centos, Windows ...) aplicant les mateixes accions.

### 1.6.1.2 Selecció de Criteris No-Tècnics

Per determinar els criteris s’ha utilitzat un catàleg de criteris no-tècnics elaborat pel grup de recerca GESSI [2] de la UPC. Hem seleccionat els criteris que tenien sentit pel cas de la selecció del subsistema recomanador del nostre projecte:

• **Software Cost**: Cost per adquirir el software/liciticies.

• **Opinions**: Opinions del software (Google Scholar).

### 1.6.1.3 Ponderació dels criteris

Un cop tenim els criteris seleccionats, el següent pas consisteix en determinar la importància de cada criteri. Aquesta tasca és molt subjectiva, i per tant, la ponderació individual entre diversos individus pot ser molt diferent. Així que, utilitzem “The analytic hierarchy process (AHP)” [3][Error! No se encuentra el origen de la referencia.], sense entrar molt en detall, la metodologia AHP consisteix en comparar tots els criteris dos a dos reduint els problemes de subjectivitat i redundància [4][Error! No se encuentra el origen de la referencia.] mitjançant els següents valors:

<table>
<thead>
<tr>
<th>Nivell d’importància</th>
<th>Descripció</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mateixa importància</td>
</tr>
<tr>
<td>3</td>
<td>Diferència moderada en importància</td>
</tr>
<tr>
<td>5</td>
<td>Diferència essencial en importància</td>
</tr>
<tr>
<td>7</td>
<td>Diferència gran en importància</td>
</tr>
<tr>
<td>9</td>
<td>Diferència extrema en importància</td>
</tr>
<tr>
<td>Recíprocs</td>
<td>1/Nivell d’importància</td>
</tr>
</tbody>
</table>

Taula 1. Nivells d’importància AHP
El resultat d’aquesta comparació entre tots els criteris seleccionats és el següent:

Cada cel·la representa la importància del criteri de la fila i respecte al criteri de la columna j. El següent pas consisteix en la normalització d’aquests valors i l’extracció de les ponderacions.

Per normalitzar simplement dividirem cada cel·la d’una columna per la suma total de la columna, com estipula el AHP. Un Cop tinguem totes les divisions computades ja només caldrà calcular la mitjana de cada fila per saber la ponderació de cada criteri. A la següent taula tenim les columnes ja normalitzades, i afegim una columna més amb les ponderacions respectives.
1.6.1.4 Selecció Recomanadors Mercat

Per poder seleccionar un recomanador, s'ha realitzat un estudi del mercat dels possibles softwares de recomanació que podrien adaptar-se als criteris seleccionats en els apartats anteriors. Tot seguit tenim una breu descripció dels aspectes més rellevants de cada software que hem seleccionat.

**Mahout**

Il·lustració 3. Logotip Mahout

Mahout és un projecte de la fundació de software Apache, el qual ofereix softwares lliures d'algoritmes escalables d'aprenentatge automàtic. La temàtica de la major part de softwares està enfocada a:

- Filtratge col·laboratiu
- Clústering
- Classificació

**Easyrec**

Il·lustració 4. Logotip Easyrec

Easyrec és una aplicació web de codi obert, la qual ofereix recomanacions personalitzades utilitzant serveis web RESTful. Està sota Llicència Pública General de la GNU i compartit a SourceForge. Està implementat en Java, utilitza una base de dades MySQL i ve amb una eina d'administració.
Lenskit

Lenskit és una eina de codi obert per a la creació, la investigació i l'estudi dels sistemes de recomanació. Proporciona implementacions d'alta qualitat dels algoritmes de filtratge col·laboratiu i està dissenyat per integrar-ho en aplicacions web i altres entorns de complexitat similar.

MyMediaLite

MyMediaLite Recommender System Library

MyMediaLite és una biblioteca polivalent d'algoritmes de sistemes de recomanació, implementats en C#. S'adreça als dos escenaris més comuns en filtratge col·laboratiu:

- Rating prediction (per exemple en una escala d’1 a 5 estrelles).
- Predicció d’items per només feedback positiu (per exemple dels clics o likes)

MyMediaLite és programari lliure, que pots ser utilitzat i distribuït sota els termes de la Llicència Pública General de GNU (GPL).

Rumblelabs

Rumble Labs és una companyia d'anàlisis predictiu centrada en l'aprofitament de les dades generades per les xarxes socials, grans volums de dades i aprenentatge automàtic en temps real. Està basat en grafs de comportament de cada consumidor individual que s'utilitza per predir les seves accions futures dins dels mitjans de comunicació i els comerços electrònics del servei a través del web, mòbil i Tablet PC. Aquest servei és comercial i s’ofereix mitjançant serveis web (REST) o amb widgets (JavaScript).
1.6.1.5 Avaluació Recomanadors

En aquest apartat s'ha procedit a avaluar els diferents software que hem escollit en l’apartat anterior, puntuant-los en una escala d’1 a 10 per cada criteri de qualitat que hem escollit. Aquesta valoració intentarà ser tan objectiva com sigui possible, però degut a que no podem provar tots els softwares ja que no és l’objectiu del projecte no podrem evitar que tingui una desviació subjectiva en algunes valoracions.

La nota final per cada software la obtindrem mitjançant el sumatori de la nota de cada criteri multiplicat per la ponderació del criteri obtinguda mitjançant el mètode AHP.

<table>
<thead>
<tr>
<th>Ponderacions</th>
<th>Mahaut</th>
<th>Easyrec</th>
<th>LumaKit</th>
<th>MyMediaLite</th>
<th>Reminders</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adequació(Appropriateness)</td>
<td>0.149</td>
<td>9</td>
<td>7</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>Exactitud(Accuracy)</td>
<td>0.149</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>Tolerància a errors(Fault tolerance)</td>
<td>0.058</td>
<td>8</td>
<td>10</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>Capacitat de recuperació(Recoverability)</td>
<td>0.058</td>
<td>7</td>
<td>9</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>Comportament temporal(Time behaviour)</td>
<td>0.058</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>Utilització de recursos(Resource utilisation)</td>
<td>0.058</td>
<td>8</td>
<td>10</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Facilitat d’aprenentatge(Learnability)</td>
<td>0.021</td>
<td>7</td>
<td>6</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>Facilitat d’ús(Ease of use)</td>
<td>0.021</td>
<td>7</td>
<td>6</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>Grau d’ajuda</td>
<td>0.021</td>
<td>7</td>
<td>7</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>Autenticitat</td>
<td>0.058</td>
<td>8</td>
<td>10</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Interoperabilitat(Interoperability)</td>
<td>0.021</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Canviabilitat</td>
<td>0.058</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Adaptabilitat(Adaptability)</td>
<td>0.058</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Software Cost</td>
<td>0.149</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Opinions(Google Schoolar)</td>
<td>0.058</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>7</td>
</tr>
</tbody>
</table>

Taula 4. Avaluació Recomanadors
1.6.1.6 Selecció del Recomanador

En l’avaluació de l’apartat anterior es veu com LensKit i Rummblelabs tenen molt bones notes en quant a adequació i exactitud, això és degut a que proporcionen moltes funcionalitats en el camp de Filtratge Col·laboratiu, sobre tot Lenskit (Ens ofereix quatre diferents algoritmes per poder fer la nostre recomanació amb diferents configuracions per cada algoritme).

La tolerància a errors i capacitat de recuperació obtenen millor nota Easyrec i Rummblelabs degut a que els dos són serveis web, i per tant, en cas de fallida només cal tornar-los a invocar amb els inputs corresponents.

Respecte a la utilització de recursos tenen tots la mateixa nota, excepte els dos serveis web, els quals no utilitzaran recursos del nostre sistema sinó dels seus servidors corresponents.

LensKit té molt millors notes en quant a aprenentatge, facilitat d’ús i grau d’ajuda degut a que el projecte està pujat al Git-Hub i té una pàgina de Wiki molt detallada. També el desenvolupador de LensKit està molt actiu amb les issues reportades per els usuaris i als fòrums.

Cap software sembla tenir problemes de integració amb un sistema distribuït ja que ho son serveis web, o estan implementats en C o Java. Ídem per la adaptabilitat.

La Canviabilitat depèn de si disposem del codi, en el cas dels softwares que son codi obert si que podem, però els serveis web òbviament no podrem modificar el seu codi.

El Cost del software és per tots gratuït excepte Rummblelabs que té un cost bastant elevat ja que per oferir-te el servei, necessiten la teva direcció web per poder estudiar-la i adaptar el seu servei. Les opinions estan basades en diversos articles on es comparen els diferents softwares.

Podem veure com la millor nota l’obté LensKit. Per tant, ha sigut el software que s’ha fet servir per entendre la metodologia de recomanació necessària pel nostre projecte i poder-la implementar. En el següent capítol, s’explica l’algoritme de recomanació que s’ha implementat per comparar les notícies.
1.6.2 Algoritme Similitud Notícies (KLD Classifier)

Un cop s'ha decidit quin software utilitzarem per entendre la lògica que ha de seguir el nostre subsistema de recomanació, es necessari entendre quin algoritme s'ha utilitzat per comparar les notícies ja que serà el motor del nostre subsistema de recomanació.

Com hem mencionat en la descripció del projecte Lupa calcula la similitud entre notícies mitjançant la diferència entre la distribució de les paraules. Això s'ha fet mitjançant la divergència de Kullback-Leibler(KL) \[ \text{Error! No se encuentra el origen de la referencia.} \] , també coneguda com entropia relativa. KL és una mesura de la similitud entre dos funcions de distribució de probabilitats P i Q:

\[
D(P||Q) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)} \tag{1}
\]

El problema de la divergència de KL és que no és una mètrica de distància ja que no és simètrica, així que utilitzarem la distància de Kullback-Leibler(KLD):

\[
D(P||Q) = \sum_{x \in X} \left( P(x) - Q(x) \right) \log \frac{P(x)}{Q(x)} \tag{2}
\]

Les notícies estan representades per un conjunt de termes(paraules) \( n = (t_i, t_j, ..., t_p) \), on cada \( t_k \) és una terme de la notícia. Per tant, podem definir les probabilitats d'una notícia \( P(t_k, n_j) \) de la següent manera:

\[
P(t_k, n_j) = \begin{cases} P(t_k | n_j) & \text{Si } t_k \text{ apareix a la notícia } n_j \\ \varepsilon & \text{Altrament} \end{cases} \tag{3}
\]

Amb:

\[
P(t_k | n_j) = \frac{tf(t_k, n_j)}{\sum_{x \in n_j} tf(t_x, n_j)}
\]

On:

- \( P(t_k | n_j) \) és la probabilitat del terme \( t_k \) en la notícia \( n_j \) amb \( \sum_{x \in n_j} P(t_x | n_j) = 1 \);
- \( tf(t_x, n_j) \) és 0 quant el terme \( t_x \) no apareix a la notícia \( n_j \), i igual a \( \#(t_x, n_j) \) quan el terme \( t_x \) apareix a la notícia \( n_j \).
- \( \beta \) és un coeficient de normalització el qual varia en funció de la mida de la notícia;
- \( \varepsilon \) és un llindar de probabilitat per tots els termes que no pertanyen a \( n_j \).

Els valors de \( \beta \) i \( \varepsilon \) han de mantenir la condició on les corresponents probabilitats sumin 1.
\[ \sum_{k \in n_j} \beta P(t_k | n_j) + \sum_{k \in e \cap n_j \cap V} \varepsilon = 1 \]

On:

- \( V \) és la unió de tots els termes de les dos notícies que estiguem comparant.

Podem estimar fàcilment \( \beta \) de la següent manera:

\[ \beta = 1 - \sum_{k \in e \cap n_j \cap V} \varepsilon \]

El valor \( \varepsilon \) és un llindar de probabilitat que ve donat per els termes no per la equació de la notícia \( P(t_k, n_j) = \begin{cases} P(t_k | n_j) & \text{Si } t_k \text{ apareix a la notícia } n_j \\ \varepsilon & \text{Altrament} \end{cases} \) (3).

Aquesta probabilitat ha de ser més petita que la probabilitat més baixa d’un terme en la notícia. Conseqüentment, aquest valor l’obtenim experimentalment.

Per tant, adaptant l’equació \( D(P||Q) = \sum_{x \in X} \left( P(x) - Q(x) \right) \log \left( \frac{P(x)}{Q(x)} \right) \) (2) de la distància de Kullback-Leibler(KLD) obtindrem la següent funció de distància pel nostre recomanador:

\[ KLD(n_i, n_j) = \sum_{k \in V} \left( P(t_k, n_i) - P(t_k, n_j) \right) \ast \log \left( \frac{P(t_k, n_i)}{P(t_k, n_j)} \right) \] (4)

### 1.6.3 Big Data

Com ja s’ha comentat anteriorment, el volum de dades a Internet està creixent de forma exponencial ja siguin generades pels propis usuaris a les xarxes socials o simplement continguts compartits al web. Això creà la necessitat de tractar informació com a grans conjunts de dades.

Big Data s’aplica a un conjunt de dades que superen la capacitat del software actual per ser recollides, processades i analitzades en temps real. En el nostre cas, volem que Lupa realitzi les operacions en temps real, ja que si processem la informació actual no volem obtenir els resultats en un futur llunyà, ja que estarem perdent temps per poder explotar els resultats.

Utilitzarem Storm per poder gestionar els grans volums de notícies que disposem actualment. També cal remarcar que el nombre de notícies que suportarà el nostre sistema anirà incrementant al llarg del temps, degut a què és un sistema pensat per anar actualitzant el data set dia a dia.

Storm és una eina per a processar en temps real de fluxos de dades. El que el fa diferent és que és una abstracció a més alt nivell que el simple enviament de missatges ja que permet definir topologies amb grafs acíclics dirigits. Les seves característiques principals són:
• Tolerància a fallades per a cada procés.
• Garanteix que cada missatge es processarà com a mínim una vegada.

Les Topologies de Storm estan compostes per dos tipus de nodes: Spouts i Bolts. Els nodes normalment reben dades, les processen i les emeten cap a altres nodes:

Com podem veure a la figura anterior la topologia s'encarrega de decidir com encadenem els diferents nodes. Per una banda, els Spouts són nodes els quals introdueixen o generen dades que seran processades per altres nodes. Per altra banda, els Bolts reben dades les processen i les poden emetre a altres Bolts o simplement tracten els resultats ja sigui guardant-los a una BD, visualitzant-los en una web, etc.

1.6.4 Programació distribuïda

La programació distribuïda és un paradigma de programació enfocat en el desenvolupament de sistemes distribuïts, escalables, transparent i tolerants a errors. Un sistema distribuït és un sistema software on els seus components és comuniquen i coordinen mitjançant la xarxa.

En el nostre cas, Storm ja s'encarrega de dividir el treball en aquest sistema distribuït. El sistema distribuït es sol anomenar Clúster. Aquest Clúster està format bàsicament per un Master Node anomenat Nimbus, ZooKeeper i un conjunt d’esclaus (Slaves) que realitzen les tasques.
El Nimbus és el nom del Master Node, el qual s’encarrega de distribuir el codi a la resta del clúster (Slaves), assignant tasques i monitoritzant els errors. Els Slaves s’encarreguen de realitzar les tasques assignades per el Nimbus. Cada Slave té un supervisor el qual es comunica amb el Nimbus mitjançant el ZooKeeper.

Il·lustració 9. Diagrama Clúster Storm

A l’hora de desplegar aquest clúster podem fer-ho en mode local amb els recursos (màquines) de TVC o bé mitjançant algun servei al núvol que ofereixi recursos com Amazon EC2. L’vantatja de desplegar-ho localment és que el cost serà reduït, ja que s’utilitzarà les pròpies màquines i els cost es limitarà a mantenir les màquines enceses. Tot i això, s’estarà limitat pel nombre de màquines i la potència que aquestes tinguin.

L’altre opció consisteix utilitzar el servei d’Amazon EC2, el qual ens ofereix les màquines que vulguem i les paguem només pel temps que les utilitzem. El principal problema és que si es fa així s’afegirà un cost extra, però no es tindrà cap limitació en quan a recursos ja que mentre paguem podrem tenir tants recursos com vulguem.

1.6.4.1 Amazon Elastic Cloud Compute (EC2)

Amazon Web Services ofereix un conjunt de serveis, els quals formen una plataforma de computació fiable, escalable i de baix cost "al núvol". Nosaltres ens centrarem en el servei EC2 el qual és un servei que proporciona capacitat de computació al núvol. Està dissenyat per facilitar recursos de computació fàcilment escalables a desenvolupadors. El nostre interès per aquest servei és el desplegament dels nodes del clúster de Storm al núvol.
Les instàncies d'Amazon són com servidors virtuals on podem executar aplicacions. Aquestes instàncies estan formades per una imatge d'Amazon també anomenada AMI (Amazon Machine Image) i un tipus d'instància:

- Una AMI defineix l'entorn de treball, el qual utilitzarem en aquesta màquina tant a nivell de software com de SO.
- El tipus d'instància defineix el hardware de la màquina, per tant, està directament relacionat amb el cost.

Les instàncies és diferencien a nivell de hardware (CPU, Memòria, I/O, Xarxa, GPU, etc.). Podem veure un subconjunt d'instàncies a la següent imatge:

Si volem que en aturar la instància no perdem la informació haurem de pagar per aquesta memòria mitjançant el servei de Amazon Elastic Block Store (EBS), el qual proporciona volums d'emmagatzematge dissenyats per utilitzar-los amb les instàncies d'EC2.
Amazon Elastic Block Store

Il·lustració 11. Amazon Elastic Block Store

Els costos d’utilitzar els serveis d’Amazon EC2 varien en funció de la instància seleccionada i del temps que la utilitzem, podem veure un subconjunt de les instàncies a la regió US West (Oregon) a la següent taula:

On-Demand Instance Prices

<table>
<thead>
<tr>
<th>Region: US West (Oregon)</th>
</tr>
</thead>
<tbody>
<tr>
<td>vCPU</td>
</tr>
<tr>
<td>General Purpose - Current Generation</td>
</tr>
<tr>
<td>m3.medium</td>
</tr>
<tr>
<td>m3.large</td>
</tr>
<tr>
<td>m3.xlarge</td>
</tr>
<tr>
<td>m3.2xlarge</td>
</tr>
<tr>
<td>General Purpose - Previous Generation</td>
</tr>
<tr>
<td>m1.small</td>
</tr>
<tr>
<td>m1.medium</td>
</tr>
<tr>
<td>m1.large</td>
</tr>
<tr>
<td>m1.xlarge</td>
</tr>
<tr>
<td>Compute Optimized - Current Generation</td>
</tr>
<tr>
<td>c3.large</td>
</tr>
<tr>
<td>c3.xlarge</td>
</tr>
<tr>
<td>c3.2xlarge</td>
</tr>
<tr>
<td>c3.4xlarge</td>
</tr>
<tr>
<td>c3.8xlarge</td>
</tr>
</tbody>
</table>

Il·lustració 12. Preus Amazon EC2 instàncies
Tenim una opció de realitzar un pagament puntual (lleuger, mitjà, intens) per cada instància que vulguem reservar i a canvi rebrem un descompte important en el preu-hora de la instància, vegem un exemple en la següent taula d’un pagament puntual lleuger:

![Il·lustració 13. Preus Amazon EC2 instàncies amb pagament puntual lleuger](image1)

Per analitzar tots aquests preus hem realitzat una estudi de la viabilitat del pressupost, per intentar simular els costos del desplegament del sistema resultant del projecte. Suposarem que farem servir 4 màquines que treballaran 24h diàries distribuïdes de la següent manera:

- 1 màquina pel Nimbus de tipus Small.
- 1 màquina pel Zookeeper de tipus Small.
- 2 màquines pels Slaves de tipus Large.

Donarem més capacitat de computació als Slaves ja que són els que s’encarreguen de realitzar tots els càlculs.

Partint d’aquest context obtenim els següents costos:

![Il·lustració 14. Pressupost contractació 4 màquines amazon(2S & 2L)](image2)

En el capítol de costos s’analitza aquestes dades i es decideix quin tipus de pagament és el mes convenient.
1.7 Decisió de la tecnologia

Les tecnologies del sistema venen definides principalment pels objectius del projecte que hem citat anteriorment.

- **Storm**: Per poder assolir l'objectiu de temps real s'ha utilitzat Storm el qual és un sistema de computació distribuïda en temps real. És escalable, tolerant a errors i garanteix que totes les dades seran processades.

- **Zookeeper**: Apache ZooKeeper és un projecte de software de l'Apache Software Foundation, que proveix un servei de configuració centralitzat per a sistemes distribuïts. En el nostre cas, s'ha fet servir per a la comunicació entre els nodes de la topologia de Storm.

- **Java**: La base del nostre sistema és Storm, degut a que la seva implementació és en java.

- **Redis**: És una Base de dades NoSQL, basada en el paradigma Clau-Valor. S'ha utilitzat per gestionar les entrades/sortides de la topologia de Storm amb eficiència i per tractar l'escalabilitat de Lupa.

- **LensKit**: És una eina de codi obert per a la creació, la investigació i l'estudi dels sistemes de recomanació. Proporciona implementacions d’alta qualitat dels algoritmes de filtratge col·laboratiu i està dissenyada per integrar-los en aplicacions web i altres entorns de complexitat similar. S'ha utilitzat per entendre la lògica de recomanació.

- **Amazon Elastic Cloud Compute (EC2)**: És un servei web que proporciona capacitat de computació al núvol. Està dissenyat per facilitar recursos de computació fàcilment escalables a desenvolupadors. Ens ha fet falta per poder desplegar-hi el clúster de Storm i obtenir l’escalabilitat que volíem.

- **Storm-deploy**: Aquest projecte està implementat pel propi dissenyador de Storm, el qual ens facilita el desplegament d’un clúster a AWS.

- **Meteorit**: És un producte Open Source que ens facilita la instal·lació de Storm i les seves dependències en una màquina.

- **Trident-mli**: És un producte Open Source enfocat a diversos algoritmes d’aprenentatge per computadors. S’ha fet servir per definir la funció de similitud entre dues notícies.

- **d3js**: Llibreria en javascript orientada a la visualització. S’ha fet servir per crear una visualització per facilitar el testing.

- **Freeling**: És un software creat pel grup de recerca de llenguatge natural de l’UPC, el qual s'ha utilitzat per millorar la qualitat de la recomanació de les notícies, ja que permet reduir cada paraula a la seva base comuna.
1.8 Anàlisis de riscos

A l'implementar un producte software hem de tenir en compte una sèrie de riscos. Quan parlem de riscos ens referim a allò que pot afectar negativament al projecte i que té certa probabilitat de passar. Així doncs, l’únia manera de minimitzar els efectes d’aquests riscos és detectar-los abans del desenvolupament del projecte, quantificar en quina mesura poden ser perjudicials i planificar solucions per actuar en cas que ens hi trobem.

A continuació definim possibles riscos als que estarà exposat el nostre projecte.

<table>
<thead>
<tr>
<th>Risc</th>
<th>Mala planificació temporal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Descripció</td>
<td>Un dels riscos més importants als que està exposat el nostre projecte és una mala planificació temporal, fet que implicarà que la duració del projecte augmenti, provocant un possible incompliment de la data d’entrega del projecte.</td>
</tr>
<tr>
<td>Impacte</td>
<td>Mitj</td>
</tr>
<tr>
<td>Probabilitat</td>
<td>Mitj</td>
</tr>
<tr>
<td>Mesures</td>
<td>Degut a que encara no hi ha una data d’entrega del projecte fixa, si les hores extra no són molt nombroses les podré realitzar igualment sense afectar negativament el projecte. En cas que la quantitat d’hores extra sigui massa elevada, es prescindirà de les funcionalitats menys importants.</td>
</tr>
</tbody>
</table>

Taula 5. Risc planificació

<table>
<thead>
<tr>
<th>Risc</th>
<th>No es disposa de suficient documentació per elaborar el projecte</th>
</tr>
</thead>
<tbody>
<tr>
<td>Descripció</td>
<td>Ja que treballaré amb certes tecnologies amb les quals no he treballat mai, em caldrà tenir suficient documentació per a poder complir tots els objectius.</td>
</tr>
<tr>
<td>Impacte</td>
<td>Alt</td>
</tr>
<tr>
<td>Probabilitat</td>
<td>Moltaixa</td>
</tr>
<tr>
<td>Mesures</td>
<td>Abans d’iniciar el projecte estudiaré totes les tecnologies que utilitzaré, intentant no incloure’n cap que estigui massa poc documentada. En cas de produir-se aquesta situació un cop iniciat el projecte, es substituiria la tecnologia en qüestió per alguna altra que ofereixi resultats similars.</td>
</tr>
</tbody>
</table>

Taula 6. Risc documentació tecnologies
Risc: Cost molt elevat dels serveis de Cloud Computing de Amazon

| Descripció | Les màquines que utilitzem del servei web EC2 d'Amazon tenen un cost econòmic. |
| Impacte | Alt |
| Probabilitat | Molt baixa |
| Mesures | Abans d'iniciar el projecte estudiarem els diferents costos que poden derivar d'aquest servei. En cas de trobar-nos en aquesta situació un cop iniciat el projecte, es substituirien els recursos d'Amazon per propis. |

Taula 7. Risc costos serveis Amazon
2 Especificació

A l’especificació es descriu en detall el comportament extern del sistema descrivint els requisits funcionals que el software ha de complir. Per fer l’especificació s’ha utilitzat el llenguatge UML (Unified Modeling Language).

2.1 Especificació casos d’ús

A l’especificació del casos d’ús s’ha descrit textualment cadascun dels requisits funcionals presentats en el capítol d’introducció.

2.1.1 Inserir notícia al recomanador

<table>
<thead>
<tr>
<th>Cas d’ús</th>
<th>Inserir notícia al recomanador</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actor</strong></td>
<td>Usuari</td>
</tr>
<tr>
<td><strong>Casos d’us relacionats</strong></td>
<td>L’usuari insereix una notícia al recomanador</td>
</tr>
<tr>
<td>Descripció</td>
<td>L’identificador de la notícia no existeix a la base de dades</td>
</tr>
<tr>
<td>Precondició</td>
<td>La notícia s’insereix correctament</td>
</tr>
<tr>
<td>Postcondició</td>
<td>1. L’usuari indica l’identificador, títol i el text de la notícia</td>
</tr>
<tr>
<td>Curs Principal</td>
<td>2. El sistema valida les dades</td>
</tr>
<tr>
<td></td>
<td>3. El sistema introdueix la notícia al sistema</td>
</tr>
<tr>
<td>Cursos Alternatius</td>
<td>2. Les dades no són correctes.</td>
</tr>
<tr>
<td></td>
<td>2.1. El sistema notifica l’error a l’usuari.</td>
</tr>
<tr>
<td></td>
<td>2.2. Torna al pas 1.</td>
</tr>
</tbody>
</table>

**Taula 8. Cas d’ús inserir notícia**
### 2.1.2 Generar recomanacions

<table>
<thead>
<tr>
<th>Cas d’ús</th>
<th>Generar recomanacions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>Sistema</td>
</tr>
<tr>
<td>Casos d’us relacionats</td>
<td>Inserir notícia al recomanador</td>
</tr>
<tr>
<td>Descripció</td>
<td>El sistema ha de processar la notícia i generar les recomanacions</td>
</tr>
<tr>
<td>Precondició</td>
<td>La notícia existeix a la base de dades</td>
</tr>
<tr>
<td>Postcondició</td>
<td>Es generen les recomanacions per a la notícia i són guardades a la base de dades</td>
</tr>
</tbody>
</table>

#### Curs Principal

1. El sistema introdueix la notícia a la Topologia de Storm.
2. El sistema normalitza la notícia mitjançant Freeling
3. El sistema calcula la distribució de les paraules i les guarda a la base de dades
4. El sistema cerca el clúster que li pertoca a la notícia dins el clúster jeràrquic de la base de dades
5. El sistema compara la notícia amb les del clúster retomat i genera les millors recomanacions.

#### Cursos Alternatius

4. El clúster resultant ha assolit la capacitat màxima
4.1. El sistema separa el clúster mitjançant k-means
4.2. El sistema guarda el canvi al clúster jeràrquic de la base de dades
4.3 Retorna el nou clúster assignat a la notícia
4.4 Torna al pas 5

---

### 2.1.3 Mostrar recomanacions

<table>
<thead>
<tr>
<th>Cas d’ús</th>
<th>Mostrar recomanacions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>Usuari</td>
</tr>
<tr>
<td>Casos d’us relacionats</td>
<td>Generar recomanacions</td>
</tr>
<tr>
<td>Descripció</td>
<td>L’usuari vol obtenir les recomanacions per una determinada notícia</td>
</tr>
<tr>
<td>Precondició</td>
<td>La notícia existeix a la base de dades</td>
</tr>
<tr>
<td>Postcondició</td>
<td>Es mostren les recomanacions per a la notícia desitjada</td>
</tr>
</tbody>
</table>

#### Curs Principal

1. L’usuari indica l’identificador de la notícia
2. El sistema valida les dades
3. El sistema mostra les recomanacions de la notícia

#### Cursos Alternatius

2. L’identificador no existeix.
2.1. El sistema notifica l’error a l’usuari.
2.2. Torna al pas 1.

---

Taula 9. Cas d’ús generar recomanacions

Taula 10. Cas d’ús mostrar recomanacions
2.2 Definició dels actors

En aquest apartat s’han descrit els diferents actors que intervenen al sistema. Com que algunes tasques són cridades automàticament pel sistema quan es compleixen certes condicions, el sistema també és un actor.

**Il·lustració 15. Definició actors**

**Usuari:** Actor que representa totes les persones que insereixen notícies o consulten recomanacions en el nostre sistema.

**Sistema:** Actor que representa el software de Lupa que, com ja s’ha mencionat anteriorment, actuarà també d’actor en alguns casos d’ús.
2.3 Diagrames de casos d’ús

Els diagrames de casos d’ús descriuen la interacció entre l’usuari i el sistema.

Il·lustració 16. Diagrames de casos d’ús

2.4 Diagrames de seqüència

Els diagrames de seqüència descriuen la interacció entre l’actor i el sistema de cada cas d’ús, permetent identificar les operacions del sistema en una seqüència temporal.

A continuació es presenten els diagrames de seqüència d’especificació dels casos d’ús presentats anteriorment. A l’etapa de disseny s’expandiran aquests diagrames de seqüència, fent-los més concrets i adaptats a la tecnologia utilitzada.
2.4.1 Inserir notícia

Il·lustració 17. Diagrama de seqüència inserir notícia

2.4.2 Mostrar recomanacions

Il·lustració 18. Diagrama de seqüència mostrar recomanacions
2.4.3 Generar recomendacions

Il·lustració 19. Diagrama de seqüència generar recomanacions

2.5 Contracte de les operacions

Els contractes de les operacions descriuen el sistema en termes de quins són els canvis d’estat i quines són les sortides que el sistema proporciona quan s’invoca l’operació. Per cada operació es genera un contracte que informa sobre la semàntica, excepcions, sortides, precondicions i postcondicions.

2.5.1 Contractes Inserir Notícia

<table>
<thead>
<tr>
<th>Operació</th>
<th>crearNoticia(id, titol, text)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paràmetres</td>
<td>Es passa com a paràmetre la informació necessària per inserir una notícia: identificador de la notícia, títol de la notícia i el text de la notícia.</td>
</tr>
<tr>
<td>Semàntica</td>
<td>Comunica al sistema que es vol inserir una notícia.</td>
</tr>
<tr>
<td>Cas d’ús</td>
<td>Inserir Notícia.</td>
</tr>
<tr>
<td>Excepcions</td>
<td>- Si id, titol, text són buits s’indica que són obligatoris.</td>
</tr>
<tr>
<td></td>
<td>- Si l’identificador ja existeix al servidor s’indica que ja existeix una notícia amb aquest id.</td>
</tr>
<tr>
<td>Precondicions</td>
<td>Les dades han estat validades i s’insereix la notícia al sistema.</td>
</tr>
<tr>
<td>Postcondicions</td>
<td></td>
</tr>
</tbody>
</table>

Taula 11. Contracte Inserir Notícia
### 2.5.2 Contractes Mostrar recomanacions

<table>
<thead>
<tr>
<th>Operació</th>
<th>mostrarRecomanacions(id)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paràmetres</td>
<td>Es passa com a paràmetre la informació necessària per poder identificar la notícia que volem saber les recomanacions: identificador de la notícia.</td>
</tr>
<tr>
<td>Semàntica</td>
<td>Comunica al sistema que es vol veure les recomanacions d'una notícia determinada.</td>
</tr>
<tr>
<td>Cas d'ús</td>
<td>Mostrar recomanacions.</td>
</tr>
</tbody>
</table>
| Excepcions | - Si identificador és buit s’indica que és obligatori.  
- Si l'identificador no existeix al servidor s’indica que no existeix una notícia amb aquest identificador. |
| Precondicions | L'identificador de la notícia és correcte. |
| Postcondicions | Es visualitza la recomanació de la notícia especificada. |

**Taula 12. Contracte Mostrar recomanacions**

### 2.5.3 Contractes Generar recomanacions

<table>
<thead>
<tr>
<th>Operació</th>
<th>generarRecomanacions(id)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paràmetres</td>
<td>Es passa com a paràmetre la informació necessària per generar les recomanacions d'una notícia determinada: identificador de la notícia.</td>
</tr>
<tr>
<td>Semàntica</td>
<td>Comunica al sistema que es generaràn es recomanacions per una notícia determinada.</td>
</tr>
<tr>
<td>Cas d'ús</td>
<td>Generar recomanacions.</td>
</tr>
<tr>
<td>Excepcions</td>
<td></td>
</tr>
<tr>
<td>Precondicions</td>
<td></td>
</tr>
<tr>
<td>Postcondicions</td>
<td>El sistema genera les recomanacions i les emmagatzema.</td>
</tr>
</tbody>
</table>

**Taula 13. Contracte Generar recomanacions**
2.6 Model Conceptual

2.6.1 Diagrama de classes

A continuació podem observar el diagrama de classes.

Il·lustració 20. Diagrama de classes

2.6.2 Atributs de les classes

A continuació s’expliquen les classes amb els seus atributs.

Classe Notícia

Classe que representa les notícies del sistema.

<table>
<thead>
<tr>
<th>Notícia</th>
</tr>
</thead>
<tbody>
<tr>
<td>id: integer</td>
</tr>
<tr>
<td>titol: String</td>
</tr>
<tr>
<td>text: String</td>
</tr>
</tbody>
</table>

**Id**: Identificador de la notícia.

**Titol**: Títol de la notícia

**Text**: Contingut textual de la notícia
Classe Freqüència

Classe que representa la freqüència d’una paraula en una notícia.

<table>
<thead>
<tr>
<th>Freqüència</th>
</tr>
</thead>
<tbody>
<tr>
<td>paraula : String</td>
</tr>
<tr>
<td>aparicions : Double</td>
</tr>
</tbody>
</table>

**Paraula**: Paraula normalitzada de la notícia  

**Aparicions**: Nombre d’ocurrències de la paraula a la notícia

Classe Clúster

Classe que representa el clústering jeràrquic. El nombre màxim de notícies per clúster és configurat en el sistema.

<table>
<thead>
<tr>
<th>Clúster</th>
</tr>
</thead>
<tbody>
<tr>
<td>nom : String</td>
</tr>
<tr>
<td>id_centroid_esq : Integer</td>
</tr>
<tr>
<td>name_cluster_esq : String</td>
</tr>
<tr>
<td>id_centroid_dret : Integer</td>
</tr>
<tr>
<td>name_cluster_dret : String</td>
</tr>
</tbody>
</table>

**Nom**: Nom del clúster

**Id_centroid_esq**: Si el clúster esta dividit(Un clúster es divideix quan supera el màxim de notícies permesos), aquest valor serà l’id del centroide del clúster esquerra.

**Nom_cluster_esq**: Si el clúster esta dividit, aquest valor serà el nom del clúster esquerra.

**Id_centroid_dret**: Si el clúster esta dividit, aquest valor serà l’id del centroide del clúster dret.

**Nom_cluster_dret**: Si el clúster esta dividit, aquest valor serà el nom del clúster dret.

Restriccions d'integritat textuais:

1- Clau primària: (Notícia, id), (Freqüència, paraula+aparicions), (Clúster, nom).

2- Una notícia només pot contenir una freqüència amb la mateixa paraula.

3- El nombre màxim de recomanacions per una notícia esta configurat pel sistema.

4- El nombre de notícies per clúster esta configurat pel sistema.
3 Disseny

A l’etapa de disseny es presenta l’arquitectura del sistema, adaptant l’especificació de l’apartat anterior a la tecnologia específica que s’utilitzarà. D’aquesta manera, la documentació queda enllestida per a passar a l’etapa d’implementació.

3.1 Patró de disseny

Tal com hem vist en els capítols anteriors, el nostre sistema té clarament la capa de presentació, domini i dades separades. Per això, s’ha elegit el patró de disseny en capes.

Aquest patró té com a avantatge que facilita la canviabilitat que és un dels nostres objectius del projecte. A continuació es resumeix que ha de complir un software que apliqui el patró en capes:

- Els components s’agrupen en capes
- La comunicació solament es produeix entre elements de la mateixa capa o de capes contigües.

Il·lustració 21. Patró de disseny
La capa de presentació sap com presentar les dades a l’usuari, però ignora quines transformacions cal fer per donar resposta a les peticions de l’usuari.

La capa del domini sap com satisfar les peticions de l’usuari, però ignora on es guarden les dades i com es presenten a l’usuari.

La capa de gestió de dades sap on i com estan emmagatzemades les dades, però ignora com tractar-les.

Amb aquesta estructura s’aconsegueix que qualsevol canvi només afecti a una capa.

En el nostre sistema s’agrupen els components de la següent manera a les capes:

**Il·lustració 22. Patró de disseny del nostre sistema**
3.1.1 Capa de Presentació

Com ja s’ha explicat anteriorment, la capa de presentació és la interfície gràfica amb la que interactua l’usuari. Les vistes es construeixen en format web (html, javascript i D3js). Aquestes vistes interactuen amb l’usuari per inserir una notícia o visualitzar recomanacions.

Cal remarcar que per inserir les 20786 notícies, s’ha utilitzat el protocol de inserció massiva de Redis, generant uns scripts per parcejar-les i inserir-les.

Tot i tenir una interfície gràfica, cal destacar, que Lupa de manera similar a Freeling és un sistema software enfocat a ser integrat amb altres sistemes (pàgines web, aplicacions, etc.). Per tant, la capa de presentació de cara a usuaris experts seria via comandes per poder-la integrar amb els seus sistemes de recomanació.

Per exemple en el cas de TVC un tècnic s’encarregarà de gestionar les entrades de notícies amb els seus identificadors i d’integrar la visualització de la pàgina de TVC amb les recomanacions de Lupa guardades a la base de dades.

3.1.2 Capa de domini

En aquesta capa tenim dos scripts en Python (P1 i P2 en el diagrama) que són invocats quan algun usuari vol inserir una notícia o visualitzar alguna recomanació. Aquests scripts s’encarreguen de validar les dades introduïdes a la capa de presentació i enviar-les a la capa de dades per poder inserir la notícia o obtenir les recomanacions del identificador específicat.

A la capa de domini, també tenim tots els nodes de Storm que impliquen càlculs o lògica del sistema.

3.1.3 Capa de gestió de dades

A l’hora de traslladar el model conceptual a base de dades, un dels punts importants a decidir és com mapejar les dades.

Primer de tot, quan inserim una nova notícia (toRedis) s’haurà de crear un hash on la clau serà el propi identificador de la notícia. El hash quedaria de la següent manera:

<table>
<thead>
<tr>
<th>HASH(ID)</th>
<th>ID</th>
<th>TITOL</th>
<th>TEXT(MAX-1KB)</th>
</tr>
</thead>
</table>

El text només guarda els primers 1024 bytes de la notícia ja que les notícies de TVC estan guardades en altres servidors. Aquesta mostra de 1KB s’utilitza per veure el contingut de la notícia a la interfície de testeig.

A l’inserir una notícia a més a més, haurem d’afegir l’identificador a una cua de notícies pendents, perquè la topologia de Storm (Spout-Storm) sapiga que l’ha de processar. També es manté una llista actualitzada de totes les notícies processades.
En segon lloc, es guarda el clústering jeràrquic (Bolt-Storm) de manera que cada clúster serà un hash i els seus atributs seran els mateixos que en el model conceptual. On cada clúster si s’ha dividit, el nom del clúster esquerra i dret seran apuntadors als noms dels clústers respectius. De manera, que podrem recorre fàcilment l’estructura com un arbre binari.

<table>
<thead>
<tr>
<th>HASH(NAME)</th>
<th>NAME</th>
<th>ID_CENTROIDE_ESQ</th>
<th>NAME_CLUSTERS_ESQ</th>
<th>ID_CENTROIDE_DRET</th>
<th>NAME_CLUSTERS_DRET</th>
</tr>
</thead>
</table>

Com podem observar, "NAME_CLUSTERS_ESQ" i "NAME_CLUSTERS_DRET" apunten a un altre hash on "NAME" contingui aquest valor respectivament.

Les notícies de cada clúster s’han guardat utilitzant una llista utilitzant l’identificador del centroide i el nivell de l’arbre binari. Això ens assegura que no es repetiran ja que una notícia no pot ser centroide de més d’un clúster al mateix nivell de l’arbre binari.

Les recomanacions són guardades (Bolt-Storm) en una llista ordenada en funció de la distància entre notícies (per defecte, es guarden només cinc recomanacions).

<table>
<thead>
<tr>
<th>LIST(RECO_ID)</th>
<th>Reco 1 - distància</th>
<th>Reco 2 - distància</th>
<th>.........................</th>
<th>Reco n - distància</th>
</tr>
</thead>
</table>

Finalment, per visualitzar una notícia s’executa un script en java (toJson) el qual parceja les recomanacions i el hash corresponent a l’identificador desitjat, retornant uns JSON que utilitzarà la capa de presentació.

### 3.2 Diagrames de seqüència

A continuació es mostren els diagrames de seqüència de disseny del sistema. Cada diagrama on l’actor és l’usuari comença quan l’usuari realitza alguna acció no necessàriament a la interfície gràfica, ja que com hem mencionat la interfície està centrada en el testeig del software.

#### 3.2.1 Inserir notícia

Per a inserir un contingut primer de tot cal comprovar que les dades siguin correctes. Si ho són, intentem crear un contingut textual al servidor de Redis. Els identificadors dels continguts són únics, així que abans de crear-lo haurem de comprovar que no existeix ja un contingut amb aquest identificador.
3.2.2 Mostrar recomanacions

Primer de tot cal comprovar que el identificador existeix. Un cop validat, ja podem obtenir les recomanacions de la base de dades Redis.
3.2.3 Generar recomanacions

Per generar recomanacions el sistema està a l'espera de que un usuari insereixi una notícia a la llista no bloquejant “text_queue”. Un cop detecta que s'ha inserit correctament una notícia procedeix a tractar-la generant les seves recomanacions i actualitzant les recomanacions si és necessari d'alguna notícia relacionada.
Il·lustració 25. Diagrama de seqüència de disseny mostrar recomanacions
4 Implementació

A l’etapa d’implementació es codifica el sistema, obtenint així un sistema de recomanació utilitzable per a l’usuari. Aquest capítol no està dedicat a explicar detalladament la implementació del sistema, sinó que únicament s’explicaran els aspectes més rellevants.

4.1 Aspectes importants

4.1.1 Importància de la claredat del codi

Per començar, independentment de com s’implementi el sistema i del llenguatge de programació utilitzat, s’ha de destacar la importància de la claredat i netedat del codi de programació, especialment per complir un dels nostres objectius:

- Programar Lupa de manera adequada per facilitar possibles ampliacions o contribucions futures.

4.1.2 Clústering Jeràrquic

La idea principal d’aplicar clústering jeràrquic al nostre projecte consisteix en disminuir l’augment del temps de resposta en relació a l’augment de les dades. Per millorar aquesta relació tenim dos solucions no excloents:

1. Implementar el sistema de manera que puguem afegir-hi més capacitat de càlcul reduint el temps de resposta.
2. Aplicar tècniques per reduir l’impacte de l’augment de dades.

La primera solució ja està aplicada mitjançant Storm i el seu desplegaments al Cloud. Això que per assolir millor els nostres objectius temporalss aplicarem clústering jeràrquic a la nostre topologia, de manera que reduirren el nombre de comparacions que haurà de realitzar una notícia al ser inserida al sistema així reduint el temps de càlcul.

Concretament, implementarem una versió de clústering jeràrquic divisiu [6][Error! No se encuentra el origen de la referencia.] aplicant l’algoritme k-means [7][Error! No se encuentra el origen de la referencia.] adaptat al nostre problema com s’explica al següent capítol.

4.1.2.1 Clústering Jeràrquic

En el nostre cas, el clúster jeràrquic ha sigut implementat mitjançant una arbre binari on cada node és un clúster que té un mida màxim de notícies, un cop assolit el màxim s’aplica l’algoritme de k-means (k = 2) per dividir les notícies del clúster en dos nous clústers que seran els dos successors del node al arbre binari. A continuació tenim un exemple visual amb una mida màxim per clúster de 4 notícies:
1. Inicialment tenim un únic clúster vuit:

Il·lustració 26. Clústering Jeràrquic 1

2. Anem afegint identificadors de les notícies que rebem m’entre les que no superem el límit màxim de notícies en el clúster:

Il·lustració 27. Clústering Jeràrquic 2

3. Un cop arribat al màxim nombre de notícies apliquem l'algorisme k-means, el qual separarà les notícies en dos grups \((k = 2)\). Suposarem que obtenim els següents centroides amb els seus clústers \(1\rightarrow 1,2\) i \(3\rightarrow 3,4\). Per centroïde, entenem a la notícia representativa del clúster.

Il·lustració 28. Clústering Jeràrquic 3

4. Tot seguit, afegim una notícia \((id = 5)\), aquesta notícia es compara amb els dos centroides \((1\ i\ 3)\) i el que sigui més similar s’afegirà al seu clúster (Suposem que és més similar a la notícia 1). Cal remarcar que les comparacions en el procés de clústering aplicant el mateix algoritme de similitud entre notícies.

Il·lustració 29. Clústering Jeràrquic 4

5. Ara per acabar d’il·lustrar el funcionament de la nostre implementació del clústering jeràrquic podem veure com inserint una notícia més \((id = 6, més\ similar\ a\ la\ notícia\ 1)\) separem el clúster.
Podem calcular fàcilment el nombre de comparacions en un cas mig que haurà de fer una notícia que l'afeigm en un clúster generat al llarg del temps de 1.000.000 de notícies. Al ser un arbre binari tenim que l'alçada en un cas mig serà:

\[ \log_2 1000000 = 19.93 \approx 20 \]

Així que el cost mig per trobar el seu clúster seran 20 comparacions. Per obtenir la recomanació necessitarem a més a més de les 20 comparacions les pròpies del clúster. Si fixem els clústers per exemple a 1000 notícies tindrem el resultat en 1020 comparacions. En canvi, sense clústering jeràrquic necessitaríem realitzar 1.000.000 de comparacions.

Per tant, veiem com amb aquesta estructura estem disminuït l'augment del cost temporal en relació amb l'augment de les dades. Per exemple el mateix càlcul amb 2 milions de notícies obtindriem un cost mig de 1021 comparacions. És a dir, augmentem 1 comparació enlloc de 1 milió com seria en el cas inicial. En l'apartat d' experimentació veurem resultats reals d'aplicar les dues tècniques.

Cal remarcar, que a canvi d'aquest decrement de comparacions, tenim el cost de gestionar el clúster i aplicar l'algoritme k-means cada vegada que hem de separar un clúster, però com veurem en l'apartat d'experimentació el cost es insignificant en front amb el de les comparacions.
4.1.3 Topologia

Com he mencionat anteriorment, el nostre sistema disposa de dues topologies una que aplica clústering jeràrquic i una altre que no. Tot seguit, s’explica pas per pas l’implementació de cada spout i bolts de les dues topologies.

4.1.3.1 Topologia sense clústering jeràrquic

Aquesta topologia està pensada per casos on no tinguem un conjunt de dades molt gran o no sigui un problema el temps de resposta del sistema. L’avantatge que presenta sobre el clústering jeràrquic és que es realitzen totes les comparacions possibles, per tant, el resultat de la recomanació serà el millor possible en funció de l’algoritme de similitud entre notícies.

Tot seguit tenim un esquema global de la topologia i una breu explicació de cada node:

- **TextRedisSpout**: Aquest spout s’encarrega de llegir les notícies d’una llista no bloquejant de Redis. Cada cop que detecta una nova notícia l’envia al FreelingBolt.

- **FreelingBolt**: El FreelingBolt com el seu nom indica, envia el contingut de la notícia rebuda del TextRedisSpout a un servidor Freeling, espera el retorn i envia el text normalitzat al CalcProbBolt.

- **CalcProbBolt**: Aquest Bolt transforma la notícia rebuda a una distribució de paraules amb la seva freqüència i la guarda a Redis. Finalment envia al DispatcherBolt la distribució de paraules de la notícia.

- **DispatcherBolt**: El DispatcherBolt envia al CompareTextBolt una tupla per cada notícia processada prèviament amb la rebuda del CalcProbBolt. És a dir, si el Dispatcher tenia enregistrades n notícies, enviarà n tuples.

- **CompareTextBolt**: Rep la distribució de paraules de la notícia i un identificador per comparar. Primer de tot, recupera de Redis la distribució de paraules de l’identificador de la segona notícia i procedeix a calcular la similitud. Un cop té la similitud la guarda a Redis si és necessari. Per necessari, s’entén que la notícia comparada té un millor resultat que alguna de les recomanacions ja establertes. Al sistema es pot configurar el nombre màxim de recomanacions que vulguem que es guardin per notícia.
4.1.3.1 Topologia amb clústering jeràrquic

La base d'aquesta topologia és la explicada en el capítol anterior, la principal diferència és que a l'hora de buscar les notícies amb les que hem de comparar apliquem el clústering jeràrquic i no comparem amb totes les possibles.

Al següent diagrama podem veure un esquema de la topologia:

A continuació s’explica la lògica del SearchClusterNodeBolt i el DispatcherClusterBolt, els altres nodes ja han estat explicats en el capítol anterior.

- **SearchClusterNodeBolt**: Mitjançant la distribució de paraules de la notícia que rep del CalcProbBolt busca el node del clústering jeràrquic que li correspon i l’afegeix. L’estructura del clústering jeràrquic està guardada a Redis, per tant, en aquest node és realitzen moltes consultes a Redis. En cas, que a l’afegir s’hagi de dividir el clúster es realitza en aquest mateix node. Finalment envia al DispatcherClusterBolt el nom del node i la distribució de paraules de la notícia.

- **DispatcherClusterBolt**: A partir del nom del clúster rebut, llegirà de Redis la llista de identificadors corresponents al clúster i s’envia una tupla per cada identificador i la distribució de paraules de la notícia nova.

4.1.4 Procés de recomanació

Partint del software de recomanació seleccionat en capítols anteriors (Lenskit), hem fet un diagrama global de com funciona Lenskit obtenint el següent:
En el nostre cas, no tenim activitat d'usuari ja que volem comparar les notícies basant-nos només amb el seu contingut. La matriu de similitud que obté Lenskit pot ser originada per comparació directe de ítems, o pot ser generada per l'activitat d'usuari donant a diferents tipus de recomanacions (item-item, user-user o un hybrid). Nosaltres, clarament volem una recomanació item-item ja que comparem la similitud de les notícies a través de les paraules.

Tampoc necessitem guardar una matriu de resultats del mida (itemxitem) ja que per cada item només caldrà que guardem les X millors recomanacions.

Així que, finalment he implementat tot el procés de recomanació. Bàsicament a partir de l'algoritme de similitud en el CompareTextBolt, calculem la distància i si es menor a una de les ja assignades eliminem la pitjor recomanació o si encara no hi havia el màxim de recomanacions l'assignem directament.

### 4.2 Desplegament en un clúster remot

Per poder utilitzar el nostre software en un clúster remot, ja sigui a Amazon o en un clúster de màquines pròpies s'han d'automatitzar les tasques repetitives. Com a tasques repetitives s'entén la instal·lació/configuració dels subsistemes (Storm, Redis i Freeling) i la configuració dels permisos de les màquines.

Per desplegar màquines a Amazon amb Storm i les seves dependencies ho podem fer fàcilment mitjançant el projecte Storm-deploy. Storm-deploy és un projecte creat per el propietari de Storm que ens permet desplegar les màquines que vulguem amb la versió de Storm que vulguem a Amazon de forma molt ràpida i senzilla d’aprendre.

Per altre banda, tenim les dependències del nostre projecte Redis i Freeling. Per facilitar aquesta tasca hem creat uns scripts que s’encarreguen de les seves instal·lacions i configuració. En el cas de Freeling apliquen directament el patch que hem fet abans de l’instal·lació. Aquest patch com es veurà en el següent capítol aplica una millora d’eficiència a Freeling.

Finament, el mateix script s’encarrega d’obrir els ports perquè les màquines d’Amazon tinguin visibilitat entre elles en els ports de Redis i Freeling.
4.3 Principals problemes trobats

Durant la implementació del sistema han anat apareixent alguns problemes que no s'havien previst inicialment i que han obligat a realitzar alguns canvis en la implementació. A continuació es llisten els més importants.

4.3.1 Implementació pròpia recomanador

Finalment, tot i realitzar l’estudi de recomanadors i seleccionar Lenskit, s'ha implementat totalment la funcionalitat del recomanador des de la lògica de recomanació (algoritme de similitud) fins l'estructura de dades en que es guarda. Esperàvem poder aprofitar Lenskit per alleugerir la càrrega d'alguna part.

4.3.2 Implementació pròpia Clústering Jeràrquic

Quant va sortir la idea d’aplicar Clústering Jeràrquic la primera idea era aprofitar alguna llibreria que ens facilités l’ús del clústering jeràrquic o l’algoritme k-means.

Estudiades algunes possibilitats de softwares lliures que implementen aquests mètodes, es va decidir implementar una versió pròpia ja que resultava menys esforç que adaptar algun software lliure a les necessitats del projecte.

També cal destacar que a l’implementar el clústering des de zero, està modulat de tal manera que tant l’algoritme de similitud com k-means, podrien ser modificats o simplement substituïts per altres algoritmes de forma senzilla.

4.3.3 Connexió Freeling

Testejant Freeling amb varies entrades ens vam adonar, que no retorna el resultats fins que no tanques la connexió amb el servidor. Això va ser un gran inconvenient, ja que si cada cop que volem processar una notícia hem d’obrir i tancar la connexió amb un servidor Freeling, resultarà el coll d’ampolla del sistema. Aquest fet ens podria haver perjudicat amb els nostres objectius de temps de resposta del sistema.

Un cop detectat el problema, es va procedir a investigar si es podia solucionar modificant el propi Freeling, ja que és Open Source i es té accés al codi.

S’ha realitzat un patch que modifica el client del Freeling de manera que cada cop que acaba una notícia envia el resultat i comença a processar la següent en la cua d’entrada.
4.3.4 Problemes amb la visualització dels resultats

Inicialment la visualització dels resultats era fàcil degut a que els tests es feien amb poques dades, ja que l'objectiu inicial era testejar el correcte funcionament de la topologia pas per pas. Però un cop ha estat implementada del tot, s'han realitzat tests amb moltes dades (> 20.000) cosa que feia difícil analitzar els resultats directament de la base de dades.

Per tant, s'ha optat per crear una visualització on podem veure el clúster jeràrquic i podem consultar les recomanacions de cada notícia. Tot seguit, es mostrà un exemple de visualització:

![Il·lustració 34. Exemple visualització clústering jeràrquic](image)

En la imatge anterior, podem veure el clúster jeràrquic on els nodes intermitjos són els centroides i les fulles són els identificadors de les notícies del clúster, representat pel centreide més pròxim.

4.4 Proves i correcció d’errors

Per tal de garantir que el sistema funciona correctament s’han utilitzat dues metodologies. Primer de tot, com és habitual en tota implementació, s'han anat provant tots els escenaris possibles d'una funcionalitat cada cop que s'acabava d'implementar per tal de corregir-ne els errors, ja que sol ser freqüent que n'apareguin i no és recomanable seguir implementant sense saber si el que s'ha fet fins ara està bé o no.

Un cop finalitzada la implementació de tot el sistema s'ha tornat a provar l'aplicació sencera simulant diversos escenaris per tal de corregir errors que s'hagin pogut passar per alt o que vinguin derivats de funcionalitats implementades posteriorment a la tractada.

Per simular escenaris s'han utilitzat fraccions de llibres procedents del Projecte Gutenberg [Error! No se encuentra el origen de la referencia.].
El Projecte Gutenberg és un recull de llibres electrònics lliures, aquests llibres s’han aprofitat per veure si introduint diversos llibres de temàtiques diferents és veia plasmat al clúster jeràrquic generat. Mitjançant aquests recursos s’han creat tests unitaris de recomanació i clústering jeràrquic.

Per poder generar aquests tests i analitzar els resultats, com hem explicat anteriorment, ha sigut necessari crear una visualització per poder representar el clúster generat, ja que no és fàcil interpretar directament de la base de dades quan es tracta d’un conjunt de dades no trivial.
5 Experimentació

A l’etapa d’experimentació s’ha realitzat diverses proves enfocades al espai, al temps de resposta del sistema i a la recomanació.

5.1 Experimentació espai

Tot i que la major part de dades a emmagatzemar son les pròpies notícies que ja estan guardades en altres sistemes de TVC. S’ha analitzat l’espai que ocupen les estructures que crea Lupa en comparació a les notícies. A la següent taula es mostren els resultats:

<table>
<thead>
<tr>
<th>Notícies TVC</th>
<th>Estrucutres Lupa</th>
</tr>
</thead>
<tbody>
<tr>
<td>Espai (MB)</td>
<td>300</td>
</tr>
<tr>
<td></td>
<td>50</td>
</tr>
</tbody>
</table>

Taula 14. Cost espai notícies versus estructures

Com es veu, el cost de les estructures es bastant menor al de les notícies. Així que no resulta un problema el cost en espai de Lupa. Durant tota l’etapa d’implementació s’ha intentat que les estructures de Lupa guardessin només les dades indispensables per evitar problemes d’espai.

5.2 Experimentació temps de resposta

S’han definit dues mesures per analitzar el temps de resposta del sistema. La primera d’elles consisteix en el temps per processar les 20786 notícies, i la segona consisteix en el temps per processar una nova inserció amb les 20786 ja calculades.

Les dues mesures seran avaluades en els següents escenaris:

- Escenari 1: S’utilitza la topologia sense clústering jeràrquic en un entorn local de 1 màquina
- Escenari 2: S’utilitza la topologia amb clústering jeràrquic en un entorn local de 1 màquina
- Escenari 3: S’utilitza la topologia amb clústering jeràrquic en l’entorn remot d’Amazon explicat en l’apartat sobre la decisió de l’arquitectura.

5.2.1 Temps d’execució total

Aquesta mesura consisteix en el temps que es tardà des de que s’introdueix la primera notícia fins que obtenim totes les recomanacions de la última. El conjunt de dades són les 20786 notícies. A continuació es mostren els resultats dels tres escenaris:

<table>
<thead>
<tr>
<th>Escenari 1</th>
<th>Escenari 2</th>
<th>Escenari 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temps</td>
<td>2 dies 22hores 30 min</td>
<td>1hora</td>
</tr>
</tbody>
</table>

Taula 15. Temps d’execució total dels escenaris

Com podem veure a la taula el temps sense clústering jeràrquic és molt elevat, això es degut a que es realitzen unes 20768*20768/2 comparacions que són aproximadament uns 215 milions de comparacions. En canvi, aplicant clústering sense comptar les
comparacions necessàries per separar clúster quan sigui necessari cada notícia farà les següents comparacions en mitjana:

1. Per recorre l’arbre binari:

\[ \log_2 20768 = 14,34 \approx 14 \]

2. Dintre del propi clúster com a màxim farà 30 comparacions ja que es la mida màxima.

Per tant tindrem aproximadament unes 44 comparacions si el clúster està a la seva màxima capacitat. En canvi, sense clústering la notícia s’ha de comparar amb totes les que havien estat processades prèviament.

5.2.2 Temps d’execució per inserció

En aquest apartat, s’ha mesurat el temps que tardem en processar la inserció d’una nova notícia tenint el conjunt de 20786 notícies ja inserides. A la següent taula tenim els resultats d’aquest experiment:

<table>
<thead>
<tr>
<th>Escenari 1</th>
<th>Escenari 2</th>
<th>Escenari 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temps</td>
<td>2 minuts</td>
<td>9 segons</td>
</tr>
</tbody>
</table>

Taula 16. Temps d’inserció dels escenaris

Com podem veure el temps sense clústering és mes gran ja que ha de realitzar 20786 comparacions. En canvi, amb clústering ha de comparar aproximadament amb 44.

Cal remarcar, que en l’escenari 1 el temps de resposta per inserció escalarà en funció de les notícies existents. Però en l’escenari 2 i 3 l’escalabilitat no depèndrà tant de les notícies existents. Com s’ha demostrat en l’apartat d’implementació del clústering jeràrquic per inserir una notícia en un conjunt de notícies de 1 milió necessitem 20 comparacions per recorre l’arbre binari, i si incrementem el conjunt de notícies a 2 milions només necessitem una comparació més. En canvi, si aplicuem aquests exemple a l’escenari 1 necessitarem 1 milió de comparacions més.

5.3 Experimentació recomanació

L’objectiu d’aquest apartat consisteix en analitzar els resultats de Lupa i veure en quin grau poden ser útils. Per realitzar aquest anàlisis hem optat per dos fonts de dades diferents:

- Projecte Gutenberg
- Notícies TVC

5.3.1 Projecte Gutenberg

Com s’ha explicat en l’apartat de prova i correcció d’errors, utilitzarem diversos llibres del projecte Gutenberg per realitzar diversos experiment. Cal destacar, que els llibres seleccionats són en anglès però el nostre sistema i Freeling poden treballar amb diverses llengües.
S'han realitzat tres experiments diferents resumits a continuació:

### 5.3.1.1 Experiment Projecte Gutenberg (I)

Aquest experiment consisteix en introduir al sistema 3 capítols de dos llibres bastant diferenciats en temàtica i èpoques:

- Adventures of Huckleberry Finn by Mark Twain
- The Major Operations of the Navies in the War of American Independence by Mahan

El conjunt de dades d'aquest experiment és el següent (les cases corresponen al ID associat al fragment):

<table>
<thead>
<tr>
<th>Cap. 1</th>
<th>Cap.2</th>
<th>Cap.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Llibre 1</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>Llibre 2</td>
<td>21</td>
<td>22</td>
</tr>
</tbody>
</table>

#### Taula 17. Conjunt de dades experiment Gutenberg I

La mida màxima del clúster en aquest experiment ha estat fixat a 3. Així doncs després d’afegir aquests fragments el resultat de la recomanació pel fragment 11 és el següent:

<table>
<thead>
<tr>
<th>ID</th>
<th>Similitud</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>1.301</td>
</tr>
<tr>
<td>12</td>
<td>1.395</td>
</tr>
<tr>
<td>23</td>
<td>2.950</td>
</tr>
<tr>
<td>22</td>
<td>3.314</td>
</tr>
<tr>
<td>21</td>
<td>3.423</td>
</tr>
</tbody>
</table>

#### Taula 18. Resultat experiment Gutenberg I

Com podem veure per el capítol 1 del llibre 1 les millors recomanacions que el nostre sistema retorna són els capítols 2,3 del mateix llibre, en canvi, els capítols del llibre 2 és troben a més del doble de distància.

### 5.3.1.2 Experiment Projecte Gutenberg (II)

Aquest experiment consisteix en introduir al sistema 3 capítols de tres llibres, on el 1 i el 3 són del mateix autor i temàtica, en canvi, el 2 llibre és totalment diferent.

- Adventures of Huckleberry Finn by Mark Twain
- The Major Operations of the Navies in the War of American Independence by Mahan
- The Adventures of Tom Sawyer by Mark Twain

El conjunt de dades d’aquest experiment és el següent (les cases corresponen al ID associat al fragment):
La mida màxima del clúster en aquest experiment ha estat fixat a 3. Un cop el sistema ha processat totes aquests fragments obtenim el següent resultat a nivell de clústering jeràrquic:

<table>
<thead>
<tr>
<th>Llibre 1</th>
<th>Cap. 1</th>
<th>Cap. 2</th>
<th>Cap. 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>11</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>Llibre 2</td>
<td>21</td>
<td>22</td>
<td>23</td>
</tr>
<tr>
<td>Llibre 3</td>
<td>31</td>
<td>32</td>
<td>33</td>
</tr>
</tbody>
</table>

Taula 19. Conjunt de dades experiment Gutenberg II

Podem veure com ha separat clarament els capítols de cada llibre dels altres, a més a més, el capítols dels llibres de Mark Twain (1 i 3) els ha agrupat a la mateixa part de l’arbre binari.

A la taula següent tenim els resultats de recomanació del capítol amb id 11.

<table>
<thead>
<tr>
<th>ID</th>
<th>Similitud</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>1.301</td>
</tr>
<tr>
<td>12</td>
<td>1.395</td>
</tr>
<tr>
<td>31</td>
<td>2.012</td>
</tr>
<tr>
<td>33</td>
<td>2.051</td>
</tr>
<tr>
<td>32</td>
<td>2.078</td>
</tr>
<tr>
<td>23</td>
<td>2.950</td>
</tr>
<tr>
<td>22</td>
<td>3.314</td>
</tr>
<tr>
<td>21</td>
<td>3.423</td>
</tr>
</tbody>
</table>

Taula 20. Resultat experiment Gutenberg II
Com podem veure per el capítol 1 del llibre 1 les millors recomanacions que el nostre sistema retorna són els capítols 2,3 del mateix llibre, a més a més, els capítols del llibre similar obtenen millors puntuacions que el llibre diferenciat.

5.3.1.2 Experiment Project Gutenberg (III)

Aquest experiment parteix de la base dels llibres del experiment previ, però en aquest experiment no el limitarem el conjunt de dades als tres primers capítols sinó que dividirem cada llibre de forma automàtica en parts de 10Kb (aproximadament 10.000 lletres) buscant el punt final més pròxim.

El conjunt de dades d’aquest experiment és el següent:

<table>
<thead>
<tr>
<th>Parts</th>
<th>ID inicial</th>
<th>ID final</th>
</tr>
</thead>
<tbody>
<tr>
<td>Llibre 1</td>
<td>56</td>
<td>100</td>
</tr>
<tr>
<td>Llibre 2</td>
<td>49</td>
<td>200</td>
</tr>
<tr>
<td>Llibre 3</td>
<td>33</td>
<td>300</td>
</tr>
</tbody>
</table>

Taula 21. Conjunt de dades experiment gutenberg I

La mida màxima del clúster en aquest experiment ha estat fixat a 30. Un cop el sistema ha processat totes aquests fragments obtenim el següent resultat a nivell de clústering jeràrquic:

Podem veure com ha separat clarament els fragments de cada llibre dels altres, a més a més, els fragments dels llibres de Mark Twain (1 i 3) els ha agrupat a la mateixa part de l’àrbre binari. No obstant, en aquest experiment apareixen alguns casos on algun
fragment no es troba amb la resta del mateix llibre, això pot ser degut al clústering ja que evitem comparacions, a canvi, de precisió. També pot ser degut a que la seva distribució de paraules es mes similar al altre clúster, per tant, podria estar bé.

A la taula següent tenim els resultats de recomanació del capítol amb id 101.

<table>
<thead>
<tr>
<th>ID</th>
<th>Similitud</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>2.223</td>
</tr>
<tr>
<td>140</td>
<td>2.376</td>
</tr>
<tr>
<td>153</td>
<td>2.390</td>
</tr>
<tr>
<td>111</td>
<td>2.469</td>
</tr>
<tr>
<td>102</td>
<td>2.488</td>
</tr>
</tbody>
</table>

Taula 22. Resultat experiment gutenberg III

Com podem veure per el capítol 1 del llibre 1 les millors recomanacions que el nostre sistema retorna són els capítols 2,3 del mateix llibre, a més a més, els capítols del llibre similar obtenen millors puntuacions que el llibre diferenciat.

En conclusió, veient el resultat del clúster jeràrquic podem ser optimistes amb el resultat, ja que és un cas bastant ideal.
5.3.2 Notícies TVC

Per aquest escenari s'han utilitzat notícies provinents de TVC. En concret, s'han analitzat 20786 notícies.

A continuació tenim un subconjunt de l'arbre generat:

Il·lustració 37. Subconjunt clústering jeràrquic experiment Notícies TVC

Com podem veure, a partir del clústering jeràrquic generat no podem decidir si la recomanació ha sigut bona, degut a que els identificadors no estan relacionades entre elles com en els casos anteriors i l'arbre és immens.

Per analitzar si la recomanació és acceptable s'utilitzen dues tècniques. La primera consisteix en seleccionar una mostra aleatòria i decidir manualment si les recomanacions són correctes. S'ha agafat un subconjunt de 10 notícies aleatòries i s'ha avaluat manualment si la recomanació era correcta o no i s'ha obtingut el següent resultat:

<table>
<thead>
<tr>
<th>ID</th>
<th>Recomanció correcta?</th>
</tr>
</thead>
<tbody>
<tr>
<td>2115453</td>
<td>Sí</td>
</tr>
<tr>
<td>2041938</td>
<td>Sí</td>
</tr>
<tr>
<td>2029514</td>
<td>Sí</td>
</tr>
<tr>
<td>2132613</td>
<td>No</td>
</tr>
<tr>
<td>2149617</td>
<td>Sí</td>
</tr>
<tr>
<td>2159793</td>
<td>Sí</td>
</tr>
<tr>
<td>2169173</td>
<td>No</td>
</tr>
<tr>
<td>2058613</td>
<td>Sí</td>
</tr>
<tr>
<td>2067593</td>
<td>Sí</td>
</tr>
<tr>
<td>2152856</td>
<td>Sí</td>
</tr>
</tbody>
</table>

Taula 23. Recomanacions mostra aleatòria notícies TVC
Il·lustració 38. Recomanació identificador 2115453

Il·lustració 39. Recomanació identificador 2041938
En aquesta avaluació obtenim un 80% de recomanacions correctes (a l'annex estan presents les notícies restants). Aquest primer indicador no és molt significatiu, però serveix per fer-nos una idea de com funciona Lupa.

L'altra opció, consisteix en una tècnica denominada “marcatges” la qual serveix per saber el grau d'interacció entre els usuaris i les recomanacions. Aquesta tècnica s'utilitzarà quan es desplegui Lupa en explotació a TVC.

La tècnica de marcatge es basa en enregistrar cada vegada que un usuari accedeixi a una recomanació i així poder avaluar el percentatge de vegades que els usuaris accedeixen a les recomanacions prestades. D'aquesta manera, amb un sol indicador podrem veure si hem millorat el nombre d'accessos a la recomanació en front del recomanador actual. Aquesta opció és molt més precisa que la primera i ens indicarà indirectament si les recomanacions son bones, ja que, l'objectiu de les recomanacions és que els usuaris les utilitzin.
6 Planificació i costos

En aquest capítol es presenta la planificació temporal inicial, la planificació temporal final, i l’anàlisi econòmica del projecte tenint en compte la majoria de factors que l’afecten.

6.1 Planificació temporal

6.1.1 Planificació inicial

El projecte l’hem dut a terme mitjançant una metodologia àgil, la qual és un mètode d’enginyeria del software basat en el desenvolupament iteratiu i incremental. Concretament hem utilitzat un marc de treball per a la gestió de projectes anomenat Scrum [9]Error! No se encuentra el origen de la referencia.]

Scrum segueix la següent metodologia per la gestió temporal d’un projecte:

- Desglossar els objectius en varies històries d’usuari.
- Valorar les històries d’usuari.
- Separar les diverses històries d’usuari en SPRINTs tenint en compte la dificultat i les precedències anteriors.

Per tant, el primer que hem fet és desglossar els objectius en vàries històries d’usuari (QUI-QUÈ-PERQUÈ):

**Objectiu:** Oferir als redactors i als usuaris finals una recomanació més acotada que la actual mitjançant la distribució de les paraules en una notícia, per exemple, si estan redactant/llegint una notícia relacionada amb la Política d’Ucraïna volem que recomani altres notícies, com per exemple, la situació social d’aquell país. Actualment només es recomanarien notícies amb el TAG Política, per tant, podríem estar recomanant notícies relacionades amb la Política Catalana que no serien rellevants pel cas.

**Històries d’usuari:**

- **LUPA** - Analitzar diverses opcions de personalització i recomanació de continguts - Establir la millor guia.
- **LUPA** - Investigació trident-ml KLD-Classifier – Establir una funció de similitud entre textos.
- **LUPA** - Experimentació software final en comparació de la recomanació inicial - Saber si hem obtingut un grau de millora en la recomanació.
- **LUPA** - Experimentació Freeling – Afinar el grau de similitud entre els textos
Objectiu: Programar el software per obtenir els resultats en temps real.

Històries d'usuari:

- LUPA - Investigació Storm - Per poder implementar un sistema distribuït per complir amb l'objectiu de temps real.
- LUPA - Experimentació Desplegament de storm en local – Testeig del nostre software en local.
- LUPA - Experimentació Desplegament de storm en Remot - Augmentar l'escalabilitat amb independència de recursos locals.
- LUPA - Implementació en local – Testeig del nostre software en local.
- LUPA - Implementació en remot – Augmentar l'escalabilitat amb independència de recursos locals del nostre software.
- LUPA - Experimentació Software final en local i desplegat al cloud d'EC2 - Calcular l'Speed-Up al desplegar al cloud.

Objectiu: Programar el software de manera que sigui contínuament escalable, ja que cada cop tindrem més noticies en el data set.

Històries d'usuari:

- LUPA - Investigació Redis – Per gestionar l'escalabilitat de data sets grans de forma eficient.
- LUPA - Experimentació Redis – Familiaritzar-nos amb les opcions de Redis.
- LUPA - Investigació serveis Amazon EC2 – Augmentar l'escalabilitat amb independència de recursos locals.
- LUPA - Experimentació desplegament clúster remot amazon EC2 – Comprovar que el nostre projecte serà viable.

A més a més, de les històries d'usuari que hem específicat anteriorment tenim un conjunt de històries més que no deriven directament dels objectius:

- LUPA - Planificació Temporal – Establir un marc de treball.
- LUPA - Definició del projecte – Acotar el projecte.
- LUPA - Especificació del software – Preparar els següents passos.
- LUPA - Disseny del software – Preparar els següents passos.
- LUPA - Documentació – Per documentar el projecte realitzat.
Les històries de documentació, especificació i disseny no són històries per si soles ja que al utilitzar una metodologia àgil no realitzarem aquestes tasques de forma independent sinó que les durem a terme de forma iterativa i incremental amb la resta de històries d’usuaris.

Un cop tenim totes les històries d’usuari el següent pas consisteix en valorar-les:

<table>
<thead>
<tr>
<th>#</th>
<th>Històries d’usuari</th>
<th>VALORACIÓ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Analitzar diverses opcions de personalització i recomanació de continguts</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>Investigació trident-ml KLD-Classifier</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>Experimentació software final en comparació de la recomanació inicial</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>Investigació Redis</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>Experimentació Redis</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>Experimentació Freeling</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>Investigació Storm</td>
<td>13</td>
</tr>
<tr>
<td>8</td>
<td>Experimentació Desplegament de storm en local</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>Experimentació Desplegament de storm en Remot</td>
<td>13</td>
</tr>
<tr>
<td>10</td>
<td>Implementació en Local</td>
<td>13</td>
</tr>
<tr>
<td>11</td>
<td>Implementació en Remot</td>
<td>13</td>
</tr>
<tr>
<td>12</td>
<td>Experimentació software final en local i desplegat al cloud d’EC2</td>
<td>13</td>
</tr>
<tr>
<td>13</td>
<td>Investigació Serveis Amazon EC2</td>
<td>13</td>
</tr>
<tr>
<td>14</td>
<td>Experimentació desplegament clúster remot amazon EC2</td>
<td>13</td>
</tr>
<tr>
<td>15</td>
<td>Planificació temporal</td>
<td>5</td>
</tr>
<tr>
<td>16</td>
<td>Definició del projecte</td>
<td>8</td>
</tr>
<tr>
<td>17</td>
<td>Especificació</td>
<td>8</td>
</tr>
<tr>
<td>18</td>
<td>Disseny</td>
<td>8</td>
</tr>
<tr>
<td>19</td>
<td>Documentació</td>
<td>13</td>
</tr>
</tbody>
</table>

Taula 24. Històries d’usuari valorades

Tot seguit podem donar una visió menys global d’algunes històries d’usuari, subdividint-les en històries més concretes:

<table>
<thead>
<tr>
<th>#</th>
<th>Històries d’usuari</th>
<th>VALORACIÓ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Analitzar diverses opcions de personalització i recomanació de continguts</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>Investigació trident-ml KLD-Classifier</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>Experimentació software final en comparació de la recomanació inicial</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>Investigació Redis</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>Experimentació Redis</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>Experimentació Freeling</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>Investigació Storm</td>
<td>13</td>
</tr>
<tr>
<td>8</td>
<td>Experimentació Desplegament de storm en local</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>Experimentació Desplegament de storm en Remot</td>
<td>13</td>
</tr>
<tr>
<td>10</td>
<td>Implementació en Local</td>
<td>13</td>
</tr>
<tr>
<td>11</td>
<td>Implementació en Remot</td>
<td>13</td>
</tr>
<tr>
<td>12</td>
<td>Experimentació software final en local i desplegat al cloud d’EC2</td>
<td>13</td>
</tr>
<tr>
<td>13</td>
<td>Investigació Serveis Amazon EC2</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>Històries d’usuari</td>
<td>VALORACIÓ</td>
</tr>
<tr>
<td>---</td>
<td>--------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>7</td>
<td>Investigació Storm</td>
<td>13</td>
</tr>
<tr>
<td>8</td>
<td>Experimentació Desplegament de storm en local</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>Experimentació Desplegament de storm en Remot</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>Investigació Redis</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>Experimentació Redis</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>Experimentació Freeling</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Investigació Serveis Amazon EC2</td>
<td>13</td>
</tr>
<tr>
<td>14</td>
<td>Experimentació desplegament clúster remot amazon EC2</td>
<td>13</td>
</tr>
<tr>
<td>1</td>
<td>Analitzar diverses opcions de personalització i recomanació de continguts</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>Investigació trident-ml KLD-Classifier</td>
<td>8</td>
</tr>
<tr>
<td>15</td>
<td>Planificació temporal</td>
<td>5</td>
</tr>
<tr>
<td>16</td>
<td>Definició del projecte</td>
<td>8</td>
</tr>
<tr>
<td>17</td>
<td>Especificació</td>
<td>8</td>
</tr>
</tbody>
</table>

Un cop tenim totes les històries d’usuari, procedim a ordenar-les evitant problemes de precedències:
Finalment hem d’assignar les històries d’usuari als diferents SPRINTS amb relació a l’ordre i la valoració. Els SPRINTS tenen una durada de 2 setmanes i una càrrega de 6h diàries.

<table>
<thead>
<tr>
<th>SPRINT</th>
<th>Històries d’usuari</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Investigació Storm</td>
</tr>
<tr>
<td>2</td>
<td>Experimentació Desplegament de storm en local</td>
</tr>
<tr>
<td>2</td>
<td>Experimentació Desplegament de storm en Remot</td>
</tr>
<tr>
<td>3</td>
<td>Investigació Redis</td>
</tr>
<tr>
<td>3</td>
<td>Experimentació Redis</td>
</tr>
<tr>
<td>3</td>
<td>Experimentació Freeling</td>
</tr>
<tr>
<td>3</td>
<td>Investigació Serveis Amazon EC2</td>
</tr>
<tr>
<td>4</td>
<td>Experimentació desplegament clúster remot amazon EC2</td>
</tr>
<tr>
<td>5</td>
<td>Analitzar diverses opcions de personalització i recomanació de continguts</td>
</tr>
<tr>
<td>5</td>
<td>Investigació trident-ml KLD-Classifier</td>
</tr>
<tr>
<td>6</td>
<td>Planificació temporal</td>
</tr>
<tr>
<td>6</td>
<td>Definició del project</td>
</tr>
<tr>
<td></td>
<td>Definició dels objectius</td>
</tr>
<tr>
<td></td>
<td>Planificació de costos</td>
</tr>
</tbody>
</table>

Taula 26. Històries d’usuari ordenades

<table>
<thead>
<tr>
<th>SPRINT</th>
<th>Període</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/1/2014-10/1/2014</td>
</tr>
<tr>
<td>2</td>
<td>13/1/2014-24/1/2014</td>
</tr>
<tr>
<td>3</td>
<td>27/1/2014-7/2/2014</td>
</tr>
<tr>
<td>4</td>
<td>10/2/2014-21/2/2014</td>
</tr>
<tr>
<td>5</td>
<td>24/2/2014-7/3/2014</td>
</tr>
<tr>
<td>6</td>
<td>10/3/2014-21/3/2014</td>
</tr>
<tr>
<td>10</td>
<td>5/5/2014-16/5/2014</td>
</tr>
<tr>
<td>11</td>
<td>19/5/2014-30/5/2014</td>
</tr>
</tbody>
</table>

Taula 27. Períodes SPRINTs
6.1.2 Planificació final

La planificació final és la que realment s'ha dut a terme i conté diferències amb la planificació inicial, especialment en les dates de finalització de les etapes. Per una banda, algunes de les etapes han necessitat més hores de les previstes, especialment la part d'experimentació de Freeling, l'implementació i l'experimentació. Això es deu als problemes que han sorgit amb l'eficiència de Freeling, la implementació pròpia del clústering jeràrquic i el disseny d'una interfície per poder visualitzar els resultats del testeig, de manera que a la planificació inicial es va fer una estimació poc precisa, i a que han anat sortint imprevistos que ha calgut corregir.

A la taula següent es mostren els SPRINTs de cada etapa de la planificació inicial i de la final per tal que es pugi comparar entre elles.

<table>
<thead>
<tr>
<th>Històries d’usuari</th>
<th>SPRINTS Inicials</th>
<th>SPRINTS Finals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Investigació Storm</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Experimentació Desplegament de storm en local</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Experimentació Desplegament de storm en Remot</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Investigació Redis</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Experimentació Redis</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Actividad</td>
<td>Sprints</td>
<td>Total del proyecto(hores)</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>---------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>Experimentació Freeling</td>
<td>3</td>
<td>3-4(+1)</td>
</tr>
<tr>
<td>Investigació Serveis Amazon EC2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Experimentació desplegament clúster remot amazon EC2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Analitzar diverses opcions de personalització i recomanació de continguts</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Investigació trident-ml KLD-Classifier</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Planificació temporal</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Definició del projecte</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Especificació</td>
<td>6-7-8-9-10</td>
<td>6-7-8-9-10</td>
</tr>
<tr>
<td>Diseny</td>
<td>6-7-8-9-10</td>
<td>6-7-8-9-10</td>
</tr>
<tr>
<td>Implementació en Local</td>
<td>7-8-9</td>
<td>6-7-8-9-10-11(+2)</td>
</tr>
<tr>
<td>Implementació en Remot</td>
<td>9-10</td>
<td>9-10</td>
</tr>
<tr>
<td>Experimentació software final en local i desplegat al cloud d’EC2</td>
<td>11</td>
<td>11-12(+1)</td>
</tr>
<tr>
<td>Experimentació software final en comparació de la recomanació inicial</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>Documentació</td>
<td>6-7-8-9-10-11</td>
<td>6-7-8-9-10-11</td>
</tr>
<tr>
<td><strong>Total del projecte(hores)</strong></td>
<td>660h</td>
<td>900h</td>
</tr>
</tbody>
</table>

En la planificació final de la taula anterior, les històries d’usuari que han requerit més dedicació de la planificada inicialment s’ha indicat entre parèntesis, el nombre de SPRINTS completats necessàries per acabar les històries. Cal remarcar, que no s’han desplaçat totes les històries a la taula per facilitar la comparació amb la planificació inicial.

### 6.2. Anàlisis econòmic

#### 6.2.1 Costos de hardware

A la taula següent es poden observar els costos del hardware necessari:

<table>
<thead>
<tr>
<th>Producte</th>
<th>Preu</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 PC Intel QuadCore 2</td>
<td>600€</td>
</tr>
<tr>
<td>Recursos Amazon Ec2</td>
<td>20€</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>620€</strong></td>
</tr>
</tbody>
</table>

*Taula 29. Costos hardware*
6.2.2 Costos de software
A la taula següent es poden observar els costos del software necessari per a elaborar el projecte i la memòria:

<table>
<thead>
<tr>
<th>Producte</th>
<th>Preu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Office 2010</td>
<td>119€</td>
</tr>
<tr>
<td>Windows 7</td>
<td>119,99€</td>
</tr>
<tr>
<td>AgroUML</td>
<td>0€</td>
</tr>
<tr>
<td>Eclipse</td>
<td>0€</td>
</tr>
<tr>
<td>VMware Player</td>
<td>0€</td>
</tr>
<tr>
<td>CentOS</td>
<td>0€</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>238,99€</strong></td>
</tr>
</tbody>
</table>

Taula 30. Costos software

6.2.3 Costos de personal
Assumint que el preu per hora és de 30€/h i el total d'hores del projecte són 660 obtenim el següent cost de personal:

<table>
<thead>
<tr>
<th>Total d'hores treballades</th>
<th>Preu/hora</th>
<th>Preu</th>
</tr>
</thead>
<tbody>
<tr>
<td>660 hores</td>
<td>30€/h</td>
<td>19.800€</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>19.800€</strong></td>
</tr>
</tbody>
</table>

Taula 31. Costos de personal
6.2.4 Costos fixes

A la taula següent es poden observar els costos fixes del projecte. Aquests costos depenen de la duració del projecte en mesos, no en hores. És a dir, com hem vist a la planificació aproximadament 5 mesos.

<table>
<thead>
<tr>
<th>Producte</th>
<th>Període</th>
<th>Preu/mes</th>
<th>Preu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet</td>
<td>5 mesos</td>
<td>35€/mes</td>
<td>175€</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td>175€</td>
</tr>
</tbody>
</table>

_Taula 32. Costos fixes_

6.2.5 Costos d’Explotació

En aquest apartat tractarem de definir el costos d’explotació del nostre software al clúster d’amazon. Cal tenir en compte, els següents aspectes:

- Instàncies seleccionades de Amazon(Micro, Small, Medium, Large, etc.)
- Temps d’utilització

Seguint l’estudi presentat al primer capítol d’aquest document, tindrem els següents recursos:

- 2 Instàncies Small i 2 Instàncies Large
- Un Consum diari de 24h

El següent pas consisteix en calcular el preu anual:

<table>
<thead>
<tr>
<th>Instància</th>
<th>Màquines</th>
<th>Preu/hora</th>
<th>Hores al dia</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>2</td>
<td>0,06$</td>
<td>24</td>
</tr>
<tr>
<td>Large</td>
<td>2</td>
<td>0,24$</td>
<td>24</td>
</tr>
<tr>
<td>Total anual</td>
<td>5256$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

_Taula 33. Costos d’explotació_

El cost anterior correspon a la opció normal, Amazon ens ofereix una altre opció on realitzem uns pagaments puntuals per cada màquina i ens ofereix uns descomptes en els preus per hora (durant com a màxim 3 anys). A continuació mostrem els nous preus per hora amb els corresponents pagaments puntuals.
A partir dels pagaments exposats en les taules anteriors, podem calcular el pressupost anual mitjançant els pagaments puntuals:
<table>
<thead>
<tr>
<th>Pagament puntual</th>
<th>Cost Anual</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>5256$</td>
</tr>
<tr>
<td>Lleuger</td>
<td>3586,4$</td>
</tr>
<tr>
<td>Mitjà</td>
<td>3225,6$</td>
</tr>
<tr>
<td>Intens</td>
<td>2916,4$</td>
</tr>
<tr>
<td>Millor opció</td>
<td>2916,4$</td>
</tr>
</tbody>
</table>

Taula 37. Costos Anuals

6.2.6 Costos totals del projecte

<table>
<thead>
<tr>
<th>Concepte</th>
<th>Preu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Costos de hardware</td>
<td>600€</td>
</tr>
<tr>
<td>Costos de software</td>
<td>238,99€</td>
</tr>
<tr>
<td>Costos de personal</td>
<td>19.800€</td>
</tr>
<tr>
<td>Costos fixes</td>
<td>175€</td>
</tr>
<tr>
<td>Total</td>
<td>20.813,99€</td>
</tr>
</tbody>
</table>

Taula 38. Costos totals del projecte

Com hem vist, tenim uns costos totals de desenvolupar el projecte de 20.813,99€ i uns costos d'explotació anuals de 2916,4$ aproximadament 2118,09€.
7 Conclusions

En aquest capítol es recullen algunes reflexions finals sobre el projecte: objectius aconseguits, problemes trobats, possibles millores, opinió personal, etc.

7.1 Objectius aconseguits

Un cop finalitzades totes les fases de desenvolupament del projecte podem afirmar que s’han complert els objectius marcats a l’inici del projecte.

S’ha desenvolupat un sistema aplicant clústering jeràrquic que com hem comentat anteriorment, si tenim un conjunt de dades de 1 milió, el nombre de comparacions que haurem de realitzar en mitjana per obtenir les seves recomanacions serà la mida del clúster més 20 comparacions per recórrer el clúster jeràrquic. Si incrementem el conjunt de dades a 2 milions haurem de realitzar només 1 comparació més. Així que, l’objectiu de recursos/escalabilitat(en temps) s’ha assolit.

L’escalabilitat en espai també s’ha vist en el primer capítol d’experimentació que no es un problema ja que les estructures creades per Lupa ocupen molt. I les notícies ja estan emmagatzemades en altres sistemes de TVC.

Durant l’experimentació s’ha estudiat el cost temporal d’inserció al sistema en diferents escenaris. Com hem vist si no apliquem el clústering jeràrquic (escenari 1) no obtenim el nostre objectiu d’eficiència ja que superem el llindar de 1 minut. En canvi, els altres dos escenaris (2 i 3) on si que s’utilitza el clústering assolim el nostre objectiu. Cal remarcar, que en l’entorn desplegat al Cloud obtenim una millora del 450% respecte al desplegament en local.

<table>
<thead>
<tr>
<th>Escenari 1</th>
<th>Escenari 2</th>
<th>Escenari 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temps</td>
<td>2 minuts</td>
<td>9 segons</td>
</tr>
</tbody>
</table>

L’objectiu d’exactitud és més difícil de mesurar, però com hem vist en l’apartat d’experimentació els resultats són molt bons. Tot i això per saber si són millors que els actuals s’haurà d’esperar a que el sistema sigui explotat i podem analitzar els marcats.

L’objectiu d’extensibilitat és més subjectiu. No és fàcil decidir si un sistema està programat de manera que facilita possibles ampliacions. No obstant, durant la implementació del sistema s’ha intentat complir aquest objectiu fent les funcionalitats el màxim d’independents. És per aquest motiu que considero que aquest objectiu s’ha complert.

Els objectius de tolerància a error i capacitat de recuperació, s’han tingut en compte durant la implementació mantenint en tot moment l’estat del sistema de manera que si tenim qualsevol error es recuperei tot el treball processat anteriorment al reiniciar el sistema. Per altra banda, Storm també s’encarregarà de gestionar la tolerància a errors dintre la seva topologia.
7.2 Futures Ampliacions

Malgrat que s’han complert els objectius fixats, sempre es poden seguir fent millores al sistema.

**Interfície**

Com ja s’ha comentat anteriorment, la interfície escollida és senzilla i funcional, ja que no era la idea crear una interfície pel sistema. S’ha desenvolupat per facilitar la fase de testing ja que no era fàcil la interpretació del clústering i de les recomanacions. En un futur, es podria implementar una interfície millorant l’estètica de les vistes i incrementar les funcionalitats (inserció massiva, idiomes, etc.).

**Millorar l’algorisme de recomanació**

Gràcies a que hem programat el sistema per a futures ampliacions o modificacions, es podrien testejar diversos algoritmes de recomanació per intentar millorar la recomanació.

També es pot intentar millorar la recomanació aplicant una de les següents ideas:
- incrementar la influència del títol.
- eliminar paraules amb una presència menor de X ocurrències.

**Eliminar continguts**

Un cas que no està contemplat és l’eliminació de continguts, encara que avui en dia, el sistemes de recomanació no solen eliminar continguts, ja que és informació que es pot utilitzar en un futur per recomanacions entre usuaris (gustos, preferències, etc.). En un futur, es podria afegir l’opció d’eliminar permanentment un contingut del sistema.
7.3 Valoració Personal

Considero que l'elaboració d'aquest projecte ha sigut molt profitós per mi.

Per una banda, durant la carrera mai s'havia dut a terme un projecte complet com en aquest cas, amb totes les seves fases. En algunes assignatures es realitzava únicament la part d'especificació i disseny, mentre que en d'altres aquesta part s'ignorava o es realitzava de manera molt breu i superficial. Per primera vegada he desenvolupat un projecte amb totes les seves fases, de principi a fi, i això ha permès consolidar i repassar molts dels conceptes apresos durant la carrera.

Per altra banda, per fer aquest projecte he hagut d'investigar i aprendre alguns sistemes, tecnologies o llibreries que fins ara no havia tocat mai i que poden resultar molt útils en el meu futur laboral: Storm, Redis, Amazon Web Services, etc. A més, tota aquesta investigació i aprenentatge s'ha hagut de fer pel meu compte, sense apunts a seguir com fins ara, que és el que ens trobarem un cop entrem al món laboral.

En conclusió, el projecte m'ha suposat una experiència molt positiwa de cara al meu futur i m'ha permès aprendre i aplicar noves tecnologies que em poder ser molt útils.
Referències

[1]. Software engineering - Software product Quality Requirements and Evaluation
[2]. Non-Technical Quality Features Catalogue
http://www.essi.upc.edu/~qms/DesCOTS/CQM/Non.html
[4]. Application of the Analytic Hierarchy Process (AHP)
[7]. K-means http://home.deib.polimi.it/matteucc/Clústering/tutorial_html/kmeans.html
[8]. Projecte gutenberg https://www.gutenberg.org
Annexos

En els annexos he posat informació complementària d’ajut en el projecte juntament amb el glossari de taules i figures.

Annex A: Resultats recomanació notícies TVC

Al capítol, d’experimentació s’han mostrar dues recomanacions. En aquest annex es presenten totes les dades obtingudes durant la experimentació.

Recomendations

<table>
<thead>
<tr>
<th>ID</th>
<th>D'èsser</th>
</tr>
</thead>
<tbody>
<tr>
<td>2255388</td>
<td>5.642</td>
</tr>
<tr>
<td>2858251</td>
<td>5.167</td>
</tr>
<tr>
<td>2715683</td>
<td>5.110</td>
</tr>
<tr>
<td>2716043</td>
<td>5.293</td>
</tr>
<tr>
<td>3721273</td>
<td>3.203</td>
</tr>
</tbody>
</table>

Selected text Queralt Castellet acaba sisena en la final de “superpipe” als X Games a Aspen

Queralt Castellet ha acabat sisena en la final de “superpipe” als X Games que s’esten celebrant a la ciutat d’Aspen, als Estats Units. La catalana ha fet una gran primera baxada en que ha treballat la segona pel·lícula polaritzant amb 79 punts. A les dèues següents, les coses no li han anat tan bé. Ha avaluat 29 i 120 punts en les dues pel·lícules i a l’època ha quedat a 12 punts de la guanyadora, la nord-americana Kelly Clark.

Il·lustració 41. Recomanació identificador 2029514
Il·lustració 42. Recomanació identificador 2132613
Recommended text:

**Sánchez-Camacho acusa d'"opportunisme polític" Esperanza Aguirre per la seva postura en el "cas Bárscenas"**

L'any passat a PP de "cas Bárscenas". La presidenta de PP escaixa, Anna Sánchez-Camacho, acusa a Esperanza Aguirre, la presidenta del PP de Madrid, de "opportunisme polític" per la seva postura en el cas Bárscenas. Aguirre ha afirmat que "en aquest moment, el PP ha de rebre ajudes per a la seva continuïtat". Sánchez-Camacho ha denunciat que Aguirre ha "omissió" alguns aspectes en el cas Bárscenas.

**Sánchez-Camacho demana als barons del PP que no questionin les peticions de finançament catalanes**

La presidenta de PPC, Anna Sánchez-Camacho, ha demanat als barons del PP que no questionin les peticions de Catalunya per mitjans de finançaments. Sánchez-Camacho ha expresat en roda de premsa que "és important no desviar la nostra atenció dels problemes reals de Catalunya". La presència de Sánchez-Camacho en Comunitat de Madrid ha estat debatuda per alguns membres del PP.

Il·lustració 43. Recomanació identificador 2149617
Gerardo Martino assegura que el Barça arriba bé a l’inici de la Lliga tot i que no s’ha vist la millor versió del seu equip.

Martino lloa la il·lusió de Puyol per tornar a jugar i confirma que Piqué arrossega molesties al pubis.

**Il·lustració 44. Recomanació identificador 2159793**
Selected text

Ratto guanya en solitari una dura etapa de muntanya que deixa Nibali més lider

L’últim Daniel Ratto ha guanyat la comunitat etapa de la Vuelta a Espanya antecipant en solitari el Col de la Galarne (Andorran). El gran beneficiat de la jornada ha estat Vincenzo Nibali, que ha mantingut el mollet vermell encara ha augmentat la distància amb els seus coneguts. L’anticida de la cursa és Filippo Pozzato ha estat marcat per la jota i té novament la posició més reial del comitiva del Cannondale, que ha treballat escapat des de l’Kilometre 3 d’un trajecte de 154’. Les condicions han estat dures fins i tot que alguns corredors, com i van Basto o Luis León Sánchez, han hagut d’abandonar efectes físics. L’assertiu de Ratto ha començat en un grup de mico, que a mida etapa s’ha acabat a ràp, i al quilòmetre 150 de fins ara tot s’han mantingut a dins de el grup de perseguidors més rendits per la majoria de favorits de la general, entre ells Nibali, Óscar Pereiro, Joséjane Rodriguez i Alejandro Valverde. Però els assajos han estat perdent potencial a mesura que les rutes

Treballadors d’Ercos barren el pas de camions a la fàbrica contra l’ERO que preveu 156 acomiadaments

Treballadors d’Ercos a Figueres, estan començant dels de les de la petita a les de la seva fàbrica per impedir que li puguin entrar els camions. És un cop de protecció contra l’ERO presentat a finals de la temporada passada per Ercos, que preveu aconseguir 156 treballadors a la planta de Figueres, equivalent a tres quartes parts de la plantilla. La intenció dels treballadors és mantenir la seva cotxera fins a finals de setembre, avui, esmenta també el pas de gruix camions que hi han d’entrar per la planta per transportar el material. En total, hi estan, avui, 200 treballadors que es van presentar a la planta de Figueres per impedir que li puguin entrar els camions fins a finals de setembre. D’aquesta manera, la companyia, que també ha aconsellat el foment de la planta de Sants. (Font: d1)

Il·lustració 45. Recomanació identificador 2169173
Recomendacions

<table>
<thead>
<tr>
<th>Núm. Técnica</th>
</tr>
</thead>
<tbody>
<tr>
<td>2058613</td>
</tr>
</tbody>
</table>

Selected text Andoni Zubizarreta viatja a Nova York per trobar-se amb Tito Vilanova

El director tecnic del Barça, Andoni Zubizarreta, va viajar d'aquesta manera a Nova York per trobar-se amb Tito Vilanova. Andoni Zubizarreta va viajar amb molta deixa, una inusual de l'any, per tal d'establir una relació amb Tito Vilanova. Aquesta és una situació molt encomanada, ja que els jugadors no tenien una relació directa amb Tito Vilanova. Andoni Zubizarreta va quedar impressionat amb la precisió i el talent de Tito Vilanova, una característica que és molt apreciada a Nova York.

Tito Vilanova reapareix avui davant la premsa, en la prèvia Athletic-Barça

L'entrenador del F. C. Barcelona, Franck Ribéry, va anunciar que la seva convocació per a la partita contra l' Athletic Barça està pronta. Ribéry va explicar que la seva presència és una prova de la voluntat del Barça per mantenir el ritme i la intensitat en la partita. Aquesta notici ha estat acollida amb entusiasme per la majoria dels espectadors del Camp Nou. Ribéry va anunciar que la seva presència és una prova de la voluntat del Barça per mantenir el ritme i la intensitat en la partita. Aquesta notici ha estat acollida amb entusiasme per la majoria dels espectadors del Camp Nou.
**Recommendations**

<table>
<thead>
<tr>
<th>ID</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>8359151</td>
<td>5.302</td>
</tr>
<tr>
<td>4778772</td>
<td>2.97176</td>
</tr>
<tr>
<td>4778773</td>
<td>5.99944</td>
</tr>
<tr>
<td>8359152</td>
<td>3.31775</td>
</tr>
<tr>
<td>4778774</td>
<td>3.39893</td>
</tr>
</tbody>
</table>

**Selected text**
Mor una dona d’edat avançada arran d’un incendi a la seva casa, a Vilanova i la Geltrú

Una dona d’edat avançada ha mort a primera hora d’aquesta tarda en l’incendi d’un habitatge a Vilanova i la Geltrú. S’ha informat als Bombers de la Generalitat, un vol ha treballat al fons d’incendi. A més a més, diversos homes han voluntariament ajudat a apaginar les llums. La víctima va passar en comarca i, a més a més, s’ha anomenat la causa accidental del foc. La víctima va passar en comarca i, a més a més, s’ha anomenat la causa accidental del foc. La víctima va passar en comarca i, a més a més, s’ha anomenat la causa accidental del foc.

**Un incendi destrueix el santuari de la Mare de Déu d’A Barca, a Muxía**

Un incendi ha cremat l’emblemàtic santuari de la Mare de Déu d’A Barca, a Muxía, a costa de la Corunya. Les flama han tenut molta força i han podut portar al santuari fins i tot les forquilles de la torre de la vila. La mare de Déu ha estat tot justament cremada, i el santuari ha estat destruït. A més, també s’ha anomenat a la Catedral de Muxía. Aquest incendi ha estat destruït el santuari de la Mare de Déu d’A Barca, a Muxía, a costa de la Corunya. Les flama han tenut molta força i han podut portar al santuari fins i tot les forquilles de la torre de la vila. La mare de Déu ha estat tot justament cremada, i el santuari ha estat destruït. A més, també s’ha anomenat a la Catedral de Muxía.

Il·lustració 47. Recomanació identificador 2067593
Annex B: Glossari de taules i il·lustracions

Taula 1. Nivells d'importància AHP
Taula 2. Comparació criteris amb el mètode AHP
Taula 3. Normalització comparació i ponderacions finals criteris
Taula 4. Avaluació Recomanadors
Taula 5. Risc planificació
Taula 6. Risc documentació tecnologies
Taula 7. Risc costos serveis Amazon
Taula 8. Cas d’ús inserir notícia
Taula 9. Cas d’ús generar recomanacions
Taula 10. Cas d’ús mostrar recomanacions
Taula 11. Contracte Inserir Notícia
Taula 12. Contracte Mostrar recomanacions
Taula 13. Contracte Generar recomanacions
Taula 14. Cost espai notícies versus estructures
Taula 15. Temps d’execució total dels escenaris
Taula 16. Temps d’inserció dels escenaris
Taula 17. Conjunt de dades experiment Gutenberg I
Taula 18. Resultat experiment Gutenberg I