



## Creació d'un sistema automatitzat de plantilles (.Rnw) per a la generació d'informes dinàmics mitjançant Sweave.

**Alumne: Aina Bio Llabrés**

Pla d'estudis: Màster en Bioinformàtica i Bioestadística

Àrea de treball: Bioinformàtica i programació

**Consultor: Alexandre Sánchez Pla**

**Responsable de l'assignatura: Alexandre Sánchez Pla, Antoni Pérez Navarro, Carles Ventura Royo, Jose Antonio Morán Moreno i Maria Jesús Marco Galindo**

Data: Diumenge, 4 de juny del 2017



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

## FITXA DEL TREBALL FINAL

<b>Títol del treball:</b>	<i>Creació d'un sistema automatitzat de plantilles (.Rnw) per a la generació d'informes dinàmics mitjançant Sweave.</i>
<b>Nom de l'autor:</b>	<i>Aina Bio Llabrés</i>
<b>Nom del consultor/a:</b>	<i>Alexandre Sánchez Pla</i>
<b>Nom del PRA:</b>	<i>Alexandre Sánchez Pla, Antoni Pérez Navarro, Carles Ventura Royo, Jose Antonio Morán Moreno i Maria Jesús Marco Galindo</i>
<b>Data de lliurament (mm/aaaa):</b>	<i>05/2017</i>
<b>Titulació o programa:</b>	<i>Màster en Bioinformàtica i Bioestadística</i>
<b>Àrea del Treball Final:</b>	<i>Bioinformàtica i programació</i>
<b>Idioma del treball:</b>	<i>Català</i>
<b>Paraules clau</b>	<i>Plantilla, R, Sweave</i>
<p><b>Resum del Treball (màxim 250 paraules):</b> <i>Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball</i></p>	
<p>Avui dia, l'ús de les aplicacions especialitzades en l'àrea de treball suposen una gran optimització de recursos tant a nivell econòmic, material com temporal. Per aquest motiu s'ha creat una aplicació web que permet la generació automàtica de plantilles dinàmiques aptes per R i Sweave amb la finalitat de generar una millora de l'eficiència i eficàcia en el treball realitzat per els treballadors dels sectors.</p> <p>Per a la seva creació s'han utilitzat diferents eines de programació com l'Atom i Xampp per tal de poder desenvolupar llenguatges de programació com ara PHP, CSS i HTML.</p> <p>Els resultats obtinguts en aquest projecte consisteixen en una pàgina web en la que es permet crear, guardar i descarregar plantilles modificables en format</p>	

RNW per, posteriorment, utilitzar-les en R i, amb l'ajuda de Sweave i Latex generar un informe científic complet.

Aquest projecte ha obert una porta per donar a conèixer el lent avanç del sector científic en el desenvolupament tecnològic i la importància de rompre amb aquesta situació actual.

**Abstract (in English, 250 words or less):**

Today, the use of specialized applications in the workspace are a great resource optimization both economic, material and temporal. For this reason it has created a web application that allows the automatic generation of dynamic templates for R and Sweave suitable for the purpose of generating improved efficiency and effectiveness in the work done by workers in the sector.

For its creation have been using different programming tools such as Atom and Xampp in order to develop programming languages such as PHP, HTML and CSS.

The results obtained in this project consist of a web page that allows to create, save and download modifiable templates RNW format for later use it in R and with the help of Sweave and Latex generate a complete scientific report.

This project has opened a door to publicize the slow progress of scientific sector in the technological development and the importance of breaking with the current situation.

# Índex

1. Introducció.....	1
1.1 Context i justificació del Treball .....	1
1.2 Objectius del Treball.....	2
1.3 Enfocament i mètode seguit.....	3
1.4 Planificació del Treball.....	5
1.5 Breu sumari de productes obtinguts .....	9
1.6 Breu descripció dels altres capítols de la memòria .....	9
2. Coneixements previs .....	11
2.1. R.....	11
2.2. Sweave.....	12
2.3. Anàlisi de microarrays .....	14
2.4. Anàlisi estadístics .....	16
2.5. Selecció i validació de biomarcadors.....	17
3. Desenvolupament del projecte .....	19
3.1. Requeriments bàsics .....	19
3.2. Desenvolupament de la base de dades .....	23
3.3. Desenvolupament de l'entorn web .....	26
3.4. Incidències en les activitats realitzades.....	30
4. Assoliment del projecte. ....	33
5. Resultats obtinguts.....	35
6. Conclusions.....	37
7. Glossari .....	38
8. Bibliografia.....	39
9. Annexes .....	41
9.1 Documentació del projecte .....	41

## Llista de figures

<i>Imatge 1: Planificació del treball per al mes de març</i> .....	7
<i>Imatge 2: Planificació del treball per al mes d'abril</i> .....	7
<i>Imatge 3: Planificació del treball per al mes de maig</i> .....	8
<i>Imatge 4: Diagrama d'arxius per treballar amb Sweave [14]</i> .....	13
<i>Imatge 5: Ordre lògic en l'anàlisi de dades de microarrays [6]</i> .....	16
<i>Imatge 6: Modificació de l'arxiu .env de Laravel amb l'editor de text Atom un cop creada la base de dades</i> .....	20
<i>Imatge 7: Captura de pantalla de l'eina Xampp</i> .....	21
<i>Imatge 8: Captura de pantalla de l'eina Atom</i> .....	22
<i>Imatge 9: A la segona línia del codi HTML s'inclou la llibreria de Bootstrap</i> .....	22
<i>Imatge 10: A la primera línia del codi HTML s'inclou la llibreria de Tinymce</i> .....	22
<i>Imatge 11: Esquema organitzatiu de les taules que componen la base de dades utilitzada</i> .....	25
<i>Imatge 12: Exemple del llenguatge HTML i CSS</i> .....	29
<i>Imatge 13: Flux d'informació de la pàgina web. En color blau es representen les interaccions de la web amb l'usuari i de color vermell els processos que es desenvolupen en segon pla</i> .....	35
<i>Imatge 14: Codi per a la creació de la base de dades de la taula "claves"</i> .....	41
<i>Imatge 15: Codi PHP per determinar les relacions entre les taules, en aquest cas entre la taula "claves" i la taula "valor"</i> .....	42
<i>Imatge 16: Consulta a la base de dades de les taules creades amb Laravel 5.4</i> .....	43

<i>Imatge 17: Fragment de codi utilitzat per l'emplenament del contingut de les taules que conformen la base de dades creades anteriorment.....</i>	<i>44</i>
<i>Imatge 18: Plana inicial de benvinguda.....</i>	<i>45</i>
<i>Imatge 19: Fragment més representatiu del codi utilitzat per a la plana inicial de benvinguda.....</i>	<i>45</i>
<i>Imatge 20: Plana de l'accés a "Cómo funciona?" on apareix una breu descripció de l'ús de la web i la obtenció de la plantilla.....</i>	<i>46</i>
<i>Imatge 21: Fragment de codi representatiu de la plana "Cómo funciona?".....</i>	<i>47</i>
<i>Imatge 22: Plana de "Cómo funciona?" amb la sessió d'usuari iniciada. ....</i>	<i>47</i>
<i>Imatge 23: Plana de la pàgina de benvinguda amb la sessió d'usuari iniciada</i>	<i>48</i>
<i>Imatge 24: Plana de l'accés a "Registrarse" amb el formulari per emplenar de l'usuari.....</i>	<i>48</i>
<i>Imatge 25: Plana de l'accés a "Entrar" amb el formulari a emplenar per l'usuari .....</i>	<i>49</i>
<i>Imatge 26: Llenguatge PHP per determinar les rutes d'accés per usuaris identificats. ....</i>	<i>49</i>
<i>Imatge 27: Plana del directori home un cop iniciada la sessió. ....</i>	<i>50</i>
<i>Imatge 28: Plana inicial amb el botó de microarrays marcat. ....</i>	<i>50</i>
<i>Imatge 29: Codi CSS aplicat als links de les diferents plantilles.....</i>	<i>51</i>
<i>Imatge 30: Plana web de la plantilla Microarrays. ....</i>	<i>51</i>
<i>Imatge 31: Fragment de codi per crear una llista de plantilles guardades per l'usuari.....</i>	<i>52</i>
<i>Imatge 32: Fragment del codi utilitzat per mostrar el contingut de la plantilla seleccionada. ....</i>	<i>53</i>

<i>Imatge 33: Modificació del text amb TinyMce.....</i>	53
<i>Imatge 34: Codi JavaScript per incorporar la funcionalitat de TinyMce.....</i>	54
<i>Imatge 35: Mostra d'un arxiu adjunt en representació del logo de la empresa.</i>	54
<i>Imatge 36: Codi JavaScript per a la inserció d'imatges i la seva visualització..</i>	55
<i>Imatge 37: Calendari per seleccionar la data de creació de l'informe .....</i>	55
<i>Imatge 38: Codi JavaScript utilitzat per la funcionalitat del calendari.....</i>	56
<i>Imatge 39: Plana de la opció de guardat.....</i>	56
<i>Imatge 40: Plana de la opció d'exportació en format RNW .....</i>	57
<i>Imatge 41: Codi JavaScript per a les funcionalitats de guardat i exportació. ...</i>	57
<i>Imatge 42: Fragment de codi PHP representatiu del Controlador.....</i>	58
<i>Imatge 43: Finestra de descàrrega de l'arxiu RNW.....</i>	59
<i>Imatge 44: Arxiu RNW obert amb R .....</i>	60
<i>Imatge 45: Informe resultant de la compilació de l'arxiu RNW amb R i Sweave .....</i>	61



# 1. Introducció

## 1.1 Context i justificació del Treball

En el context actual on les noves tecnologies són una eina imprescindible per dur a terme activitats d'estudi i investigació, l'ús de les aplicacions especialitzades en l'àrea de treball suposen una gran optimització de recursos tant a nivell econòmic, material com temporal per als investigadors o l'equip de treball. A més, al poder automatitzar els procediments s'aconsegueix agilitzar i evitar un gran nombre d'errors humans durant el procediment, permetent una millora de l'eficiència i eficàcia dels mètodes utilitzats.

Per aquest motiu, aquest TFM té com a finalitat la creació d'una aplicació web que permeti generar plantilles d'informes associats a un projecte, autogestionables per l'usuari i que doni lloc a un arxiu .Rnw amb l'objectiu de facilitar i agilitzar la seva elaboració.

En aquest cas, s'ha escollit l'àrea de la programació bioinformàtica degut a que, actualment, el desenvolupament dels avanços tecnològics cada cop es realitza de forma més accelerada i es fa més necessari que aquests avanços s'introdueixin dins el camp de la investigació per poder seguir utilitzant al màxim tots els avantatges que suposa la seva utilització. Així, aquest camp resulta essencial anivellar l'avanç tecnològic i l'avanç científic per mitjà de la creació nous projectes que permetin facilitar les tasques d'investigació. Un sistema que permeti agilitzar les tasques relacionades amb l'elaboració de codis de programació per dur a terme un estudi de dades amb la posterior redacció de la documentació permetria dedicar més temps a altres tasques no automatitzades. Aquesta situació ha conduït al tema escollit per aquest TFM, la creació d'un sistema automatitzat de creació de plantilles per la generació d'informes dinàmics.

En els darrers anys s'ha vist un increment de l'ús de R en bioinformàtica, on és molt habitual crear informes sobre els anàlisis realitzats. Aquests

solen contenir tant el text explicatiu com els resultats obtinguts a través de codi de programació, ja sigui en forma gràfica, esquematitzada, amb imatges o graelles. Per poder combinar ambdós aspectes, es freqüent utilitzar, junt amb l'ús de R i Latex, sistemes com Sweave o Rmarkdown que permeten combinar el text ordinari amb el codi R necessari per a la realització de l'estudi. Si més no, la redacció d'aquests informes suposen una feina llarga i repetitiva degut a que el procediments que es duen a terme no varien i els canvis que es solen realitzar són a nivell de resultats. Per aquest motiu, es vol desenvolupar un sistema de creació automàtica de plantilles per tal d'estalviar temps i complexitat als/les estadistes o bioinformàtics/ques en la seva elaboració sense que la qualitat del producte resultant es vegi afectada i que permeti agilitzar les tasques relacionades amb l'elaboració de codis de programació.

A més, aquesta eina que es vol crear, també té com objectiu facilitar la feina als treballadors que no tenen alts coneixements de programació, per aquest motiu s'ha intentat imitar el format de l'informe a obtenir en una plana web on el desenvolupament d'aquest resulta més intuïtiu. Per desenvolupar el contingut de les plantilles provisionals sobre les que fer les modificacions s'ha utilitzat la informació facilitada per la Unitat d'Estadística i Bioinformàtica del Vall d'Hebron Institut de Recerca

## **1.2 Objectius del Treball**

Per tal de poder avaluar correctament l'avanç del projecte és necessari establir uns objectius. En aquest cas només s'ha considerat un únic objectiu general. Per tal de facilitar el seu assoliment s'han establert dos objectius específics que s'hauran de complir al llarg del programa. Així, els objectius del treball es resumeixen en:

- Objectius generals:
  - Desenvolupar un generador automàtic de plantilles d'informes dinàmics en format Rnw.

- Objectius específics:
  - Crear una base de dades amb els diferents components per a tres tipus d'informe: anàlisi de microarrays, anàlisi estadístic i selecció i validació de biomarcadors.
  - Implementar l'eina en entorn web permetent la modificació del contingut i l'exportació de l'arxiu en format Rnw.

### 1.3 Enfocament i mètode seguit

Avui en dia es molt comú la utilització de Microarrays en el camp de la investigació, especialment en l'estudi i desenvolupament de malalties, ja que permeten realitzar una gran quantitat de mesures de forma simultània. Aquest fet permet obtenir grans quantitats de dades que permeten conèixer múltiples mapes d'expressió gènica, detectar biomarcadors i inclús construir firmes gèniques per a determinades malalties [17, 25]. Així un equip de bioinformàtica que treballi en aquest sector es freqüent que realitzi tres tipus d'informes on es reflecteixi l'estudi realitzat: un per a l'anàlisi de microarrays, un altra per a l'anàlisi estadístic i un altra per a la selecció i validació de biomarcadors. Per aquest motiu, tot i l'existència de molts altres estudis, aquest TFM es centra en facilitar aquests tres tipus de plantilles.

Per tal de facilitar l'accés a l'eina que es vol desenvolupar en aquest projecte s'ha decidit crear una pàgina web. D'aquesta manera el producte és completament independent del sistema operatiu amb el que es treballi i no resulta necessari una instal·lació concreta de cap eina ni aplicació llevat de l'eina base R i els paquets respectius per a la creació de documents, en aquest cas Latex i Sweave. A més, l'ús de l'entorn web permet construir una aplicació visualment atractiva que la converteix en un instrument més intuïtiu, àgil i ràpid.

Així doncs, per tal de dur a terme aquest projecte i obtenir els resultats esperats, caldrà utilitzar diferents llenguatges i eines de programació que permetin desenvolupar tant la base de dades com l'entorn web. Tot i que aquests es descriuran de forma més exhaustiva en punts posteriors d'aquesta memòria cal tenir-ne un esbós per tal de comprendre millor la complexitat que suposa la realització d'aquesta tasca.

En primer lloc, per tal d'elaborar la base de dades, s'utilitzarà el llenguatge SQL. En aquesta base de dades s'hi ha d'incloure cada una de les parts que formen un informe. Aquestes poden estar formades per text estàtic, que no permet modificacions, com són els encapçalaments, o text dinàmic, que pot ser modificat, com ara les conclusions. A més, aquests texts poden variar en la seva longitud, un factor que també s'ha de tenir en compte ja que podem trobar-ne de més breus, com els títols, o més extensos, com les descripcions. Per altra banda també s'han d'incloure els *chunks* de codi en R propis per desenvolupar l'informe. També es important considerar que no tots els informes tenen el mateix contingut, pel que caldrà diferenciar aquelles parts que siguin comuns en tots els informes i aquelles que siguin específiques per a cada un d'ells. Finalment, s'ha de tenir en compte, a l'hora de relacionar la base de dades que l'usuari ha de poder afegir nous registres o modificar els existents.

En segon lloc s'utilitzarà PHP, a través del *framework* Laravel 5.4, per crear l'aplicació web amb les funcionalitats necessàries. A més, també es farà ús dels llenguatges HTML, CSS i, de forma molt puntual, JavaScript per organitzar i dissenyar la vista d'aquesta pàgina web, el que es coneix com disseny web, que resulta clau per a una correcta visualització de totes les funcions que s'ofereixen. El resultat conjunt de tots aquests llenguatges permet que la web serveixi d'intermediari entre l'usuari i la base de dades per a que es pugui crear un nou projecte, modificar-ne un d'existent i inclús editar la base de dades. D'aquesta manera, interactuant a través de les pestanyes de selecció, és possible seleccionar el contingut desitjat per a generar un informe fet a mida en format .Rnw. A més la web ha de permetre accedir a un registre existent

o crear-ne un de nou per tal de poder desar els projectes que es desitgin o les modificacions realitzades i poder-hi accedir més endavant.

Finalment, al utilitzar-se llenguatges de programació desconeguts fins al moment, serà necessari un aprenentatge previ. Una possible estratègia de treball seria documentar-se a l'inici del treball sobre els diferents llenguatges de programació i posteriorment realitzar tota la tasca de programació. Ara bé, utilitzar diferents llenguatges de programació suposa abastir un coneixement molt ampli, pel que, possiblement, sigui més apropiat realitzar aquest aprenentatge de mica en mica. Per aquest motiu, s'ha decidit documentar-se sobre el llenguatge de programació en concret que s'utilitzarà en cada tasca abans del seu ús pel que, abans de realitzar la tasca amb un llenguatge de programació concret, s'han reservat unes hores a l'aprenentatge d'aquest. Així es poden centrar les idees en una sola tasca, fet que facilita un millor rendiment i, al seu torn, permet l'obtenció de millors resultats

#### **1.4 Planificació del Treball**

Una bona planificació del treball resulta clau per al desenvolupament del projecte, ja que permet distribuir les activitats dins el termini de temps establert i conèixer en tot moment l'estat de l'avanç l'entrega.

Així, per tal d'ajustar el temps del que es disposa per realitzar aquest projecte de la forma més precisa possible s'han establert un seguit de tasques necessàries per complir els objectius específics. A més, cada tasca té assignades les hores de dedicació que s'hi hauria d'invertir com a màxim per finalitzar-les. Cal tenir en compte que dins aquestes hores ja s'hi considera un temps de marge prudencial per als possibles contratemps que puguin sorgir durant l'avanç del procés i, així, evitar la repercussió en les tasques futures que podria comportar un retard en els terminis establerts. A continuació es detallen les anomenades tasques amb el temps determinat per realitzar cada una d'elles:

- Organització dels diferents components que conformen els informes: 16h
- Documentació i aprenentatge sobre base de dades (BDD): 12h
- Disseny de la base de dades: 20h
- Creació de la base de dades: 28h
- Documentació i aprenentatge sobre HTML i CSS: 16h
- Creació de l'entorn web amb HTML: 32h
- Maquetació amb CSS: 28h
- Documentació i aprenentatge sobre Laravel (PHP): 20h
- Programació de la web amb Laravel: 36h
- Inserció de dades: 20h
- Comprovació, correcció i perfeccionament del projecte: 36h

Amb aquesta distribució, les hores dedicades a les tasques per al desenvolupament del projecte és de 264, un 70% del total d'hores disponibles. A més, d'aquestes 264 hores un 20% s'invertirà en la documentació i l'aprenentatge de les eines a utilitzar. De les 111 hores restants, un 30% del total d'hores disponibles, s'han distribuït entre la resta de tasques de l'assignatura de la següent manera:

- PAC 1 Pla de treball: 24h
- PAC 2 Desenvolupament del treball I: 16h
- PAC 3 Desenvolupament del treball II: 16h
- Memòria del TFM: 35h
- Preparació de la defensa: 20h

A l'hora de distribuir les tasques en el calendari s'han tingut en compte les activitats externes a les assignatures del màster que s'està cursant, així com el temps dedicat en les altres assignatures del curs. Per aquest motiu s'ha considerat una dedicació de 4 hores diàries en els dies d'entre setmana i una dedicació de 8 hores diàries en els dies de cap de setmana.

Com es pot veure en les següents imatges, s'han distribuït les activitats al llarg del mes de març (imatge 1), el mes d'abril (imatge 2) i el mes de maig (imatge 3) considerant el temps planificat per a cada activitat i les hores diàries que es creu que es poden dedicar al dia.

MARÇ 2017						
DILLUNS	DIMARTS	DIMECRES	DIJOUS	DIVENDRES	DISSABTE	DIUMENGE
		1	2	3	4	5
6	7	8	9	10	11	12
PAC 1: Pla de treball						
13	14	15	16	17	18	19
			Organització components d'informes			
20	21	22	23	24	25	26
Documentació BDD			Disseny BDD			
27	28	29	30	31	1	2
Creació BDD						

Imatge 1: Planificació del treball per al mes de març

ABRIL 2017						
DILLUNS	DIMARTS	DIMECRES	DIJOUS	DIVENDRES	DISSABTE	DIUMENGE
					1	2
3	4	5	6	7	8	9
PAC 2: Desenvolupament del treball I			Documentació htm i css			
10	11	12	13	14	15	16
Creació de l'entorn web amb html						
17	18	19	20	21	22	23
Maquetació amb css						
24	25	26	27	28	29	30
Documentació Laravel (PHP)			PAC 3: Desenvolupament del treball II			

Imatge 2: Planificació del treball per al mes d'abril

MAIG 2017						
DILLUNS	DIMARTS	DIMECRES	DIJOUS	DIVENDRES	DISSABTE	DIUMENGE
1	2	3	4	5	6	7
Programació web amb Laravel						
8	9	10	11	12	13	14
Inserció de dades						
15	16	17	18	19	20	21
Comprovació del projecte					PAC 3: Memòria TFM	
22	23	24	25	26	27	28
				Preparació de la defensa		
29	30	31				

*Imatge 3: Planificació del treball per al mes de maig*

Finalment s'han establert un seguit de fites a assolir al llarg del projecte. Aquestes resulten essencials per tal de poder dur a terme el correcte seguiment del projecte. A més, també permeten avaluar amb major facilitat l'avanç en les successives etapes que es van desenvolupant al llarg del projecte. Per tant, a l'hora de valorar els resultats obtinguts, el correcte compliment d'aquestes resultaran una peça clau. Les fites proposades en aquest projecte són les següents:

- Creació de la base de dades amb SQL
- Creació de l'entorn web amb HTML i maquetació amb CSS
- Programació web amb Laravel
- Entrega del projecte
- Entrega de la PAC 1
- Entrega de la PAC 2
- Entrega de la PAC 3
- Entrega de la Memòria del TFM
- Realització de la defensa del TFM



## 1.5 Breu sumari de productes obtinguts

Els productes obtinguts al llarg del desenvolupament del projecte són els següents:

- **Pla de treball:** en el qual s'hi ha plasmat el projecte que es vol desenvolupar així com una bona organització temporal de les tasques on s'ha tingut en compte la disponibilitat d'hores per treballar i els possibles riscos existents.
- **Memòria:** que reflecteix tot el treball desenvolupat amb explicacions clares. A més de presentar tots aquells imprevistos, canvis o millores dutes a terme durant l'elaboració del projecte.
- **Software com a producte final:** que consisteix en un software de qualitat, que respon als objectius inicials i demostra el correcte assoliment dels coneixements apresos durant el curs.
- **Presentació virtual:** per tal de defensar el projecte davant un jurat i resoldre les qüestions que es plantegin.

## 1.6 Breu descripció dels altres capítols de la memòria

En els pròxims capítols de la memòria es vol reflectir tot el treball realitzat al llarg del projecte. Per aquest motiu és important establir l'ordre correcte dels punts a desenvolupar en la memòria i els seus continguts. Així doncs, s'han inclòs els següents capítols a la memòria:

- **Coneixements previs:** on s'exposen tots els coneixements bàsics a partir dels quals s'ha desenvolupat el treball i marquen les seccions en que aquest punt es troba dividit :
  - R
  - Sweave

- Anàlisi de microarrays
- Anàlisi estadístics
- Selecció i validació de biomarcadors
- **Desenvolupament del projecte:** on s'explica detalladament els requeriments necessaris per dur a terme el projecte i l'elaboració tant de la base de dades com de l'entorn web. A més, s'especifiquen les incidències sorgides al llarg de procés així com les respectives accions de mitigació. Tots aquests continguts s'han dividit en els següents punts.
  - Requeriments bàsics
  - Desenvolupament de la base de dades
  - Desenvolupament de l'entorn web
  - Incidències en les activitats realitzades
- **Assoliment del projecte:** on s'autoavalua el correcte compliment dels objectius i les fites marcades.
- **Resultats obtinguts:** que plasma el funcionament, en aquest cas, de la pàgina web que s'ha creat amb totes les seves funcionalitats.
- **Conclusions:** on es reflexiona sobre els continguts aportats en el projecte i es du a terme un anàlisi crític de la metodologia planificada.

Finalment, també s'inclouen els capítols comuns referents al glossari, la bibliografia i els annexos.

## 2. Coneixements previs

### 2.1. R

R és un llenguatge i un entorn pensat per a la computació i els gràfics estadístics. És un projecte GNU (projecte col·laboratiu de software lliure amb l'objectiu de crear un sistema operatiu completament lliure) similar al llenguatge S (llenguatge de programació estadístic) que va ser desenvolupat en els Laboratoris Bell (abans d'AT & T, ara Lucent Technologies) per John Chambers et al. Així, R pot ser considerat com una implementació diferent del llenguatge S ja que hi ha algunes diferències importants tot i que molt codi escrit per S es mantén inalterat sota l'ús de R.

Aquesta eina ofereix una àmplia varietat de tècniques gràfiques, estadístiques (proves estadístiques clàssiques, anàlisi de sèries temporals, classificació, agrupament, ...) i és altament extensible proporcionant una ruta de codi obert per a la participació de la investigació estadística. D'aquesta manera, permet integrar els serveis de programació per a la manipulació de dades, càlcul i representacions gràfiques per mitjà de la integració d'eines intermèdies per a l'anàlisi o instal·lacions de paquets que ofereixen diferents funcionalitats.

En resum, R és un entorn de programació dissenyat específicament per a treballs estadístics i, probablement, és un dels llenguatges més utilitzats per la comunitat estadística, sent a més molt popular en el camp de la investigació biomèdica, la bioinformàtica i les matemàtiques financeres. Es tracta, a més, d'un programa modular, el que significa que els usuaris poden escriure extensions i distribuir els codis a altres usuaris. El codi es distribueix en forma de llibreries llestes per a la seva utilització que es denominen paquets. La major part de la funcionalitat de R es troba en aquests paquets, no només en els realitzats pels desenvolupadors del programa mare sinó també els realitzats per altres usuaris. [16, 17]

## 2.2. Sweave

Es un component del llenguatge de programació R que permet integrar el seu codi amb el codi de documents escrits en un document Latex. En el manual oficial [15] es troba definit de la següent manera: “Sweave proporciona un marc flexible per mesclar text i codi R amb l’objectiu de generar documents de forma automàtica”. És a dir, aquest component permet la redacció d’informes en els que es pot combinar el text ordinari amb el propi codi de R i els corresponents resultats. [14]

Així doncs s’aconsegueix que un únic arxiu que conté text i codi R, obtenint com a producte final del procés un text amb el format desitjat en el que de forma opcional es pot mostrar, o no, el codi R utilitzat i en el que es mostren, també, els resultats d’aquests, ja siguin càlculs, gràfics, etc. Aquest sistema permet que, en un moment donat, es puguin realitzar canvis en les dades introduïdes o en el format del document resultant generant un nou informe. A més, Sweave es un sistema independent i es pot utilitzar amb qualsevol editor de text. El més comú és combinar-lo amb Latex, un editor que ofereix un gran ventall de possibilitats a l’hora de dissenyar informes. [14, 15]

Aquestes funcionalitats que ofereix es deuen als arxius noweb utilitzats. Aquests són els responsables de poder combinar el codi font de programes i la documentació corresponent dins un mateix arxiu. Per aquest motiu es important comprendre el seu funcionament, ja que el treball que realitza es basa en aquesta idea . [14, 15]

Un arxiu noweb es un arxiu de text simple format per *chunks*, és a dir, segments de text ordinari o codi de programació. Els primers es troben indicats amb el signe @ seguit d’un salt de línia com es mostra a continuació: [14, 15]

@

En aquest cas es mostra un exemple de text ordinari

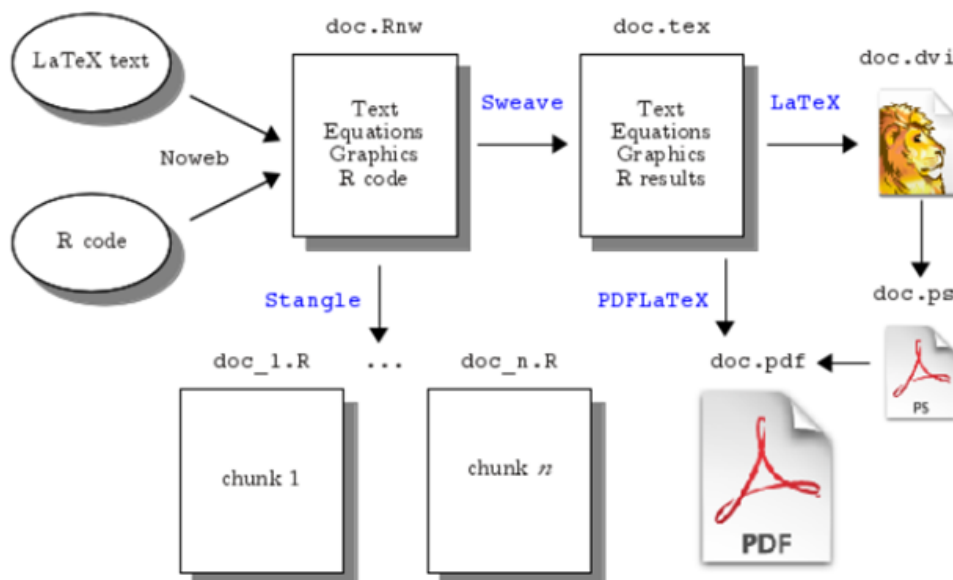
Els segons s'indiquen amb <<name>>= seguit d'un salt de línia. Cal tenir en compte que dins l'espai que ocupa "name" és on s'especifiquen les característiques que tindrà el codi en el document final (si es vol que aparegui el codi o el resultat, les característiques de les imatges o gràfics resultats, etc.). A continuació es mostra un exemple: [14, 15]

```
<<exemple, echo=T, warning=F>>=  
n=2+3
```

Finalment, també existeix la possibilitat de crear comentaris que només es visualitzen en la pantalla de codi i no queden reflectits en l'informe final. Aquests s'assenyalen amb el signe % a l'inici del comentari com es veu a continuació:

```
% Exemple d'un comentari
```

L'extensió habitual d'aquests arxius és .nw però en aquest projecte ens interessarà l'extensió .Rnw que és la que reflexa la relació amb R. A la següent imatge (imatge 4) es mostra de forma gràfica tot el procés que es segueix fins a la obtenció del document final. [14, 15]



Imatge 4: Diagrama d'arxius per treballar amb Sweave [14]

### 2.3. Anàlisi de microarrays

Com ja s'ha comentat a l'inici de la memòria, per a la determinació dels nivells d'expressió dels gens es du a terme un estudi de l'ARN, conegut com transcriptòmica, que ha contribuït en gran mesura als avanços relacionats amb la patogènesis de les malalties. [17,25]

La principal tècnica que s'utilitza avui en dia per aquest tipus d'anàlisi és l'ús de microarrays. Aquest consisteix en un aparell que permet l'estudi simultani de múltiples unitats (gens, proteïnes o metabòlits) sobre un substrat sòlid que pot ser cristall, plàstic o sílice, i es sotmeten a l'acció de molècules diana, expressió de les quals es vol analitzar. [17]

L'auge d'aquesta tècnica es deu, principalment, a la gran quantitat de mesures simultànies que permet realitzar. Per aquest motiu, avui en dia, es considera una metodologia consolidada que permet l'estudi de molts gens a l'hora, el que ha permès una nova manera d'estudiar el transcriptoma, en especial l'expressió gènica. Degut a les seves característiques, l'ús de microarrays obre un gran ventall d'aplicacions, no només en l'estudi de malalties, el càncer o la resposta a medicaments, sinó també en molts altres àmbits de la biologia com ara els ritmes circadians de les fruites. [17]

El seu funcionament es basa en la detecció, per mitjà de fluorescència, de la hibridació d'una sonda específica, anomenada *probe*, i la molècula diana o *target*. Un escàner s'encarrega de llegir els nivells de fluorescència obtinguts que seran representatius de la quantitat de molècules diana presents en la mostra problema. Així, podem classificar els microarrays segons el nombre de mostres simultànies que s'hibriden durant el procés: [17,25]

- **Microarrays d'un color o arrays de oligonucleòtids:** només s'hibrida una mostra, pel que, només s'utilitza un marcador fluorescent. Així, per a cada array, s'obté directament una mesura numèrica d'expressió absoluta a partir de la lectura de la intensitat de la fluorescència resultant.

- **Microarrays de dos colors o spotted arrays:** basats en la hibridació competitiva de dues mostres, cada una marcada amb un marcador fluorescent diferents que acostumen a ser Cy3 (fluorescència verda) i Cy5 (fluorescència vermella). Un cop s'ha dut a terme la hibridació, mitjançant un làser s'il·lumina l'array provocant que el marcador emeti fluorescència d'un color o l'altre, el que generarà dues imatges que es superposaran per poder realitzar un anàlisi conjunt. Així s'obté un nivell d'expressió relatiu, ja que el valor final representa el nivell d'expressió d'una mostra respecte a l'altre.

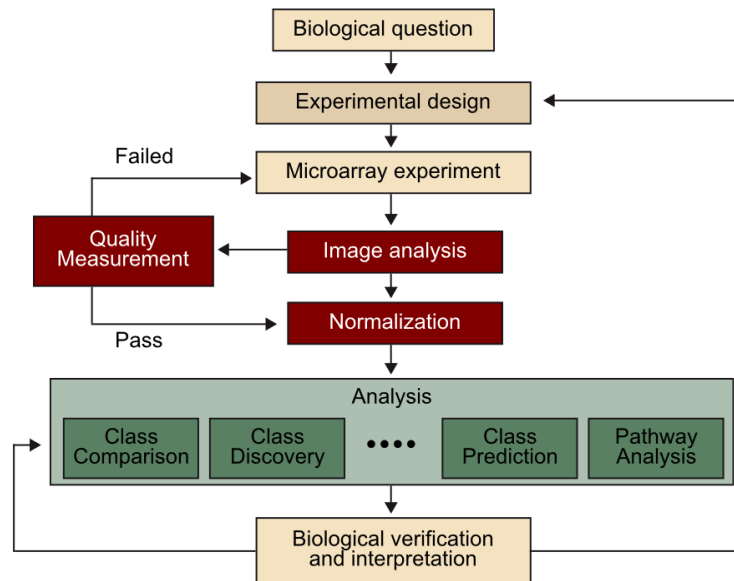
Un cop obtinguts els resultats dels microarrays, aquests han de ser analitzats, per això és necessari recórrer a l'ús d'eines bioinformàtiques o softwares lliures que permetin el processament de grans bases de dades, entre elles TM4, SPSS o Bioconductor. Aquest darrer és l'eina a la que es troba enfocat aquest treball, ja que és un projecte de codi obert per a l'anàlisi de dades de genòmica, que conté un repositori de paquets de R i té com objectiu desenvolupar i integrar un software per a l'anàlisi estadístic de dades de laboratori en biologia molecular, especialment l'anàlisi de microarrays.

Finalment, l'anàlisi de dades de microarrays (MDA) ha de seguir el mètode científic i es desenvolupa de forma ordenada seguint els següents punts: [6]

- Definició del disseny experimental
- Les dades obtingudes, també conegudes com *raw data*, es sometent a controls de qualitat.
- En cas de que la qualitat de les dades sigui acceptable es podran analitzar. Per fer-ho es poden seguir diferents formes de processament o transformacions que es comprenen com a normalització.

- Les dades normalitzades són utilitzades per als anàlisis estadístics especificats en el disseny experimental.
- Finalment, els resultats s'utilitzen com a base per a la interpretació biològica de l'experiment.

Aquest procediment es resumeix en la següent imatge (imatge 5)



Imatge 5: Ordre lògic en l'anàlisi de dades de microarrays [6]

## 2.4. Anàlisis estadístics

L'anàlisi estadístic té com a objectiu l'estudi de les dades obtingudes mitjançant mètodes estadístics, un cop han estat sotmeses al control de qualitat i normalització, per determinar aquells gens que es troben diferencialment expressats. [6]

La selecció de gens diferencialment expressats es tradueix en la comparació de variables que, en gran part dels casos, no resulta senzill determinar el punt de tall a partir del qual determinar diferència entre elles. Per aquest motiu s'han desenvolupat un gran nombre de mètodes estadístics per resoldre-ho. La majoria són extensions dels mètodes



estadístics clàssics com la prova *t-student* o l'anàlisi de la variància, adaptats per tenir en compte les característiques dels microarrays. [25]

Cal tenir en compte que, en el cas dels microarrays, el fet que hi hagi dues tecnologies que mesuren la expressió de dues formes diferents fa que s'hagi de diferenciar la metodologia a emprar en cada cas. És necessari recordar que els arrays de dos colors combinen dues mostres en un xip i generen una mesura d'expressió relativa. Això fa que, per comparar dues mostres d'un mateix individu, siguin l'opció naturalment més apropiada. Però en el cas de voler comparar mostres independents de diferents individus s'utilitzen els arrays d'un color, que generen una mesura d'expressió absoluta. [25]

## **2.5. Selecció i validació de biomarcadors**

L'objectiu de la selecció i validació de biomarcadors és la construcció i validació d'un classificador que, amb la unitat a estudiar seleccionada, permeti diferenciar entre dos o més grups amb la major probabilitat de encert possible. No s'ha de confondre la construcció d'un classificador amb la selecció de variables significativament diferents. En aquestes últimes el que es busca és un valor a partir del qual prendre decisions per considerar si es tracta de grups significativament diferents o no. En la construcció de classificadors, en canvi, tot i que també es consideren variables amb valors distints, es busquen variables que es combinin d'alguna manera, donada per el "classificador", que permeti utilitzar-les per diferenciar els grups amb la menor probabilitat d'error possible i de forma que segueixi sent vàlid amb noves dades independents. [7, 25]

Així l'anàlisi estadístic que s'utilitza per a la selecció de biomarcadors es du a terme a diferents nivells de validació creuada amb l'objectiu d'evitar el sobreajustament que molts cops es dona degut a una sobrevaloració de les probabilitats de classificació associades al esmentats marcadors. Aquesta tècnica de validació creuada s'anomena LOOCV (Leave One Out Cross Validation). [7, 18]

Amb les unitats a estudiar seleccionades es construeix un classificador utilitzant distintes aproximacions, ja que no existeix la certesa que un mètode sigui millor a un altre per a tots els problemes. El resultat d'aquest procés és una estimació de la qualitat de cada combinació que permet seleccionar els biomarcadors més adients en cada cas i la millor forma de combinar-los per diferenciar els grups de l'estudi. [7, 18]

Un cop seleccionats els classificadors més adients i el millor procediment de classificació, aquests cal avaluar-los sobre un conjunt de proves independents per obtenir una nova estimació de les probabilitats dels classificadors. [7, 18]

## 3. Desenvolupament del projecte

### 3.1. Requeriments bàsics

Com ja s'ha avançat, en el projecte s'ha utilitzat Laravel [3], descrit més endavant, per dur a terme el desenvolupament del llenguatge PHP. A continuació es descriuen les passes que s'ha hagut de seguir per a la seva instal·lació, que resulta clau per al projecte.

1. Obtenir la versió 5.6 o superior de PHP
2. Instal·lació de Composer, una eina que utilitza Laravel per a la gestió de dependències en PHP i que permet gestionar i administrar les biblioteques utilitzades en el projecte. [9]
3. Afegir PHP i Composer com a directoris al path, en aquest cas, de Windows. En cas de Linux i MAC només cal afegir Composer al directori “/usr/local/bin” degut a que PHP ja ve inclòs.
4. Instal·lació i creació d'un nou projecte de Laravel seguint els passos de la propia web [28].
5. Dins la carpeta on es troba Laravel, executar la comanda “composer install” a la consola de l'ordinador per tal d'instal·lar els paquets de Laravel necessaris per al seu funcionament.
6. Executar la comanda “composer update”, de nou a a la consola de l'ordinador, per actualitzar la versió més moderna dels paquets de Laravel obtinguts en la passa anterior.
7. Configuració de l'arxiu .env de Laravel accedint des de l'editor de text. Cal modificar la ruta de la base de dades (normalment la IP) i el seu nom. A més, també s'ha d'especificar l'usuari i contrasenya per accedir a la base de dades en qüestió (imatge 6).

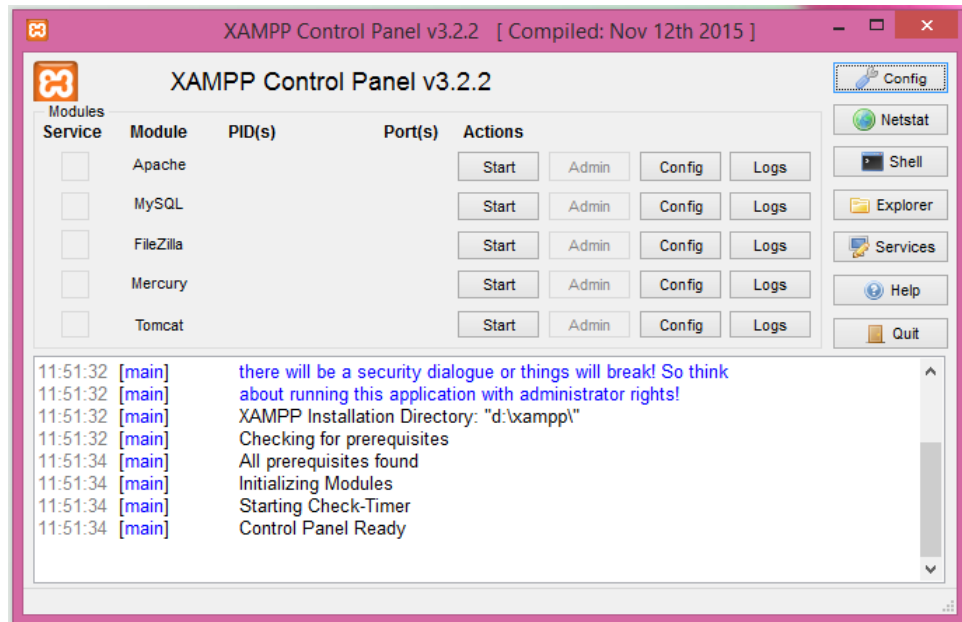
```
1 APP_ENV=local
2 APP_KEY=base64:Bkaf166kspKrTlW00rpjXm8W0DWGFnnvooDpTvRxaFE=
3 APP_DEBUG=true
4 APP_LOG_LEVEL=debug
5 APP_URL=http://localhost
6
7 DB_CONNECTION=mysql
8 DB_HOST=localhost
9 DB_PORT=3306
10 DB_DATABASE=plantillas
11 DB_USERNAME=root
12 DB_PASSWORD=
13
14 BROADCAST_DRIVER=log
15 CACHE_DRIVER=file
16 SESSION_DRIVER=file
17 QUEUE_DRIVER=sync
18
19 REDIS_HOST=127.0.0.1
20 REDIS_PASSWORD=null
21 REDIS_PORT=6379
22
23 MAIL_DRIVER=smtp
24 MAIL_HOST=smtp.mailtrap.io
25 MAIL_PORT=2525
26 MAIL_USERNAME=null
27 MAIL_PASSWORD=null
28 MAIL_ENCRYPTION=null
29
30 PUSHER_APP_ID=
31 PUSHER_APP_KEY=
32 PUSHER_APP_SECRET=
33
```

*Imatge 6: Modificació de l'arxiu .env de Laravel amb l'editor de text Atom un cop creada la base de dades*

8. Executar la comanda “php artisan key:generate” a la consola de l'ordinador per al xifrat de contrasenyes i claus utilitzades internament per Laravel.

Per altra banda, per poder dur a terme el desenvolupament local del programa, és necessari un servidor web. En aquest cas s'ha utilitzat Xampp, que és l'habitual per al sistema operatiu de Windows. Aquest es tracta d'un sistema d'instal·lació de servidor web de software lliure que, a més, també permet la gestió de base de dades [4]. D'entre les diferents opcions que permet utilitzar aquesta eina (Imatge 7) s'han utilitzat les següents:

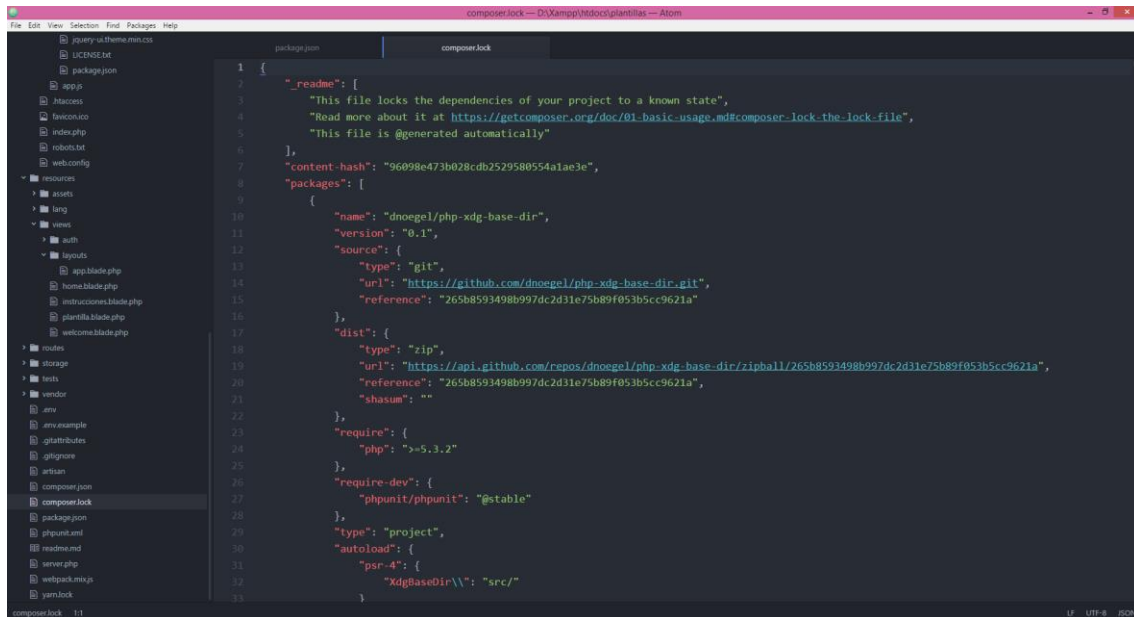
- Apache: servidor web HTTP de codi obert [5].
- MySQL: sistema de gestió de bases de dades amb llicència GPL (General Public License). [1]



Imatge 7: Captura de pantalla de l'eina Xampp

Per la seva instal·lació, tan sols és necessari descarregar l'arxiu corresponent de la pròpia pàgina de descarrega [26] i seguir el procediment habitual. Cal tenir en compte que en altres sistemes operatius, com ara MAC, s'utilitza MAMP, una versió optimitzada semblant a Xampp creada específicament per a l'esmentat sistema operatiu. En cas de Linux no existeix una eina concreta i, per tant, s'han de realitzar les instal·lacions d'Apache i MySQL de forma independent.

Referent al codi de programació, només és necessari un editor de text pla per al seu desenvolupament. Aquest pot ser Notepath++, Bloc de notes o Eclipse entre molts d'altres. En aquest cas s'ha utilitzat Atom (imatge 8), un editor obert, i per tant gratuït, de codi font amb el que s'han desenvolupat tots els llenguatges necessaris per a la creació de la pàgina web (PHP, HTML, CSS i JavaScript). Per a la seva instal·lació, tan sols s'ha de descarregar l'arxiu corresponent de la pàgina de descarrega [27] i seguir el procediment habitual.



Imatge 8: Captura de pantalla de l'eina Atom

Per altra banda, per a dur a terme la maquetació de la web, en aquest projecte s'ha decidit utilitzar Bootstrap [10], un *framework*, també de codi obert, utilitzat per al disseny web. Aquest no requereix instal·lació, tot i que si és necessari incloure la llibreria de Bootstrap a l'HTML, com es mostra en la segona línia de la imatge 9, per poder-lo utilitzar.

```
<!-- Styles -->
<link href="{{ asset('css/app.css') }}" rel="stylesheet">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
integrity="sha384-BViiSIeK1dGmJRAKycuHAHRg320MjUcw7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">
<link rel="stylesheet" href="{{ asset('js/datepicker/jquery-ui.css') }}">
<link href="{{ asset('css/plantilla.css') }}" rel="stylesheet">
```

Imatge 9: A la segona línia del codi HTML s'inclou la llibreria de Bootstrap.

A més, com a editor de text s'ha decidit utilitzar TinyMce [11], un editor de codi HTML utilitzat al final del projecte que converteix el codi de textos HTML en format estàndard. A més ha permès aplicar l'opció de modificar el contingut de les plantilles. Per al seu ús senzillament s'ha inclòs la llibreria a l'HTML, com es pot veure a la primera línia de la imatge 10.

```
<script src="https://cloud.tinymce.com/stable/tinymce.min.js"></script>
<script type="text/javascript">
    function readURL(input) {
        if (input.files && input.files[0]) {
            var reader = new FileReader();
            reader.onload = function (e) {
```

Imatge 10: A la primera línia del codi HTML s'inclou la llibreria de TinyMce

Cal tenir en compte que, junt amb la memòria també es realitza l'entrega d'un arxiu adjunt on es troba la carpeta de Laravel amb el projecte realitzat. Si es descarrega l'arxiu i es mou a la carpeta del servidor web (en el cas de Xampp a la carpeta "C:\Xampp\htdocs") no és necessari dur a terme totes les passes descrites en la instal·lació de Laravel. En aquest cas, els passos 4, 5, 6 i 8 no són necessaris, tot i que el penúltim es recomanable. A més, a la passa 7 tan sols és necessari modificar la ruta de la base de dades.

Finalment, per a l'ús de l'aplicació cal tenir migrada la base de dades, tal i com es descriu a la documentació del projecte que es troba a l'annexa, tot i que, en cas d'haver descarregat l'arxiu, només cal executar la comanda "php artisan migrate --seed" a la consola.

### 3.2. Desenvolupament de la base de dades

Per a la creació de la base de dades es va decidir utilitzar el programa MySQL™ [1]. Es tracta d'un sistema de gestió de base de dades, marca registrada d'Oracle Corporation i les seves filials, que ofereix una un servidor molt ràpid i robust de dades SQL (Structured Query Language). Per poder treballar amb MySQL s'ha utilitzat MySQL Workbench [2], una eina gràfica creada expressament per treballar amb servidors i bases de dades MySQL. La seva funcionalitat es basa en cinc àrees de treball:

- **Desenvolupament de SQL:** permet crear i gestionar les connexions als servidors de bases de dades a més de configurar els paràmetres de connexió. Així, aquesta eina proporciona la capacitat d'executar consultes SQL en les connexions de base de dades utilitzant l'editor predefinit de SQL.
- **Modelatge de dades:** permet dissenyar o crear models de la base de dades de forma gràfica, oferint la possibilitat de modificar l'estructura esquemàtica de la base de dades en viu, i editar tots els seus aspectes amb l'Editor de taules. Aquest últim ofereix

instal·lacions de fàcil ús per a l'edició de taules, columnes, índexs, disparadors, creació de particions, opcions, insercions i privilegis, rutines i vistes.

- **Administració del servidor:** permet administrar instàncies del servidor MySQL mitjançant l'administració d'usuaris, la realització de còpies de seguretat i recuperació, la inspecció de les dades d'auditoria, la visualització de l'estat de base de dades, i el seguiment del rendiment del servidor MySQL.
- **Migració de dades:** permet migrar de Microsoft SQL Server, Microsoft Access, Sybase ASE, SQLite, SQL Anywhere, PostgreSQL, i altres taules RDBMS (Relational Database Management System), objectes i dades a MySQL.
- **MySQL Enterprise Support:** ofereix suport per a productes empresarials com MySQL Enterprise Backup, MySQL Firewall i auditoria de MySQL. En aquest cas, no ha estat necessari l'ús d'aquesta funcionalitat ja que no corresponia al tipus de projecte que s'ha desenvolupat.

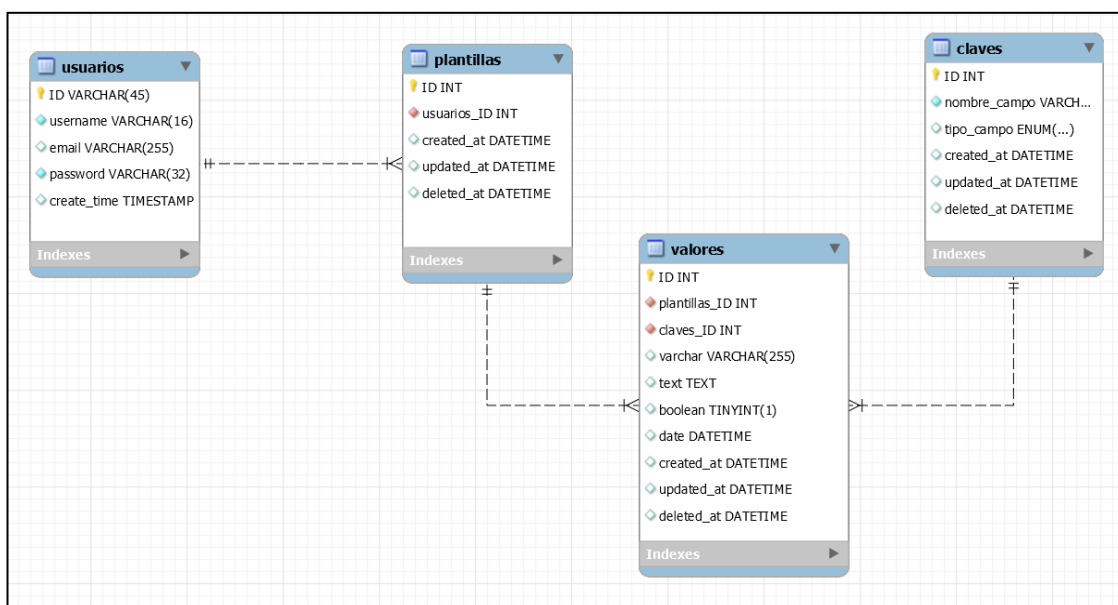
Cal destacar que, en aquest cas, s'ha utilitzat l'edició *Community* de MySQL Workbench ja que oferia tot el necessari per al disseny de la base de dades de forma gratuïta.

A la següent imatge (imatge 11) es representa gràficament l'estructuració de cada taula i la seva relació amb l'entorn. Com es pot veure, la base de dades d'aquest projecte està formada per quatre taules:

- **Usuarios:** fa referència a la identificació de l'usuari que utilitzarà la web per realitzar la plantilla.
- **Plantillas:** necessària per a la identificació del tipus de plantilla que, relacionada amb la primera taula permet la personalització i preservació d'aquestes.



- **Valores:** fa referència al contingut de cada plantilla. Aquesta taula també es troba relacionada amb “plantillas”, el que permetrà crear l’informe desitjat, i per altra banda es troba relacionada amb “claves”, necessari per a la identificació dels seus camps.
- **Claves:** necessària per a la identificació dels camps i organització de “valores” per aquest motiu es troba relacionada amb aquesta última.



Imatge 11: Esquema organitzatiu de les taules que componen la base de dades utilitzada.

Si bé estava pensat que l’estructura de la base de dades estigués formada per les diferents taules específiques per a cada tipus d’informe, finalment es va optar per aquesta composició aparentment més senzilla però que suposa una major dificultat en l’elaboració de la web. Aquesta estructura és el resultat de l’estudi i anàlisi dels processos que ha de dur a terme el programa. Així s’aconsegueix una plantilla més flexible i totalment gestionable per l’usuari de manera que es pot introduir qualsevol tipus de dada disminuint els impediments, ja que el nombre de creació de nous camps no es veu limitat. D’haver creat la base de dades amb l’altre sistema de taules, l’estructura no hagués permès reestructuracions i per tant els camps sempre serien els mateixos.

### 3.3. Desenvolupament de l'entorn web

Com ja s'ha comentat en punts anteriors, l'ús de Laravel [3] resulta clau per a la creació de l'entorn web en PHP. Aquest, és un *framework* de codi obert per al desenvolupament web i d'aplicacions en llenguatge PHP. [12]

El motiu pel qual s'ha decidit utilitzar aquest *framework* és la capacitat que dona de facilitar les tasques comunes que s'utilitzen en la majoria dels projectes web, com ara: [12]

- Mecanisme de rutes ràpid i simple.
- Múltiples back-ends per a la sessió i emmagatzematge.
- ORM (Object-Relational mapping) de base de dades intuïtiva.
- Migracions de bases de dades.

Així doncs, es important tenir en compte l'ús de Laravel per al projecte, ja que facilita en gran mesura l'escriptura del llenguatge PHP. Tot i així, un correcte domini del PHP resulta essencial per poder desenvolupar totes les funcions d'aquest projecte.

PHP (Hypertext Preprocessor) és un llenguatge de codi obert Orientat al desenvolupament d'aplicacions web dinàmiques amb accés a informació emmagatzemada en una base de dades i que pot ser incrustat en HTML.

El principal motiu pel qual s'ha usat aquest llenguatge és la seva simplicitat, que permet crear aplicacions complexes amb una corba d'aprenentatge molt curta oferint, al seu torn, moltes característiques avançades com aplicacions de contingut dinàmic. A més, PHP pot ser desplegat en la majoria dels servidors web i en gairebé tots els sistemes operatius i plataformes sense cost algun. També permet la connexió a diferents tipus de servidors de bases de dades entre ells MySQL, utilitzat en aquest projecte. [13]

Cal destacar que, el fet de poder emprar-se en tots els sistemes operatius principals implica l'admissió de la majoria de servidors web d'avui dia, incloent Apatxe, IIS, i molts altres. D'aquesta manera, amb PHP, es té la llibertat de triar el sistema operatiu i el servidor web que es desitgi. [13]

Per altra banda, amb aquest llenguatge no s'està limitat a generar HTML. Entre les capacitats de PHP s'inclouen la creació d'imatges i fitxers PDF generades sobre la marxa. També es pot generar fàcilment qualsevol tipus de text, com XHTML i qualsevol altre tipus de fitxer XML, el que dona la possibilitat de crear el document en format RNW que es persegueix en aquest projecte. A més, es possible autogenerar aquests fitxers i guardar-los en el sistema de fitxers en comptes d'imprimir-los en pantalla, creant una *caché* per a contingut dinàmic.

La simbologia requerida per indicar l'ús d'aquest llenguatge és el següent:

```
<?php
Codi desitjat
php>
```

Totes les funcions que es desitgin dur a terme han de trobar-se entre el codi especificat, sempre seguint la normativa del llenguatge PHP, per a que es llegeixi com a tal.

Per altra banda, com ja s'ha anat esmentant en aquest apartat, el llenguatge PHP es pot combinar amb HTML. Aquest últim, també ha estat utilitzat en el desenvolupament del projecte per al disseny de la plana que es mostra a l'usuari. [13]

L'HTML (Hyper Text Markup Language) fa referència al llenguatge utilitzat para a l'elaboració de pàgines web. Els seus creadors el defineixen com "el llenguatge de marcat estàndard utilitzat per crear pàgines web i els seus elements formen els blocs de construcció de tots els llocs web." [19]

Així, és possible dissenyar la pròpia pàgina web a partir de les funcionalitats que ofereix. Per poder-lo utilitzar és necessari fer ús de les etiquetes emmarcades amb els símbols << ... >>, que indiquen diferents funcions. Gràcies a aquestes és possible estructurar, fins a un cert punt, l'aparença d'un documento.

HTML consta de varis components que permeten crear diferències entre els diferents elements gràcies, sobre tot, als seus atributs. Per aquest motiu, és important conèixer el funcionament d'aquest llenguatge. [13]

Els elements són l'estructura bàsica de l'HTML. Un element generalment té una etiqueta d'inici i tancament i un atribut en l'etiqueta d'inici. El valor que es doni a l'atribut marca la funcionalitat de l'element i el distingeix de la resta. A continuació es mostra un exemple:

```
<nom_de_l'element atribut="valor">
```

```
Codi del contingut desitjat
```

```
</ nom_de_l'element >
```

Tot i aquesta normativa, alguns elements ja contenen funcions predefinides, per exemple <br>, que crea un espai. Aquests no tenen contingut, ni atribut, ni porten una etiqueta de tancament.

Finalment, també es possible afegir una classe. Aquestes tan poden venir predefinides pel propi codi però també és possible crear una classe definida per un mateix. En aquest punt és on entra a formar part el llenguatge CSS. [20]

Aquest últim ha estat creat per World Wild Web Consorce, els mateixos creadors de HTML. El CSS (*Cascading Stylesheets*) és un llenguatge de disseny gràfic que complementa a l'HTML. A partir d'aquest és possible especificar la posició del text, els marges de les figures que apareixen a plana web etc. El codi especificat per aquest és el següent:

```
.nom_de_la_classe{
```

```
Característiques }
```

Així, amb les classes que definides amb CSS, és possible modelar el contingut dels elements escrits en llenguatge HTML afegint-la a l'etiqueta d'inici. La següent imatge (imatge 12) representa el llenguatge HTML i CSS descrits:

```
<div class="col-sm-7 positiontext">
  <div class="panel panel-default">
    <input id="exporthidden" type="hidden" name="export" value="1" disabled>
    <div align='center' class="panel-heading estilo">{!!$plantilla->nombre!!}</div>
    <div align='center' class="panel-body">
      @if (!empty($plantilla->logo))
        
      @else
        <img class="sinpie" width="200" height="100" src=""/>
      @endif
      <br>
      <label for="imagen">Adjuntar logo de la empresa o laboratorio:</label>
      <input name="archivo1" type="file" id="archivo1"><br/>
    </div>
  </div>
```

Imatge 12: Exemple del llenguatge HTML i CSS.

A més, també s'ha usat, de forma molt puntual, el llenguatge Javascript [21], un llenguatge orientat a objectes que ha permès donar funcionalitat als botons inclosos en el projecte, com ara el botó "Guardar". Per utilitzar-lo es suficient amb escriure el seu llenguatge entre la següent etiqueta:

```
<script> Codi <\script>
```

Si a aquest se li dona una identitat o un nom és possible relacionar el codi Javascript amb HTML introduint un atribut en aquest darrer que contengui el mateix nom.

Finalment, per resoldre tots els dubtes de codi de programació i trobar el codi requerit per satisfer les funcionalitats necessàries, s'ha consultat la web de consultes Stackoverflow [22] i Github [23] a més de les pàgines oficials per a cada llenguatge.

### 3.4. Incidències en les activitats realitzades

Durant el temps programat per a la documentació de la base de dades, a més d'aprendre el seu funcionament, es va conèixer l'existència de MySQL Workbench, una eina visual de disseny de base de dades que integra el desenvolupament software, és a dir, el disseny, la creació, l'administració i manteniment de la base de dades. Per aquest motiu, un cop finalitzat el procés de documentació es van poder realitzar les tasques de disseny i creació de base de dades a l'hora, simplificant notablement el temps programat per la creació de la base de dades.

A més, es va programar la realització del primer informe sobre el desenvolupament del treball abans de la creació de base de dades, el que no permetia una correcta autoavaluació del avanç del projecte, ja que totes les activitats anteriors eren més bé teòriques, el que dificulta la valoració crítica dels resultats.

Per tal de mitigar el efectes descrits en l'apartat anterior es van reorganitzar la tasca de creació de la base de dades, amb una duració prevista de 28h, i la tasca de realització de la PAC 2.

Aquesta reorganització va consistir en intercanviar ambdues temporitzacions de les tasques de manera que la creació de la base de dades ha tingut lloc en la setmana en la que estava prevista la realització d'aquest informe i viceversa. D'aquesta manera, per una banda, el disseny i la creació de la base de dades passaven a ser consecutives, mantenint el procés lògic segons la nova situació després del descobriment de l'eina Workbench. Per altra banda, es solucionava l'incident amb la segona prova d'avaluació continua ja que, al ser posterior a la creació de la base de dades s'ha pogut avaluar de forma més eficient el desenvolupament del treball a partir dels resultats obtinguts, exposats en el punt 4 d'aquest mateix informe, que de l'altre manera no hagués estat possible.

Per altra banda, durant el temps programat per a la documentació i aprenentatge de HTML i CSS també es va desenvolupar el llenguatge

PHP, ja que l'eina Atom permet dur a terme la construcció de la pàgina web amb els llenguatges PHP, HTML, CSS i SQL al mateix torn.

Quan es va realitzar la programació del projecte es va organitzar l'aprenentatge i ús dels diferents llenguatges per separat. A l'hora de realitzar el treball ha resultat més pràctic l'ús de tots els llenguatges a l'hora per tal d'anar finalitzant cada funcionalitat a mesura que s'anava creant. Per exemple, per crear la barra d'eines que apareix a la part superior de la pàgina web (imatge 11) es va utilitzar PHP per a la funcionalitat d'aquesta, HTML per crear l'objecte i CSS per afegir les característiques de disseny. En el cas d'haver fet aquesta mateixa feina en temps diferents hagués resultat més difícil organitzar el codi.

Cal destacar que la duració de l'activitat de documentació i aprenentatge es va prolongar més temps del que estava programat degut a la complexitat dels llenguatges. Tot i així, el poder treballar amb tots ells en la mateixa eina i al mateix temps va permetre compensar el temps excedit per l'activitat anterior utilitzant, això sí, tot el temps establert per a la creació i disseny de la web.

Així doncs, per tal de mitigar els efectes d'aquests contratemps, es va ajuntar el temps corresponent a l'aprenentatge dels diferents llenguatges per, posteriorment, dedicar el temps restant a la creació de l'entorn web. Així, va ser possible compensar temporalment els canvis realitzats en la planificació inicial.

Finalment, durant el temps restant, un cop entregat el segon informe sobre el desenvolupament del treball, estava planejat acabar el projecte i perfeccionar algunes característiques i funcionalitats.

En aquest període de temps va sorgir un imprevist en relació a l'exportació del document. Tot i que l'acció d'exportació es duia a terme sense cap dificultat, el format del text en que s'exportava va entrar en conflicte amb el programa R, utilitzat per obrir l'arxiu.

Així, en un primer intent es va voler crear l'arxiu amb el text en format UTF 8 (8-bit Unicode Transformation Format). Aquest és un format de codificació de caràcters que utilitza símbols de longitud variable [24]. El motiu d'utilitzar aquest format és a que es troba definit com a format estàndard i es una de les possibilitats de codificació reconegudes per Unicode, llenguatges web i R.

Per algun motiu que es desconeix R podia llegir l'arxiu a la perfecció però en canvi no el podia compilar amb Latex. Per tal d'intentar solucionar aquesta incidència es va realitzar la compilació amb diferents formats de lectura que R permet, però no va tenir sense èxit.

Posteriorment, també es va intentar solucionar des de R per mitjà de d'instal·lació de paquets per a la lectura de formats UTF 8, junt amb la combinació dels diferents llenguatges abans comentats. Malauradament, tampoc es va poder solucionar.

La única sortida i la més eficient va consistir en afegir algunes línies de codi en PHP al projecte de manera que, l'arxiu descarregat des de la pàgina web, s'obtingués en text pla. L'únic inconvenient d'utilitzar aquest sistema és que, el text pla que s'obre amb R no es llegeix com cal i, tant els caràcter accentuats com els característics de la nostra llengua, no es veuen com en un text ordinari.

Per altra banda, també es va incloure, en l'arxiu que es descarrega, el paquet UTF 8 comentant en línies anteriors per a que es realitzés correctament la lectura d'accents i caràcters llatins com la lletra ñ. D'aquesta manera, el document obtingut després de la compilació amb Latex mostra tots els caràcters representats correctament al text.



## 4. Assoliment del projecte.

Per avaluar l'assoliment d'aquest projecte és necessari considerar el grau de compliment dels objectius i les tasques proposades a l'inici del projecte. Cal recordar que els objectius plantejats en la planificació són els següents:

- Objectius generals:
  - Desenvolupar un generador automàtic de plantilles d'informes dinàmics en format Rnw.
  
- Objectius específics:
  - Crear una base de dades amb els diferents components per a tres tipus d'informe: anàlisi de microarrays, anàlisi estadístic i selecció i validació de biomarcadors.
  - Implementar l'eina en entorn web permetent la modificació del contingut i l'exportació de l'arxiu en format Rnw.

Com ja s'ha comentat anteriorment, a l'hora de crear la base de dades no hi ha hagut cap contratemps ni incidència i es van poder finalitzar les corresponents activitats abans del temps previst.

A l'hora de la implementació de l'eina en entorn web, si bé és veritat que va sorgir algun contratemps a l'últim moment, aquest no va impedir que es finalitzés el projecte a temps.

Així, tenint en compte el calendari d'activitats plantejat en la programació i les incidències comentades en el punt anterior, podem considerar que ambdós objectius específics s'han assolit correctament.

El correcte compliment dels objectius específics han permès que l'objectiu general, en el qual es basa la feina realitzada, es pogués assolir amb èxit, gairebé sense inconvenients.

Per altra banda, les tasques plantejades en la planificació són:

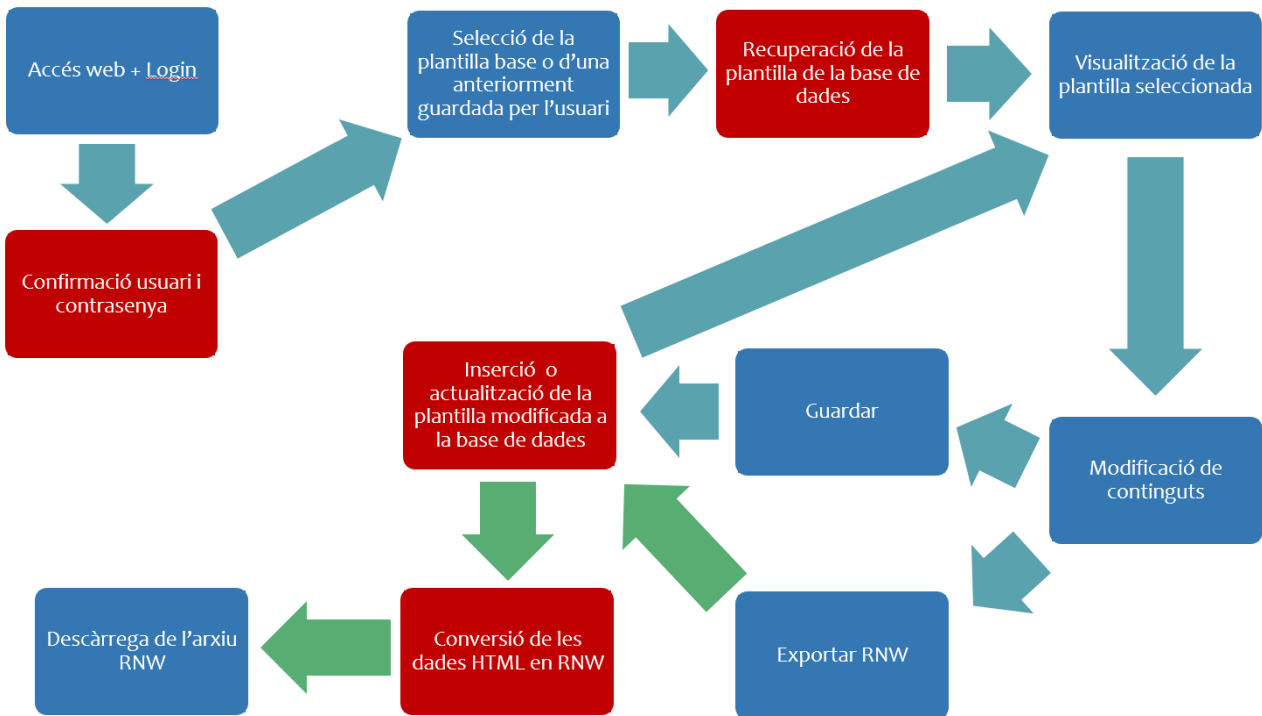
- Creació de la base de dades amb SQL
- Creació de l'entorn web amb HTML i maquetació amb CSS
- Programació web amb Laravel
- Entrega del projecte
- Entrega de la PAC 1
- Entrega de la PAC 2
- Entrega de la PAC 3
- Entrega de la Memòria del TFM
- Realització de la defensa del TFM

Com ja s'ha comentat en el punt anterior de les incidències, finalment es va decidir ajuntar la segona i tercera tasca per tal de facilitar el projecte. Tot i aquest petit canvi, la resta d'entregues sempre han estat realitzades dins del temps programat a falta de l'entrega de la present memòria i la realització de la defensa que sembla que també podran ser entregades dins el límit de temps establert.

Així doncs, és just considerar que les tasques proposades s'han realitzat correctament segons la planificació establerta complint amb els terminis exigits.

## 5. Resultats obtinguts

Com a resultat de tot el procés de desenvolupament s'ha obtingut una pàgina web regida per el següent flux de treball (imatge 13):



Imatge 13: Flux d'informació de la pàgina web. En color blau es representen les interaccions de la web amb l'usuari i de color vermell els processos que es desenvolupen en segon pla.

En primer lloc, s'entra a la web i s'accedeix al compte d'usuari introduint el correu electrònic i la contrasenya amb la que s'ha dut a terme el registre. Aquestes dades són corroborades per el controlador d'autenticació de Laravel que, en cas de ser correctes, permeten l'accés al contingut de la web (imatges 18,23,24,25 i 27 de l'annexa).

En segon lloc, l'usuari selecciona un dels tres tipus de plantilles que es proposen ("Microarrays", "Biomarcadores" o "Estadística") i, un cop ha accedit a la plana de modificació d'aquestes, té l'opció de seleccionar, de la llista lateral, una de les plantilles anteriorment creades i guardades per ell mateix. Aquesta selecció inicia un procés de recuperació de la plantilla que es troba a la base de dades a partir d'identificadors únics

per a cadascuna d'elles. Així, es possible la visualització de la plantilla seleccionada amb una estructuració que imita el format resultat de l'ús de Sweave per a la creació d'informes (imatges 28 i 30 de l'annexa).

Per tal de poder realitzar canvis a la plantilla al gust de l'usuari, s'ha construït amb HTML tot un seguit de camps editables o "inputs" que permeten modificar els elements existents i la inserció d'imatges (imatges 33, 35 i 37 de l'annexa).

A continuació, es pot escollir la opció de "Guardar" o la de "Exportar a RNW". En ambdós casos es realitzarà un guardat de la plantilla modificada a través de la inserció de la nova plantilla a la base de dades o, en cas que sigui una plantilla prèviament creada per l'usuari, es duu a terme una actualització de les dades del registre pertinent (imatges 39 i 40 de l'annexa).

En cas d'haver escollir l'opció de guardar, l'usuari es redirigit cap a la visualització de la plantilla modificada. Per altra banda, l'opció d'exportació a RNW inicia un procés de conversió de les dades HTML en format RNW, mitjançant tota una sèrie de procediments que transformen les cadenes de text amb el format introduït per l'usuari (negreta, cursiva, llistats, etc.) en l'equivalent d'aquests formats per al llenguatge Sweave. Si el procés s'ha dut a terme amb èxit, l'usuari obté un arxiu amb extensió RNW que es descarrega automàticament i es pot executar amb R (imatges 43, 44 i 45 de l'annexa).

## 6. Conclusions

Aquest projecte ha permès donar a conèixer el lent avanç del sector científic en el desenvolupament tecnològic en comparació amb la resta de sectors i la importància de rompre amb aquesta situació actual. Una automatització dels procediments bàsics permetria una gran optimització de recursos tant a nivell econòmic, material com temporal per als investigador o l'equip de treball. A més, resultaria en una millora de l'eficiència i eficàcia dels mètodes utilitzats.

A més, la creació d'una aplicació web que permeti generar plantilles d'informes associats a un projecte, autogestionables per l'usuari i que doni lloc a un arxiu .Rnw, facilita la feina als treballadors, tant si tenen o no alts coneixements de programació. Aquest fet, repercuteix directament en una major agilització de les tasques relacionades amb l'elaboració de codis de programació i, per tant, en les tasques d'investigació.

Per altra banda, a nivell personal, degut al desenvolupament d'aquest projecte ha estat possible ampliar, en gran mesura, tots els coneixements necessaris de llenguatge web i base de dades. Si més no, ha estat gràcies als tutorials dels propis *frameworks* utilitzats i, en alguns casos, amb l'ajuda dels fòrums per resoldre dubtes, que ha estat possible assolir tots els objectius proposats. La correcta adequació de la metodologia, tot i les petites variacions en la planificació comentades en punts anteriors de la memòria, també han fet possible el seu compliment.

Finalment, aquest projecte, amb un major nombre de funcionalitats, podria donar molt més fruit. Tot i que el resultat obtingut és el que estava planejat, inclús incorporant alguna millora, com l'adició d'imatges, no s'ha pogut desenvolupar cap funció per a l'administració dels comptes d'usuari o per a l'adició de nous camps editables d'elements estadístics. Aquestes petites millores podrien ser línies de treball futur que no s'han pogut explorar en aquest treball i han quedat pendents.

## 7. Glossari

**Back-end:** és la part de programació dedicada a la manipulació de dades. Per tant, s'encarrega d'interactuar amb les bases de dades, sessions d'usuaris i allotjar la pàgina en un servidor. Els llenguatges que es solen utilitzar són PHP, Python, , Java, etc,

**Caché:** Emmagatzematge de dades.

**Chunks:** fragment d'informació

**Compilació:** traducció d'un llenguatge de programació a un altre diferent.

**Framework:** entorn de treball

**Front-end:** són totes aquelles eines que visualitza l'usuari, és a dir, les que es materialitzen en la web. Generalment el front-end es el formen els tres llenguatges principals utilitzats en aquest projecte.

**Inputs:** en HTML són aquells fragments de codi que permeten la introducció dades per part de l'usuari per a un posterior processament.

**Localhost:** domini de la màquina virtual local que identifica cada dispositiu segon el número del seu port (IP).

**Migrar:** en programació és la transferència entre base de dades d'una informació estructurada d'una determinada manera.

**Servidor:** eina que permet atendre les peticions de l'usuari i retornar-li una resposta.

**Sistema operatiu:** és el conjunt del programes que serveixen de pont entre l'usuari i l'ordinador, controlant-ne el funcionament. Els més coneguts són Windows i MAC.

**Software:** conjunt d'eines que permeten al programador escriure programes informàtics usant diferents llenguatges de programació.

## 8. Bibliografia

Totes les URL s'han consultat nombrosos cops al llarg del projecte, per aquest motiu es mostra una franja de temps en les que s'han dut a terme les consultes. Només aquelles visitades de forma ocasional s'especifiquen les dates

- [1] <https://www.mysql.com/> 07/03/2017, 14/03/2017
- [2] <https://dev.mysql.com/doc/workbench/en/> 07/03/2017-30/03/2017
- [3] <https://laravel.com/> 07/03/2017, 13/04/2017-15/05/2017
- [4] <https://www.apachefriends.org/es/index.html> 15/05/2017
- [5] <https://httpd.apache.org/> 15/05/2017
- [6] M. Carme Ruíz de Villa i Àlex Sánchez Pla. Análisis de datos de Microarrays. PID\_00192743, Universitat Oberta de Catalunya.
- [7] Kyle Strimbu i Jorge A. Tavel. What are Biomarkers?. Curr Opin HIV AIDS. 5(6): 463-466. Novembre 2010.
- [8] <https://atom.io/> 07/03/2017, 15/03/2017, 15/05/2017
- [9] <https://getcomposer.org/> 07/03/2017, 15/03/2017, 15/05/2017
- [10] <http://getbootstrap.com/> 09/04/2017, 15/05/2017
- [11] <https://www.tinymce.com/> 06/05/2017, 15/05/2017
- [12] <https://github.com/laravel/laravel> 15/05/2017
- [13] <http://php.net/manual/es/intro-what-is.php> 13/04/2017-15/05/2017
- [14] Francesc Carmona. Generación automática de informes con Sweave y Latex. Departamento de Estadística Universidad de Barcelona. Març 2012
- [15] Friederich Leish i R-core. Sweave User Manual. Novembre 2016

- [16] <https://www.r-project.org/> 15/05/2017
- [17] M. Carme Ruíz de Villa i Àlex Sánchez Pla. Preliminares. PID\_00192742, Universitat Oberta de Catalunya.
- [18] <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1184048/> Article
- [19] <https://www.w3.org/html/> 08/04/2017-15/05/2017
- [20] <https://www.w3.org/Style/CSS/Overview.en.html>  
08/04/2017-15/05/2017
- [21] <https://www.javascript.com/> 03/05/2017-15/05/2017
- [22] <https://stackoverflow.com/> 21/03/2017-13/05/2017
- [23] <https://github.com/> 21/03/2017-13/05/2017
- [24] <https://www.ecured.cu/UTF-8> 13/05/2017
- [25] Paula Díez et al. Data Analysis Strategies for Protein Microarrays. Microarrays(Basel). 1(2):64-83. Setembre 2012
- [26] <https://www.apachefriends.org/es/download.html> 15/03/2017,  
01/06/2017
- [27] <https://github.com/atom/atom/releases/tag/v1.17.2> 01/06/2017
- [28] <https://laravel.com/docs/5.4/installation> 15/03/2017, 03/06/2017



## 9. Annexes

### 9.1 Documentació del projecte

Per tal d'exposar el sistema de creació de la pàgina web de la forma més clara possible s'ha decidit comentar el codi utilitzat dels diferents llenguatges amb la respectiva imatge per a que es pugui consultar i facilitar-ne el seguiment. A més, el codi utilitzat també s'acompanya de les imatges representatives de la pàgina web.

En primer lloc, és necessari comentar la formació i migració de la base de dades que es va dur a terme amb llenguatge PHP tot seguint el tutorial de la web de Laravel [3]. Així, per descriure els camps d'una de les taules que conformen la base de dades (imatge 11), primer és necessari crear un arxiu en format .php on es desenvoluparà el codi necessari. En la imatge 14 es mostra el contingut de l'arxiu "Clave.php" amb el codi utilitzat per a la creació de la taula "claves". Com es pot veure, s'ha dut a terme una llista de components que ha de contenir aquesta taula, com ara la id, el nom del camp o el format d'aquest.

```
<?php

use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class Clave extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('claves', function (Blueprint $table) {
            $table->increments('id');
            $table->string('nombre_campo');
            $table->enum('tipo_campo', ['titulo', 'subtitulo', 'fecha', 'imagen', 'texto']);
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('claves');
    }
}
```

Imatge 14: Codi per a la creació de la base de dades de la taula "claves".

Un cop creades totes les taules amb el respectiu contingut, és necessari especificar la relació entre elles tal i com es representa a l'esquema planejat de MySQL en punts anteriors. Així, en un directori diferent, anomenat "Providers" i, de nou, amb llenguatge PHP, s'especifiquen les relacions existents entre les taules. En aquest cas, seguint amb la taula "claves", tal i com es va planificar, només es troba relacionada amb la taula "valores" de manera que una sola clau, per exemple "titulo", pot tenir molts valors (imatge 15).

```
<?php
namespace App;

use Illuminate\Database\Eloquent\Model;

class Clave extends Model
{
    protected $table = 'claves';
    protected $fillable = ['id','nombre_campo', 'tipo_campo', 'created_at','updated_at'];

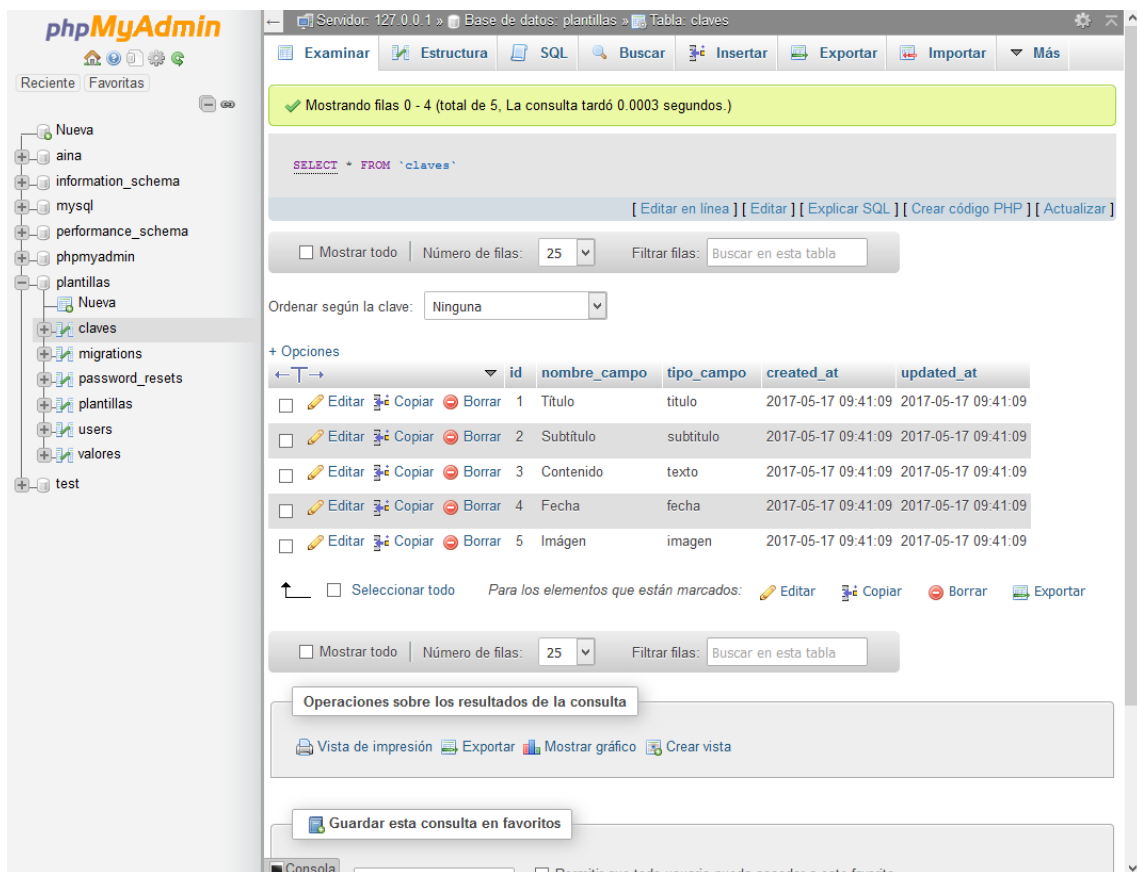
    public function valores()
    {
        return $this->hasMany('App\Valor');
    }
}
```

Imatge 15: Codi PHP per determinar les relacions entre les taules, en aquest cas entre la taula "claves" i la taula "valor"

Com es pot veure en el codi de la imatge anterior, en primer lloc es fa referència al nom de la taula que es tracta i tot seguit s'esmenta el seu contingut. És important, per evitar errors en la creació que, tant el nom com el contingut especificat en ambdós directoris, coincideixin a la perfecció.

Seguit de la descripció de la taula a la que es fa referència s'utilitza una funció mitjançant la qual es crea el llaç entre les taules. En el tutorial de Laravel 5.4 s'especifica de forma molt clara les diferents opcions de relació entre taules. En aquest cas, pels motius comentats anteriorment, interessa la opció "hasMany".

Un cop especificat el contingut de la taula i les relacions entre elles, es realitza una migració mitjançant la comanda “php artisan migrate --seed” a la consola de comanda, en aquest cas de Windows. Aquesta acció permet crear i modificar la base de dades des de l’editor de text. Per comprovar la correcta creació de la taula en qüestió és possible accedir a la base de dades del *localhost* per consultar-la (imatge 16).



Imatge 16: Consulta a la base de dades de les taules creades amb Laravel 5.4

Efectivament, la taula “claves” s’ha creat correctament amb el contingut especificat amb Laravel. A més, a la llista situada a l’esquerra de la imatge apareixen totes les taules esmentades en l’esquema especificat en el punt 3.2 d’aquesta memòria.

Cal tenir en compte que aquestes taules, en un principi, només consisteixen en els camps que les caracteritzen, pel que és necessari crear el contingut d’aquestes. Aquest contingut, tal com s’especifica en la planificació es va crear amb posterioritat, un cop realitzada la pàgina web.

Així doncs, finalment es du a terme la creació de continguts de les taules. Per realitzar-ho senzillament s’ha de crear un arxiu en la secció “seeds” on s’especifiquin els continguts (imatge 17).

```
<?php
use Illuminate\Database\Seeder;
use Carbon\Carbon;
use App\Clave;
use App\Valor;

class BiomarcadoresSeeder extends Seeder
{
    public function run()
    {
        $nombres=array(
            'Título',
            'Subtítulo',
            'Contenido',
            'Fecha',
            'Imágen',
        );
        $claves_tipo=array(
            'titulo',
            'subtitulo',
            'texto',
            'fecha',
            'imagen'
        );
        $valores=array(
            'Título',
            'Nombre',
            'Entidad u organismo que participa',
            Carbon::now('Europe/Madrid')->format('d/m/Y'),
            'Introducción'
```

En la imatge esquerra es mostra un fragment del codi utilitzat per a l’emplenament de les taules. El contingut de la funció “nombres” ha estat creat cara a l’exportació de l’arxiu en format RNW i no per a la introducció de dades, pel que aquest fragment no es detalla.

Així, a la funció “claves\_tipo” hi apareix el contingut de la taula “claves” format per “título” i “subtítulo” entre d’altres. A sota d’aquest, en la funció “valores” hi apareix l’inici del contingut de “valores”, on s’hi troba el text que es mostra a la web.

Imatge 17: Fragment de codi utilitzat per l’emplenament del contingut de les taules que conformen la base de dades creades anteriorment.

Un cop descrita la creació de la base de dades, falta visualitzar-ho a la web. A continuació es detalla el procediment realitzat i el codi utilitzat, de forma general, per a la creació de l’entorn web a través del qual l’usuari interactua amb l’eina creada.

Per poder visualitzar la plana web al llarg de la seva creació, de nou s’accedeix al *localhost*. A partir d’aquest es realitzaran les captures de pantalla necessàries que acompanyaran al codi i a la descripció d’aquest.

La primera plana que es mostra al accedir a la web és la pàgina inicial de benvinguda (imatge 18).



Imatge 18: Plana inicial de benvinguda

Per obtenir aquest resultat no ha estat necessari un codi gaire complex (imatge 19), ja que aquesta plana no es caracteritza per tenir un gran nombre de funcionalitats.

```
<div class="flex-center position-ref full-height">
  <div class="top-right sublink">
    @if (Auth::check())
      <a class="" href="{{ route('logout') }}" onclick="event.preventDefault();
        document.getElementById('logout-form').submit();">
        Salir
      </a>

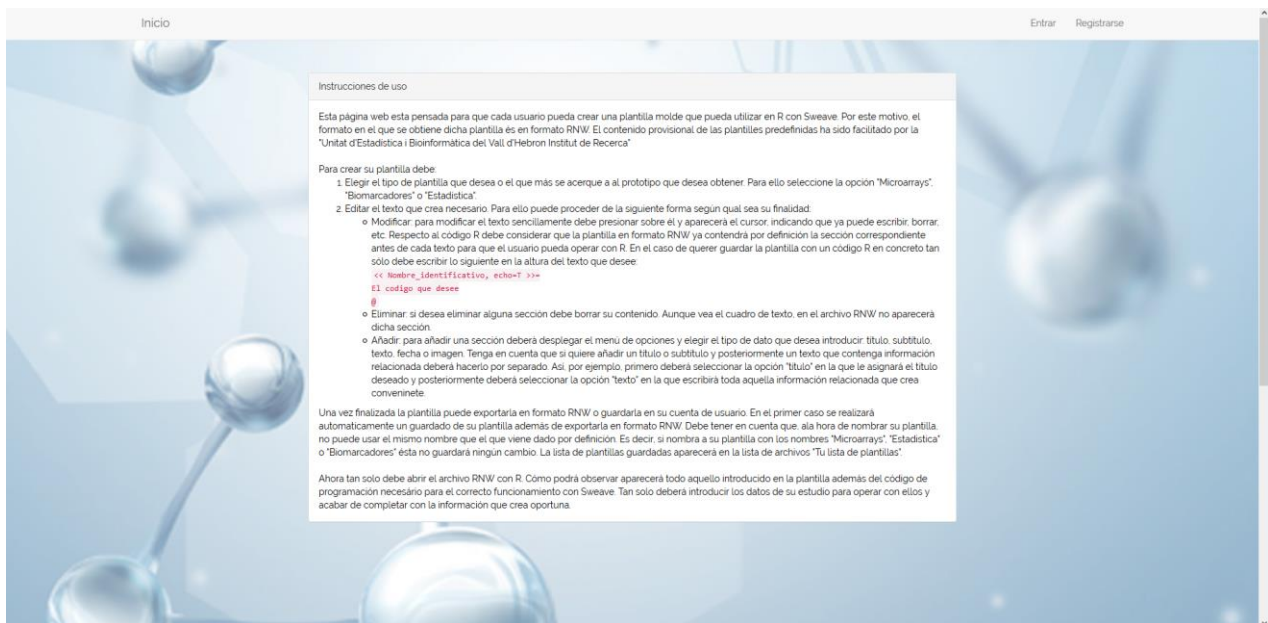
      <form id="logout-form" action="{{ route('logout') }}" method="POST" style="display: none;
        {{ csrf_field() }}
      </form>
    @else
      <a href="{{ url('/login') }}">Entrar</a>
      <a href="{{ url('/register') }}">Registrarse</a>
    @endif
  </div>
  <div class="content">
    <div class="title m-b-md">
      Crea tu plantilla
    </div>
    <div class="sublink top-left">
      <a href="{{ url('/instrucciones') }}">Cómo funciona?</a>
    </div>
    <div class="links">
      <a href="{{ route('plantilla', ['tipo' => 'microarrays']) }}">Microarrays</a>
      <a href="{{ route('plantilla', ['tipo' => 'biomarcadores']) }}">Biomarcadores</a>
      <a href="{{ route('plantilla', ['tipo' => 'estadistica']) }}">Estadística</a>
    </div>
  </div>
</div>
```

Imatge 19: Fragment més representatiu del codi utilitzat per a la plana inicial de benvinguda

En el fragment de codi que es veu a la Imatge 19 es mostren els diferents enllaços que apareixen a la plana: “Entrar”, “Registrarse”, “Cómo funciona?” i l’accés als tres tipus de plantilles. Aquests poden ser enllaços en els que s’especifica directament la url a la que han de redirigir quan es cliqui sobre ells, com és el cas de “Entrar”, “Registrarse” i “Cómo funciona?”, i enllaços en els que s’ha utilitzat la funció “route”, cracterístic de Laravel que redirigeix a arxius de ruta creats previament amb PHP, com és el cas dels tres tipus de plantilla.

El disseny de la web s’ha aconseguit a través dels atributs indicats en cada element de HTML. Per exemple, per al títol “Cómo funciona?” s’han utilitzat les classes “sublink”, creada personalment amb CSS, on s’especifica l’estil, tamany i color de la lletra, i la classe predefinida “top-left”, que situa el text a la part superior esquerra.

Si es clica sobre aquest, s’accedeix a la plana corresponent (imatge 20) on es poden veure unes breus instruccions del funcionament de la pàgina i com crear la plantilla per obtenir l’informe final. Aquesta plana és la més senzilla que l’anterior, ja que no és necessari incorporar cap tipus de funció i només hi apareix text.



Imatge 20: Plana de l’accés a “Cómo funciona?” on apareix una breu descripció de l’ús de la web i la obtenció de la plantilla.

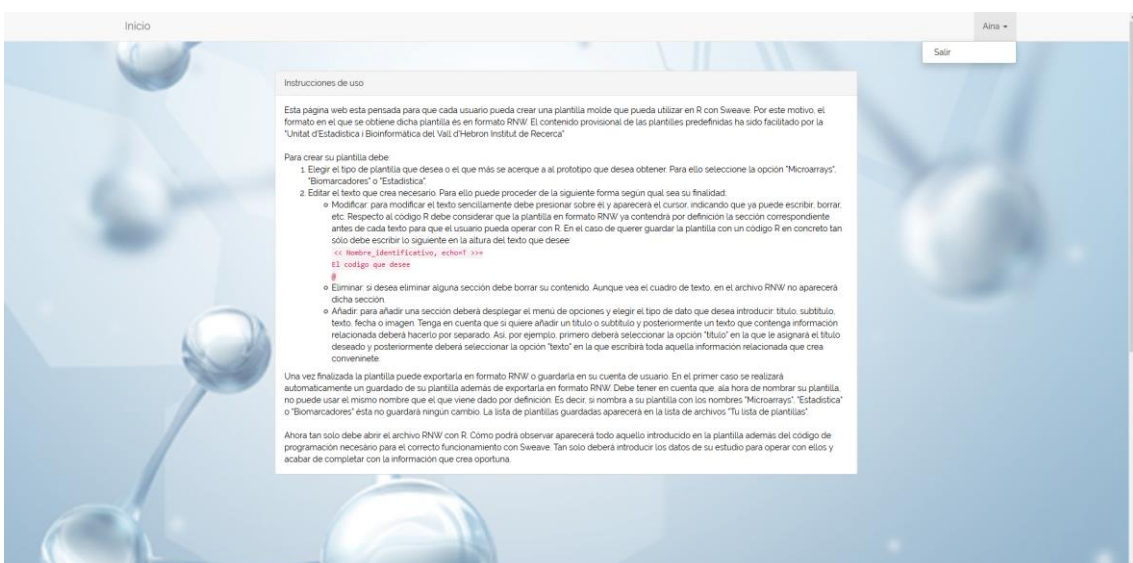
El codi utilitzat en aquest cas només es l'escrit en format HTML amb les respectives característiques de disseny (imatge 21).

```
<div class="container">
  <div class="row">
    <div class="col-md-8 col-md-offset-2 positiontext">
      <div class="panel panel-default">
        <div class="panel-heading estilo">Instrucciones de uso</div>

        <div class="panel-body">
          Esta página web esta pensada para que cada usuario pueda crear una plantilla molde que pueda utilizar en R con Sweave. Por este motivo, el formato en el que se obtiene dicha plantilla es en formato RNW.<br>
          <br>
          Para crear su plantilla debe:
          <ol>
            <li> Elegir el tipo de plantilla que desea o el que más se acerque a al prototipo que desea obtener. Para ello seleccione la opción "Microarrays", "Biomarcadores" o "Estadística".
            <li> Editar el texto que crea necesario. Para ello puede proceder de la siguiente forma según qual sea su finalidad:
              <ul>
                <li>Modificar: para modificar el texto sencillamente debe presionar sobre él y aparecerá el cursor, indicando que ya puede escribir, borrar, etc. Respecto al código R debe considerar que la plantilla en formato RNW ya contendrá por definición la sección correspondiente antes de cada texto para que el usuario pueda operar con R. En el caso de querer guardar la plantilla con un código R en concreto tan sólo debe escribir lo siguiente en la altura del texto que desee:<br>
                  <code>
                    &lt;&lt; Nombre_identificativo, echo=T &gt;&gt;= <br>
                    El codigo que desee<br>
                    @
                  </code>
                </li>
              </ul>
            </li>
          </ol>
        </div>
      </div>
    </div>
  </div>
</div>
```

Imatge 21: Fragment de codi representatiu de la plana "Cómo funciona?"

Cal destacar que la barra superior s'ha programat en un directori diferent per a que afecti a totes les pàgines per igual a excepció de la plana de benvinguda. A més, es va haver d'afegir la classe "fixed", ja predefinida, per mantenir-la sempre al mateix lloc independentment del moviment de la pàgina. També, es va condicionar per a que, en cas de no haver iniciat sessió, a l'extrem dret d'aquesta es mostri l'opció d'accedir al compte o d'enregistrar-se, mentres que, amb la sessió iniciada es motra el nom de l'usuari en forma de menú desplegable (imatge 22) que permet finalitzar la sessió en qualsevol moment.



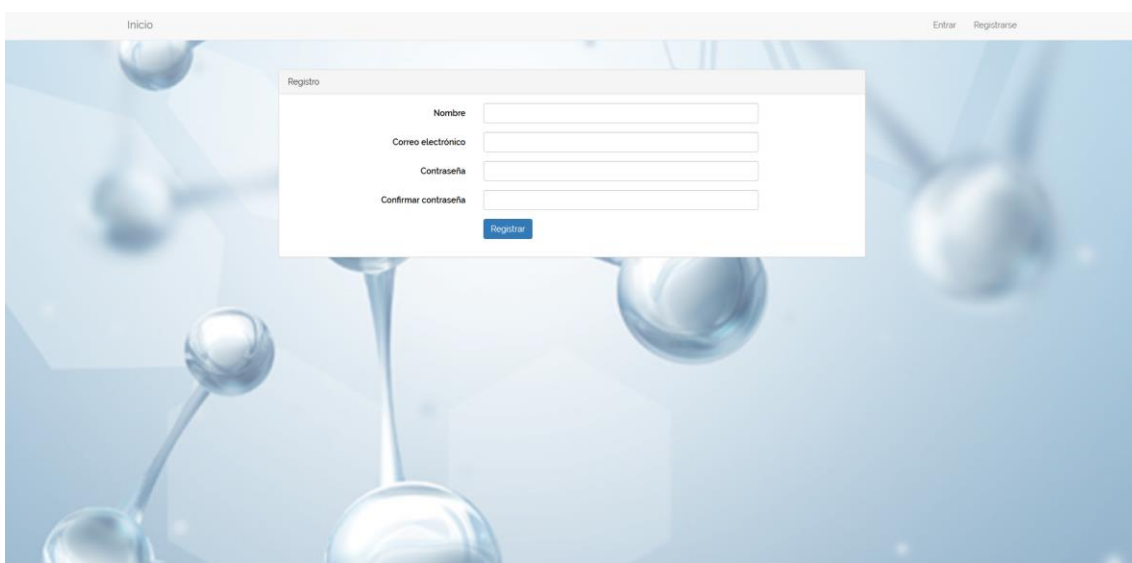
Imatge 22: Plana de "Cómo funciona?" amb la sessió d'usuari iniciada.

De la mateixa manera, un cop iniciada la sessió, a l'extrem superior dret de la pàgina de benvinguda tampoc apareixen les opcions d'entrar o enregistrar-se, sinó que apareix l'opció de "Salir" (imatge 23). Aquesta opció i la barra superior es troben predefinides amb Laravel.



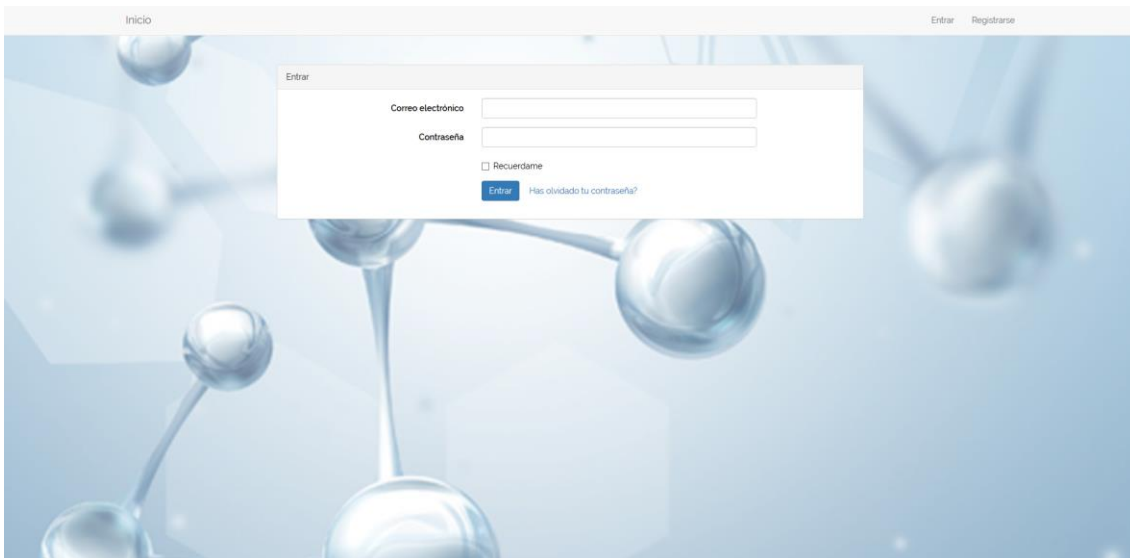
Imatge 23: Plana de la pàgina de benvinguda amb la sessió d'usuari iniciada

De nou, a la plana inicial, trobam els dos links corresponents al registre de l'usuari i a l'inici de la sessió. Així, fent clic sobre "Registrar-se" s'accedeix a la plana de registre on apareix un formulari per introduir totes les dades necessàries (imatge 24) i, per aquells usuaris ja registrats, fent clic a "Entrar" s'accedeix a la plana d'inici de sessió on apareix el formulari corresponent (Imatge 25).



Imatge 24: Plana de l'accés a "Registrar-se" amb el formulari per emplenar de l'usuari.





Imatge 25: Plana de l'accés a "Entrar" amb el formulari a emplenar per l'usuari

En ambdós casos es tracta, de nou, de funcions predefinides per Laravel. Per tant, tan sols s'ha hagut d'introduir el text desitjat a l'espai dedicat per a l'enunciat del formulari. El que sí es va afegir va ser la condició que, en cas no estar enregistrat, al intentar accedir a una de les plantilles, la web redirigís directament a la plana d'inici de sessió. Aquesta funció es va poder realitzar amb codi PHP (imatge 26) de manera que només s'accepten les rutes descrites sota `Auth::routes` si l'usuari és identificat correctament.

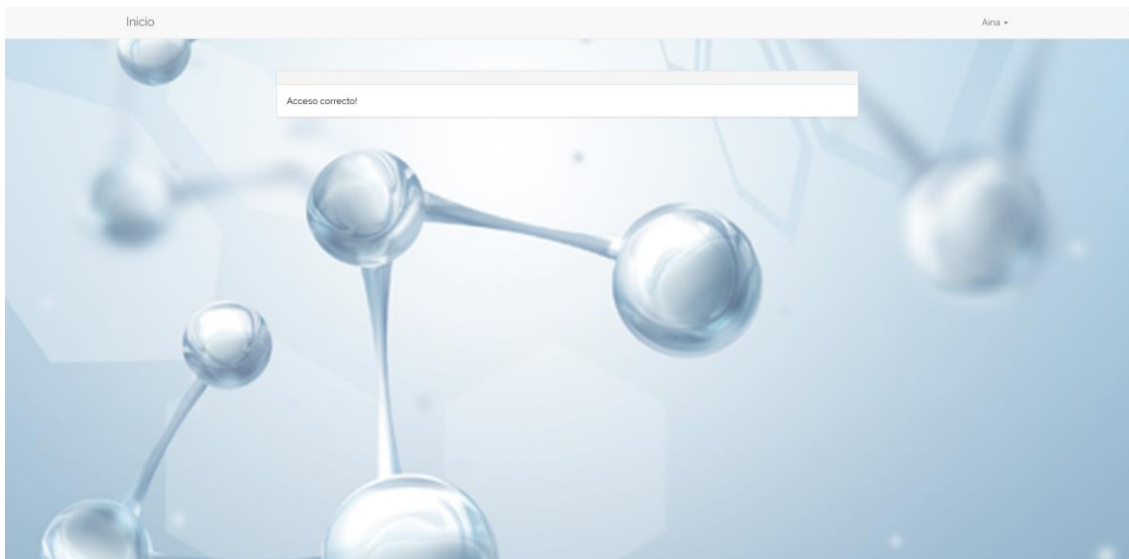
```
Route::get('/', function () {
    return view('welcome');
})->name('portada');

Route::get('/instrucciones',function () {
    return view('instrucciones');
})->name('instrucciones');
Auth::routes();

Route::get('/home', 'HomeController@index');
Route::get('plantilla/{tipo}', 'HomeController@plantilla')->name('plantilla');
Route::post('formPost', 'HomeController@formPost')->name('formPost');
Route::post('formExport', 'HomeController@formExport')->name('formExport');
```

Imatge 26: Llenguatge PHP per determinar les rutes d'accés per usuaris identificats.

En el cas que, l'usuari hagi completat el formulari d'accés al seu compte, la web redirigeix automàticament a la pàgina *home* (imatge 27), també predefinida per Laravel, on s'informa que l'accés s'ha realitzat amb èxit. Per tornar a la plana inicial només cal clicar a "Inicio".



Imatge 27: Plana del directori home un cop iniciada la sessió.

Continuant amb la pàgina inicial, només queda comentar l'accés a les plantilles i com s'organitzen. Per fer-ho només es posarà d'exemple la plantilla Microarrays, ja que totes funcionen de la mateixa manera.

Una característica dels links per accedir a les plantilles és que, al posicionar el cursor sobre ells es marquen en color obscur (imatge 28).



Imatge 28: Plana inicial amb el botó de microarrays marcat.

Aquest efecte s'ha aconseguit afegint una classe pròpia en CSS per als links de les plantilles (imatge 29).

```
.links > a {
  text-decoration: none;
  text-transform: uppercase;
  font-weight: 600;
  font-size: 26px;
  letter-spacing: .1rem;
  color: #0431B4;
  padding: 0 50px;
}

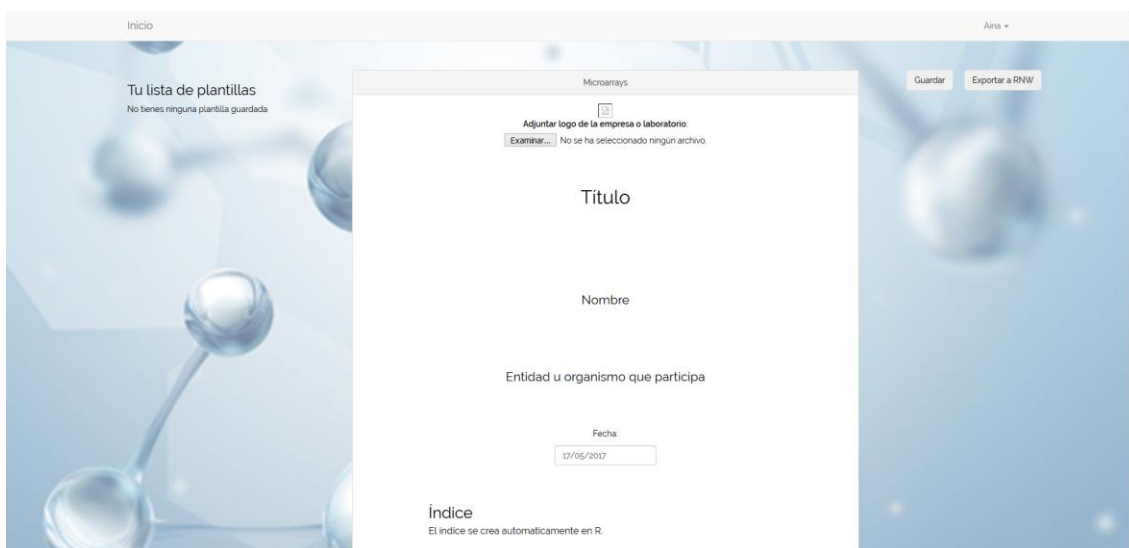
.links a:hover {
  text-decoration: none;
  text-transform: uppercase;
  font-weight: 600;
  font-size: 26px;
  letter-spacing: .1rem;
  padding: 0 50px;
  background-color: #0431B4;
  color: #FFFFFF;
}

.sublink > a {
  text-decoration: none;
  text-transform: uppercase;
  font-weight: 600;
}
```

Imatge 29: Codi CSS aplicat als links de les diferents plantilles.

Com es pot veure en el codi, s'han determinat una sèrie de característiques per a cada classe, en aquest cas les classes "links". La primera és la que caracteritza el diseny dels links en estat neutre, mentres que la segona especifica les seves característiques quan són seleccionats. Al incloure aquesta classe únicament a l'element <div> que determina aquests enllaços, s'evita que les característiques es comparteixin amb la resta de text que trobem a la pàgina. De fet, per a la resta de links se'ls ha otorgat la classe "sublink" que, tot i que a l'imatge no es ve al complet, varia una mica, per exemple, en el tamany de la lletra.

Un cop dins la plantilla seleccionada, en aquest cas la corresponent a Microarrays (imatge 30), es poden dur a terme diferents funcions. En primer lloc, una de les característiques d'aquesta plana és la llista de plantilles guardades per l'usuari que apareix a la banda esquerra.



Imatge 30: Plana web de la plantilla Microarrays.

A la imatge 31 es pot veure el codi utilitzat. En aquest cas, trobam un combinació de codi HTML i PHP. És amb el segon, amb els que s’ha pogut fer referència a la les plantilles guardades per l’usuari en la base de dades mitjançant el conjunt dels símbols “{!! ... !!}”. Així, al seu interior s’ha especificat que, per a la funció “listaPlantillas”, creada en un altre directori, apareguin els noms de les plantilles que s’hi emmagatzemen. Aquesta acció és durà a terme en el cas que la llista tingui continguts, el si no és així es mostra missatge “No tienes ninguna plantilla guardada”.

```
<div class="sidebar-module sidebar-module-inset fixedposition3" style="word-break: break-all;" ;>
<h3>Tu lista de plantillas</h3>
  @if (count ($listaPlantillas)==0)
    <p> No tienes ninguna plantilla guardada </p>
  @else
    <ol class="list-unstyled">
      @foreach ($listaPlantillas as $listaPlantilla)
        <li><a href="{!!route('plantilla', ['tipo' =>
          $listaPlantilla->nombre])!!}">{!!$listaPlantilla->nombre!!}</a></li>
      @endforeach
    </ol>
  @endif
</div>
```

Imatge 31: Fragment de codi per crear una llista de plantilles guardades per l'usuari.

En segon lloc trobem la plantilla en sí. Aquesta mostra els valors guardats a la base de dades que s’ha descrit a l’inici d’aquest punt. Així, com es pot veure en el codi (imatge 32), per a cada camp (títol, subtítol, etc.) es mostren els valors corresponents. Un cop més, aquesta acció s’ha dut a terme amb PHP. Cal recordar que, al utilitzar la funció “route” de Laravel per redireccionar l’enllaç que condueix a la plantilla seleccionada, s’especifica al *framework* de forma automàtica que utilitzi la informació de la base d’aquesta i no d’una altra plantilla. D’aquesta manera es possible utilitzar el codi de forma unificada per a totes les plantilles. A més, com que es diferencia entre els diferents camps, és possible generar una classe amb CSS per a cada un d’ells, de tal manera que el resultat obtingut, independentment de la plantilla que es seleccioni, és un format de plantilla molt similar al que s’obté un cop es compila l’arxiu a pdf des de R amb Sweave.

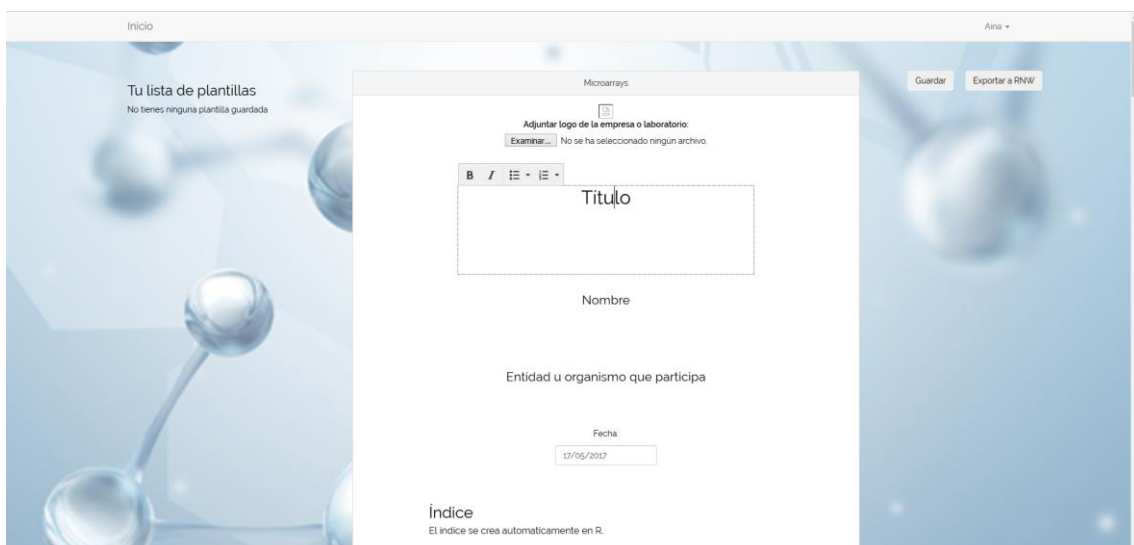
```

<div class="blog-post">
  <div class="blog-post-title">
    Índice
  </div>
  <div class="blog-post-text">
    El índice se crea automáticamente en R.
  </div>
  @for ($i=4; $i < count($valores) ; $i++)
    @if ($valores[$i]->clave->tipo_campo=='titulo')
      <div class="editable blog-post-title" type="text"
        id="{!!$valores[$i]->id.'_'.$valores[$i]->clave->tipo_campo!!}">{!!$valores[$i]->varchar!!}
      </div>
    @elseif ($valores[$i]->clave->tipo_campo=='subtitulo')
      <div class="editable blog-post-subtitle" type="text"
        id="{!!$valores[$i]->id.'_'.$valores[$i]->clave->tipo_campo!!}">{!!$valores[$i]->varchar!!}
      </div>
    @elseif ($valores[$i]->clave->tipo_campo=='texto')
      <div class="editable blog-post-text"
        id="{!!$valores[$i]->id.'_'.$valores[$i]->clave->tipo_campo!!}">{!!$valores[$i]->text!!}
      </div>
    @elseif ($valores[$i]->clave->tipo_campo=='imagen')
      <div>
        <input type="text" name="{!!$valores[$i]->id.'_'.$valores[$i]->clave->tipo_campo!!}"
          value="{!!$valores[$i]->varchar!!}">
      </div>
    @elseif ($valores[$i]->clave->tipo_campo=='fecha')
      <div class="centerdate">
        <p> Fecha:</p>
        <input data-provide="datepicker" class="datepicker form-control" id="data" name="data"
          type="date"
          name="{!!$valores[$i]->id.'_'.$valores[$i]->clave->tipo_campo!!}"
          value="{!!$valores[$i]->date!!}">
      </div>
    @endif
  @endif

```

Imatge 32: Fragment del codi utilitzat per mostrar el contingut de la plantilla seleccionada.

Una de les condicions que havia de complir aquest projecte era la modificació del contingut de la plantilla per part de l'usuari. Aquesta funcionalitat s'ha dut a terme a través de l'editor de text TinyMce. Amb aquest es possible seleccionar el text a modificar (imatge 33) per eliminar o incorporar aquells elements que es creguin necessaris.



Imatge 33: Modificació del text amb TinyMce

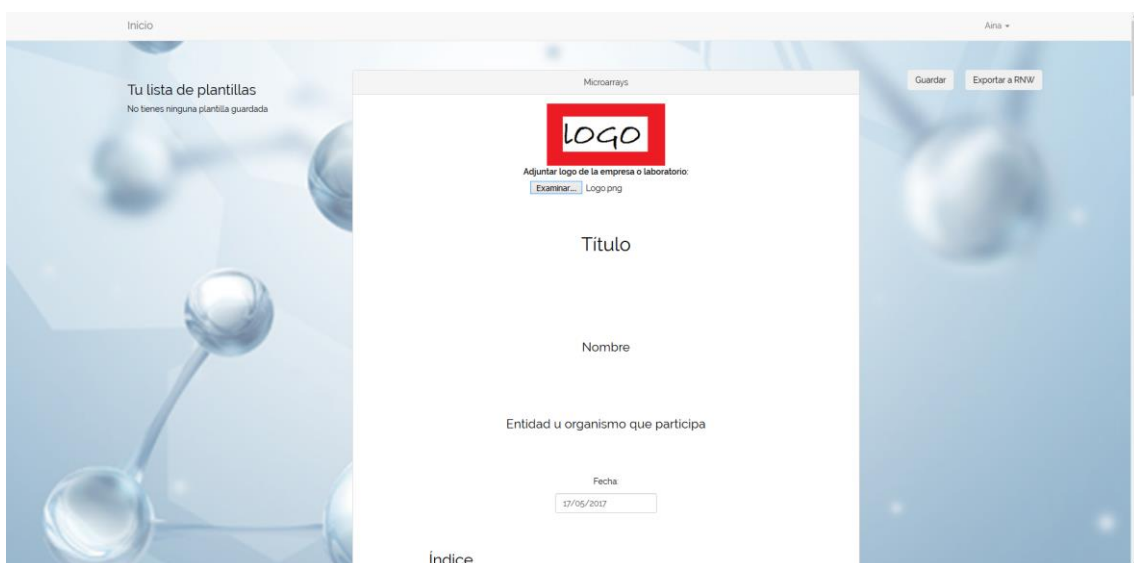
Per al seu ús, però, ha estat necessari utilitzar JavaScript (imatge 34). Aquest llenguatge s'ha utilitzat de forma molt puntual en el projecte per poder incorporar el comportament dinàmic de la pàgina, com l'aparició d'imatges, l'aplicació d'un calendari, etc.

```
tinymce.init({
  selector: 'div.editable',
  inline: true,
  menubar: false,
  toolbar: ' bold italic | bullist numlist',
  plugins: ['advlist autolink lists link image charmap print preview anchor'],
});
```

Imatge 34: Codi JavaScript per incorporar la funcionalitat de TinyMce.

Així, tan sols ha estat necessari especificar el selector, en aquest cas els elements <div> amb la classe “editable” i les característiques desitjades. Aquestes últimes consisteixen en eliminar la barra de menú i donar l'opció de crear enumeracions o editar el format de lletra a negreta o cursiva.

Retornant a la plantilla que es mostra, també s'han incorporat diferents funcions que s'apliquen a l'arxiu RNW que s'exporta. Així, s'ha introduït, com a millora del projecte la possibilitat d'afegir imatges. Aquesta opció és la primera en aparèixer en la plantilla per a que l'usuari pugui adjuntar, per exemple, la imatge de l'empresa (imatge 35).



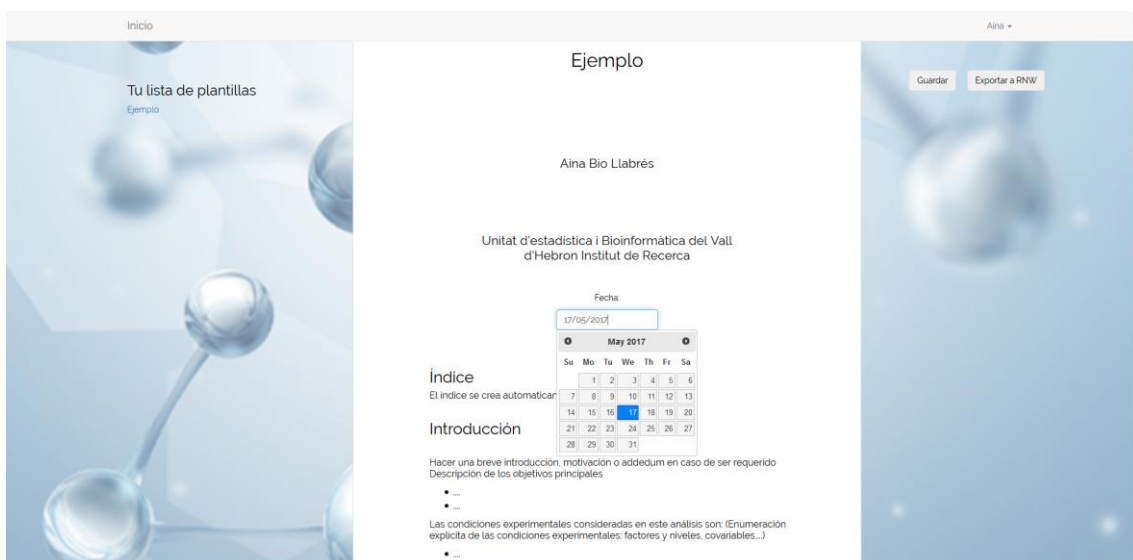
Imatge 35: Mostra d'un arxiu adjunt en representació del logo de la empresa

Per tal de que es puguin visualitzar i guardar amb èxit els arxius pujats per l'usuari ha estat necessari incorporar un element del tipus input amb identificació "archivo1" al llenguatge HTML. Amb aquest s'ha afegit la opció "examinar" mitjançant el codi comunament establert per la inserció d'imatges amb identitat "fotologo". Posteriorment, amb JavaScript, es fa referència a l'input esmentat per donar-li les funcionalitats requerides (imatge 36).

```
function readURL(input) {
    if (input.files && input.files[0]) {
        var reader = new FileReader();
        reader.onload = function (e) {
            $('#fotologo').attr('src', e.target.result);
        }
        reader.readAsDataURL(input.files[0]);
    }
    return input.id;
}
$("#archivo1").change(function () {
    inputId = readURL(this);
});
```

Imatge 36: Codi JavaScript per a la inserció d'imatges i la seva visualització

Per altra banda, com s'ha avançat anteriorment, també s'ha introduït un calendari per seleccionar la data de l'informe (imatge 37). Aquest es troba ocult a no ser que es vulgui modificar la data. Si es així s'obre un calendari tàctil per a la selecció. De nou, per afegir aquesta funció també ha estat necessari utilitzar JavaScript (imatge 38).



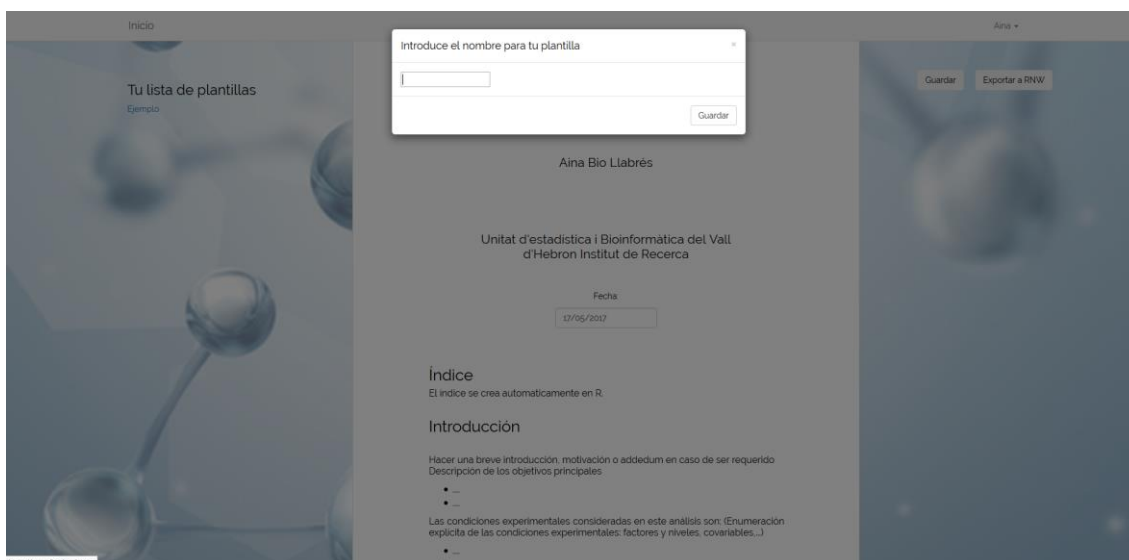
Imatge 37: Calendari per seleccionar la data de creació de l'informe

```
$(document).ready(function(){
    $('.datepicker').datepicker({
        dateFormat: 'dd/mm/yy',
        todayBtn: "linked",
        language: "es"
    });
});
```

Imatge 38: Codi JavaScript utilitzat per la funcionalitat del calendari

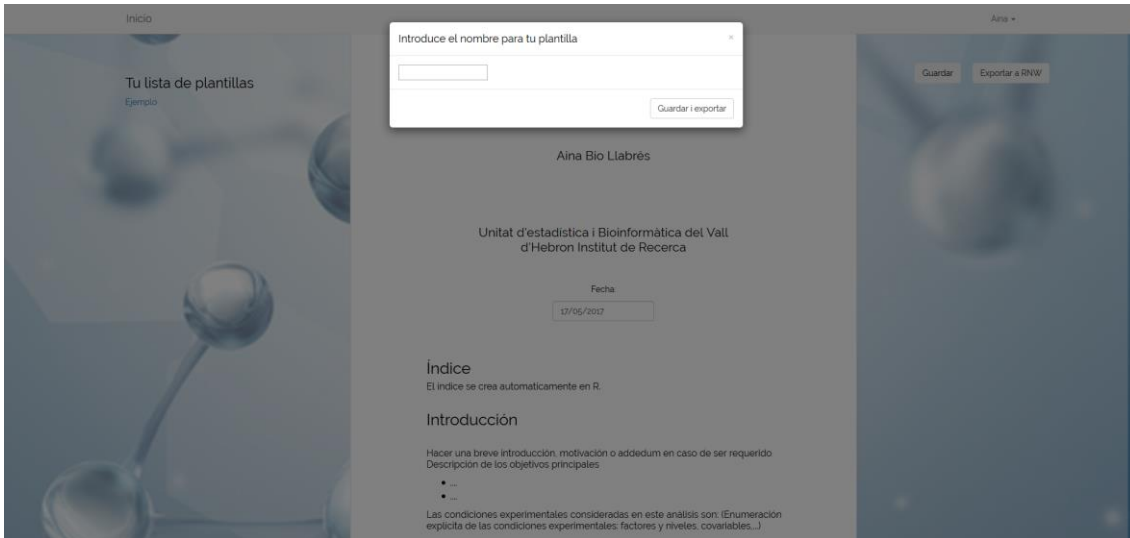
Com es pot veure en la imatge anterior, el codi utilitzat per la funcionalitat del calendari és molt simple. Aquest fet es degut a la utilització de la funció estàndard ja definida “datepicker”, coneguda pels programadors. D’aquesta manera, a la secció “dateFormat”, tan sols és necessari especificar el format de la data que es mostra, en aquest cas dd/mm/aaaa.

Un cop modificada la plantilla es pot optar per guardar-la o exportar-la en format RNW. Tant en la primera com en la segona opció (imatge 39 i imatge 40 respectivament) s’obre una finestra amb el formulari per a que es pugui anomenar la plantilla.



Imatge 39: Plana de la opció de guardat





Imatge 40: Plana de la opció d'exportació en format RNW

La diferència entre ambdues opcions resideix en el botó inferior on, en la opció “Guardar”, apareix el respectiu botó de guardat i, en la opció “Exportar a RNW”, apareix el respectiu botó d’exportació. En aquest últim cal destacar que, a més d’exportar-se també es realitza un guardat de la plantilla modificada.

Ambdues funcionalitats, de nou, han estat implementades per mitjà de JavaScript (imatge 41), fent referència als respectius elements de HTML on es creen els botons de guardat i exportació. Així, com es pot veure, en ambdós casos s’acudeix a la funció “formulario”, utilitzat per al guardat, mentre que el botó que permet, a més, l’exportació, també recorre a l’opció de “exporthidden”. El codi restant s’ha utilitzat per a la funcionalitat del propi botó.

```

$('#guardar').click(function(e){
    $('#formulario').submit();
});

$('#guardarexportar').click(function(e){
    $('#exporthidden').prop('disabled', false);
    $('#formulario').submit();
});

$('.save').click(function(){
    $('#exporthidden').prop('disabled', true);
    var button = $(this).attr('data-button');
    $('.modalSave').addClass('invisible');
    $(button).toggleClass('invisible');
});

```

Imatge 41: Codi JavaScript per a les funcionalitats de guardat i exportació.

Per a que el codi escrit en JavaScript sigui funcional cal conèixer el directori de controladors. En la imatge 42 es mostra un fragment del codi utilitzat per aplicar les diferents funcionalitats de la plantilla. A la part superior de la anterior imatge s’hi troba el codi d’exportació. En aquest s’ordena que es creï l’arxiu a partir de la funció “cadena”, que el guardi al directori “files” i que es mostri la finestra de descàrrega. A sota d’aquest fragment apareix la creació de la plantilla pròpiament dita a partir de la “cadena” nombrada anteriorment. Com es pot veure en aquesta, es crea el text amb tot el codi necessari per a que sigui funcional amb R, incloent els paquets requerits.

```

if (!empty($request->export) && $request->export == '1') {
    $cadena = $this->formExport($plantilla);
    $pathToFile = public_path('files/'.$plantilla->nombre.'.rnw');
    file_put_contents($pathToFile, $cadena);
    return response()->download($pathToFile);
}
else{
    return redirect()->route('plantilla', ['tipo'=> $plantilla->nombre]);
}
}

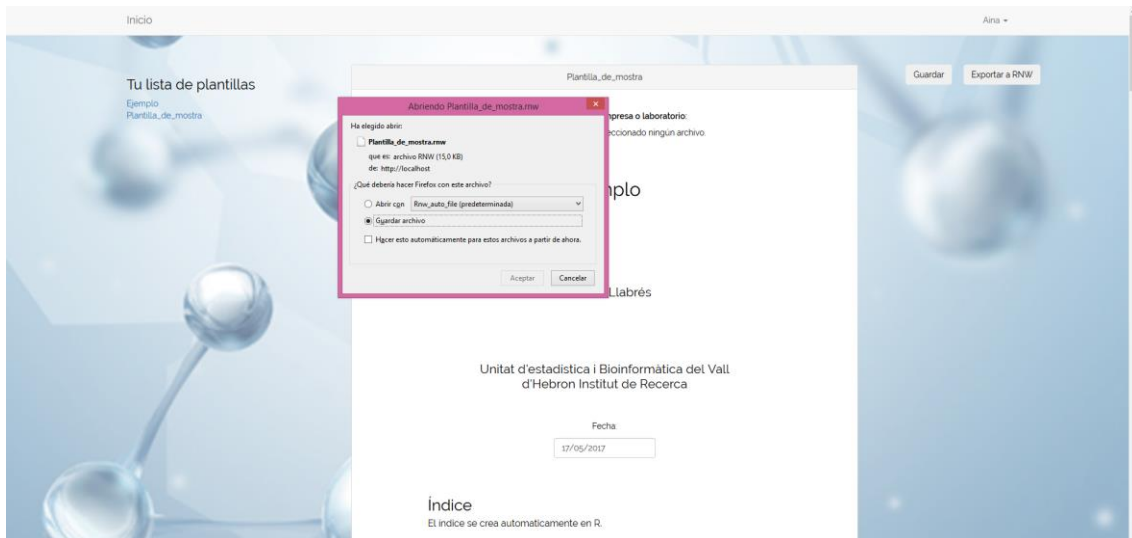
public function formExport($plantilla)
{
    $valores = $plantilla->valores()->with('clave')->orderBy('orden')->get();
    $rutaFile = asset('files/'.$plantilla->logo);
    $cadena
    ="\documentclass{article}\n\usepackage{underscore}\n\usepackage{longtable}\n\usepackage[utf8]{inputenc}\n\usepackage[margin=1in]{geometry}\n\usepackage[spanish]{babel}\n\usepackage{hyperref}\n\usepackage{graphicx}\n\usepackage[table,xcdraw]{xcolor}\n\usepackage{fancyvr}\n\usepackage[lscape]\n\usepackage{anyfontsize}\n\marginwidth{3.0cm}\n\marginheight{3.5cm}\n\marginwidth{2.5cm}\n\marginheight{2.5cm}\n\renewcommand{\baselinestretch}{1.3}\n\interlineadwidth{0.5cm}\n\begin{document}\n\title{\begin{figure}[htbp]\n\centering\n\includegraphics[width=35mm]{\".$rutaFile.\"}\n\end{figure}\n\vspace{1cm}\n\".$valores[0]->text.\"}\n\author{\".$valores[1]->text.\"}\n\date{\".$valores[2]->text.\"}\n\tableofcontents\n\newpage\n\tableofcontents\n\newpage\n";
    for ($i=4; $i < count($valores) ; $i++) {
        if (!empty ($valores[$i])) {
            if ($valores[$i]->clave->tipo_campo=='titulo') {
                $cadena.="\\section{\".$valores[$i]->varchar.\"}";
            }
        }
    }
}

```

Imatge 42: Fragment de codi PHP representatiu del Controlador.

Cal destacar que, tot i que en la imatge anterior no hi surt representat, es realitza una conversió de les dades HTML en format RNW, mitjançant tota una sèrie de procediments que transformen les cadenes de text amb el format introduït per l’usuari (negreta, cursiva, llistats, etc.) en l’equivalent d’aquests formats per al llenguatge Sweave. Així, els caràcters <br> que indiquen un espai en format HTML, són substituïts per “\”, els homòlegs en R.

Finalment, al seleccionar l’opció “Guardar i exportar” s’obre, automàticament, la finestra habitual de descarrega d’arxius on es pot seleccionar “Abrir” o “Guardar como” (imatge 43).



Imatge 43: Finestra de descàrrega de l'arxiu RNW.

Si es presta atenció a la imatge anterior, es pot veure a “Tu lista de plantillas” una llista de plantilles creades per l’usuari. En aquest cas n’apareixen dues com exemple per visualitzar el format de la llista.

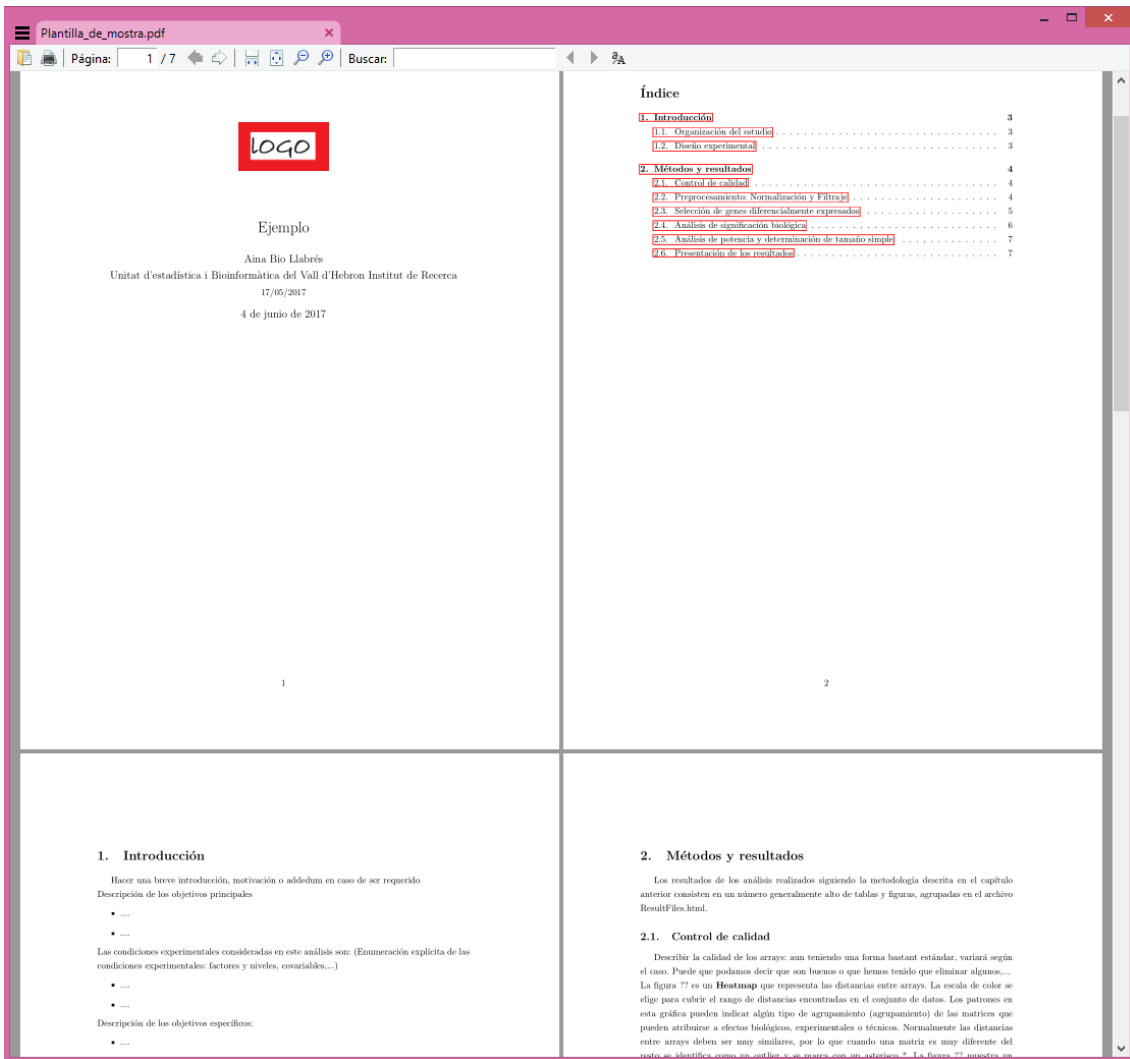
Al obrir l’arxiu RNW descarregat amb R (imatge 44), s’obtenen tots els codis necessaris per a l’elaboració d’un document compilable amb Sweave. Així, primer es presenten una llista de paquets que poden resultar necessaris. Cal esmentar que aquesta llista es una mostra en mode exemple i, en cas que l’usuari en necessiti alguns altres sempre són modificables, a excepció del paquet UTF8.

A més, també apareix el codi característic de Sweave per a la inserció de càlculs o gràfics amb R, així com el corresponent al sistema de numeració i disseny de la tipografia (negreta, espais en blanc, etc.). Respecte a la imatge, seria necessari un servidor per a que R pogués accedir al directori i carregar-la. Al no existir, de moment, s’ha canviat de forma manual la ruta predeterminada per la ruta local de la imatge.

```
5 \usepackage{margin=1in}{geometry}
6 \usepackage{spanish}{babel}
7 \usepackage{hyperref}
8 \usepackage{graphicx}
9 \usepackage{table,xcdraw}{xcolor}
10 \usepackage{fancyvrb}
11 \usepackage{lscap}
12 \usepackage{anysize}
13 \marginwidth{3.0cm}{3.5cm}{2.5cm}{2.5cm}
14 \renewcommand{\baselinestretch}{1.3} %interlineado
15 \begin{document}
16 \title{\begin{figure}[htbp]
17 \centering
18 \includegraphics[width=35mm]{C:/Users/Aina/Desktop/Logo}
19 \end{figure}
20 \vspace{1cm}
21 Ejemplo}
22 \author{Aina Bio LabrÀes \
23 Unitat d'estadística i BioinformÀtica del Vall d'Hebron Institut de Recerca\
24 {\normalsize 17/05/2017} }
25 \vspace{1cm}
26 \maketitle
27 \newpage
28 \tableofcontents
29 \newpage
30
31 \section[IntroducciÀn]
32 <<nombreseccion,echo=F,fig=T, warning=F, message=F>>=
33 @
34 Hacer una breve introducciÀn, motivaciÀn o addendum en caso de ser requerido
35 \DescripciÀn de los objetivos principales
36 \begin{itemize}
37 \item ....
38 \item ....
39 \end{itemize}
40 Las condiciones experimentales consideradas en este anÀlisis son: (EnumeraciÀn explÀcita de las condiciones experimentales: factores y niveles, covariables,...)
41 \begin{itemize}
42 \item ....
43 \item ....
44 \end{itemize}
45 DescripciÀn de los objetivos especÀficos:
46 \begin{itemize}
47 \item ....
48 \item ....
49 \end{itemize}
50 DescripciÀn de observaciones especiales como los efectos batch o el emparejamiento de muestras
51 \subsection[organizaciÀn del estudio]
52 <<nombreseccion,echo=F,fig=T, warning=F, message=F>>=
53 @
54 El anÀlisis se ha realizado siguiendo el habitual "pipeline" para el anÀlisis de datos de microarrays adaptado a este experimento en particular. A grandes rasgos, cada uno de los puntos siguientes corresponde a una secciÀn del informe de anÀlisis.
55 \begin{enumerate}
56 \item Control de calidad: ¿Tenemos buenos datos? ¿DeberÀa rechazarse alguna diapositiva?
57 \item Preproceso de datos: resumen, filtrado y normalizaciÀn.
58 \item SelecciÀn de genes expresados diferencialmente para cada conjunto de condiciones.
59 \item BÀsqueda de patrones de expresiÀn gÈnica y agrupaciÀn de muestras.
60 \end{enumerate}
61 \end{document}
```

Imatge 44: Arxiu RNW obert amb R

Un cop inserits tots els camps i realitzades les modificacions oportunes es pot compilar a PDF. El resultat es un informe (imatge 45) amb una estructura molt similar a la que es podia visualitzar en la pàgina web. Aquest conté tant la imatge seleccionada com el text predefinit. A més, a partir de la funció que facilita R, tots els títols i subtítols han estat numerats i incorporat en un índex dinàmic que permet accedir a l'apartat d'interès clicant sobre ell.



Imatge 45: Informe resultant de la compilació de l'arxiu RNW amb R i Sweave