

Desenvolupament d'una web-app per fomentar l'aprenentatge de biologia

Cristian Vega Sánchez

Resum– Aquest projecte ha consistit a crear una web-app per a la facultat de biologia animal de la UAB. Aquesta aplicació té com a objectiu poder divulgar i fomentar coneixements biològics animals així com explicar les diferents tècniques genòmiques, buscar animals enregistrats pels alumnes/professorat, consultar informació paleogènica (amb les seves corresponents tècniques), Projectes i investigació actuals. Disposa també d'un glossari de les diferents paraules, preguntes freqüents i un mòdul per poder realitzar *quiz* autoavaluatius de coneixements. Tot això dins d'un projecte real a mida sota demanda d'un client (professorat biologia), passant per totes les fases del desenvolupament software així com els requeriments, extracció, implementació, acceptació i validació continuada. S'ha creat un servidor web amb *Back end* format per *NodeJS* amb *Express (Api RestFul)* i el *Express (Front end)* creat en *ReactJS* amb l'ajuda del *framework Material-UI* amb una base de dades *MySql/MariaDB*.

Paraules clau– Web-App, Biologia Animal, Aprenentatge, Divulgació, Desenvolupament Software a mesura, ReactJS, NodeJS, Express,Material-UI,Api RestFul, MySql.

Abstract– This project consisted in creating a web-app for the faculty of animal biology at the UAB. This application aims to disseminate and promote biological knowledge of animals and explain the different genomic techniques, search for animals registered by students / teachers, consult paleogenomic information (with their corresponding techniques), current projects and research. Glossary of the different words, FAQs and a module to perform self-assessment quizzes of knowledge. All this within a real project tailored to the demand of a client (Biology teachers), going through all phases of software development as well as requirements, extraction, implementation, acceptance, continuous validation. A web server has been created with Back end formed by *NodeJS* with *Express (Api RESTful)* and the Front end created in *ReactJS* with the help of the framework *Material-UI* with a database *MySql / MariaDB*.

Keywords– Web-App, Animal Biology, Learning, Dissemination, Custom Software Development, ReactJS, NodeJS, Express,Material-UI,Api RestFul, MySql.

1 INTRODUCCIÓ

VOLIEM realitzar un projecte pedagògic per poder donar visibilitat i fàcil accessibilitat a la biologia animal, en el que els alumnes de 4t de Biologia generessin un seguit de materials sobre genomes d'animals i que estiguessin disponibles per tothom que volgués consultar-los en una web-app. D'aquesta manera l'aplicació funciona tant, per consultar informació d'una manera

estructurada i atractiva com per poder actualitzar el contingut de la pàgina amb les investigacions que els diferents alumnes realitzant durant el grau. Mitjançant aquesta aplicació qualsevol persona podrà consultar d'una manera fàcil i des de qualsevol dispositiu (ordinador /mòbil/tauleta...) tot allò que envolta la biologia animal i els genomes dels animals. Creiem que és una bona opció, ja que augmentarà l'interès de l'alumnat i també farà arribar a tothom les investigacions que es realitzant a la UAB.

A nivell més personal suposa un repte en el qual plasmar tot el coneixement adquirit durant el Grau i l'estada de pràctiques, així com passar per totes les fases del desenvolupament d'un projecte a mesura amb un client real.

- E-mail de contacte: cristian.vegas@e-campus.uab.cat
- Menció realitzada: Enginyeria del Software
- Treball tutoritzat per: Yolanda Benítez Fernández (departament)
- Curs 2019/20

2 ESTAT DE L'ART

L'estat actual en el qual es troba el professorat de 4t de Biologia, és que l'alumnat de l'últim curs realitzant un projecte en el qual els alumnes han de realitzar una investigació sobre els genomes de diferents animals. Aquesta investigació es plasma en un document que s'entrega al professor per la seva correcció i puntuació, que posteriorment el guarda de manera local. S'ha realitzat una recerca d'informació i tecnologies referent a projectes similars al que es vol realitzar, buscar aplicacions semblants per a veure diferents opcions disponibles per oferir al client. Com que es tracta d'un producte dissenyat a mida en aquest apartat ens hem centrat més en l'estat de l'art de les eines que utilitzarem, així com els diferents *Stacks* tecnològics i possibles *frameworks* per la creació de la pàgina web. Quant a la part visual (Interfase de l'aplicació) hem mirat plantilles de dissenys amb una ambientació acadèmica, orientades al *frameworks* que utilitzarem:

- **Material-UI (React):** per a una ràpida i senzilla creació de pàgines web amb resultat d'alta qualitat i un producte formal.

Pel que fa a l'stack tecnològic el que buscàvem era un que ens permetés un desenvolupament d'una API de forma senzilla, i sobretot stacks tecnològics orientats a un sol individu (Full Stack) per un desenvolupament més ràpid i no tan orientat a equips de desenvolupament.

- **MERN:** *MongoDB* (Base de dades), *Express* (Api Rest), *React* (*JavaScript Framework / Front end*), *NodeJS* (*Back end / Servidor*).

** Finalment vam haver de canviar el tipus de bases de dades per adaptar-nos als servidors (*MariaDB/MySQL*).

3 OBJECTIUS

Com a objectius principals del projecte podem trobar:

- **Correcta extracció de requisits:** Ser capaços de recopilar totes aquelles necessitats que el nostre client final vol en el seu producte, sempre tenint en compte les limitacions de temps i coneixements. És molt important una bona comunicació i feedback per les dues parts.
- **Adquisició de coneixement:** Ser capaç d'afrontar totes aquelles mancances de coneixement i enfrontar tots aquells reptes que vaguin apareixent en el desenvolupament del producte.
- **Oberta a tothom:** L'objectiu principal és que aquesta pàgina sigui utilitzada per l'alumnat però també per qualsevol persona que tingui dubtes al respecte sobre els temes relacionats amb els genomes i les diferents recerques actives de la universitat.
- **Integració amb els servidors de la UAB:** Ser capaç de configurar i muntar tot l'entorn de producció la màquina prestada pels serveis SID de la Facultat de Biologia.
- **Aplicació multi llenguatge:** L'aplicació ha de permetre seleccionar un dels següents idiomes: Angles/Català/Castella.

- **Mantenibilitat:** L'aplicació ha de perdurar en el temps i ser actualitzable. Per tal que el professorat pugui portar al dia la pàgina web (actualització del contingut) se li habilitarà una pàgina per poder modificar o afegir el contingut.

- **Disponibilitat:** L'aplicació ha de tenir una alta disponibilitat. Ja que la idea és que es pugui consultar en qualsevol moment (24/7). Aquesta disponibilitat està molt vinculada als servidors de la UAB.

Objectius secundaris:

- **Disseny:** Realitzar un disseny de qualitat per tal de definir clarament l'aplicació i evitar la refactorització del codi al màxim.
- **Bones Pràctiques:** Aprendre les bones pràctiques del llenguatge així com optimització de codi per l'obtenció d'un producte de qualitat.
- **Planificació:** Complir tots el deadlines plantejats a la planificació del projecte i haver fet una bona estimació de les diferents tasques.

4 METODOLOGIA I PLANIFICACIÓ

En aquest apartat definirem les metodologies de treball i software en l'elaboració del projecte:

4.1 Metodologia de treball

Hem decidit apostar per una metodologia de treball de tipus *Kanban*, per així tenir una clara visió i planificació de totes les tasques del projecte. Gràcies a aquest tipus de Metodologia i a les eines *Monday* (*Scrum Board*) podem veure d'una manera fàcil la traçabilitat de les diferents tasques així com en l'estat en què es troben, està bastant orientant per equips de desenvolupadors, però en aquest cas l'utilitzarem de manera individual, ja que incentiva l'ús de bones pràctiques, així com haver de redactar i definir detalladament les tasques abans de començar-les a fer.

També realitzem *Spring Plannings* setmanals per tal de no perdre l'hàbit de treball, redefinició de tasques i afegir-ne de noves, similar a les metodologies Agile - SCRUM.

Les categories utilitzades són: *TODO*, *WIP* (*Work In Progress*), *READY QA* i *DONE*. També hem utilitzat etiquetes per fer-ho més visual (disseny, reunions, servidors, implementació, informe, etcètera).

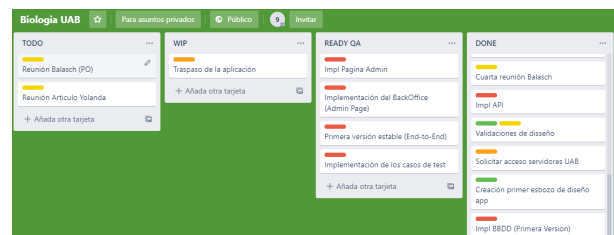


Fig. 1: Imatge del tauler Kanban/Scrum

** Finalment vam traspasar el tauler *Scrum* de *Monday a Trello* (pel funcionament gratuït).

4.2 Metodologia software

Quant a la metodologia software hem preferit passar del disseny directament a la implementació i finalment fer el testing i la revisió de bugs/errors. D'aquesta manera ens centrem en el funcionament principal de l'aplicació i després del refinament, assumint tots els riscos que això implica com l'augment de temps de la refactorització del codi. Hem decidit apostar per aquesta metodologia assumint d'un augment del temps de realització, ja que haurem de refer molt treball (Refactor) però per altra banda consolidarem el coneixement adquirits d'aquestes noves tecnologies (Refinament continu), ja que tenim poc coneixement previ.

4.3 Planificació

Les primeres quatre setmanes del projecte se'n van dedicar íntegrament a la seva planificació i la primera toma de contacte amb el product owner (Profesorat de Biologia), amb la tutora del projecte (Yolanda Benítez) i amb el SID (Serveis d'informàtica distribuïda) de Biologia. Pot semblar massa temps però creiem que és necessari perquè és una fase molt important per establir unes bones bases pel bon desenvolupament del projecte.

El projecte el podem dividir principalment en set etapes o tasques principals:

- **Extracció de requeriments:** Captura els requisits del professorat de la facultat de biologia. Entendrà el problema i les necessitats de l'alumnat. Una forta extracció inicial de la idea del projecte que s'anirà validant amb reunions periòdiques.
- **Disseny:** Un cop definides les funcionalitats/requisits del projecte. Realització de prototips i mockups per tal de tancar amb el Product Owner (Juan Carlos Balasch Alemany) la part visual de l'aplicació.
- **Estudi de les tecnologies:** Així com realitzar cursos en línia de JavaScript i React, practicar amb l'API de Material-UI, creació de projectes prova així com creació de l'ecosistema software (Stack Tecnologic).
- **Implementació:** Un cop recollits tots els requisits i amb el disseny de l'aplicació tancat, passem a la part més tècnica de programació i implementació del codi de l'aplicació. Dintre d'aquesta implementació podrien diferenciar en 3 grans blocs: Base de dades, Front end i Back end.
- **Refactoring i Clean Code:** Un cop amb una versió End-to-End de l'aplicació funcional, passem a netejar el codi: evitar codi repetit (reutilització de components), neteja dels imports utilitzats (adaptar només als necessaris), revisió de nom de variables i funcions, etcètera.
- **Testing:** Realització de proves per verificar un cert nivell d'estabilitat i qualitat del projecte. Test unitari, Proves d'integració, proves visuals (Front End) i Exploratory testing (a partir dels casos de prova).
- **Verificació:** l'objectiu d'aquesta etapa és assegurar que el programari compleixi plenament amb els requisits previstos i acordats amb el client.

| TAREA | ASIGNADO Participantes | INICIO | FIN |
|---|----------------------------------|---------|---------|
| Requeriments | | | |
| Captura de requisits | Cristian i Juan Carles | 15-2-20 | 22-2-20 |
| Definició d'objectius | Cristian i Juan Carles | 22-2-20 | 3-3-20 |
| Definició de la planificació | Cristian i Yolanda | 3-3-20 | 13-3-20 |
| Definició de la metodologia | Cristian | 13-3-20 | 23-3-20 |
| Validació de requisits | Cristian i Juan Carles | 23-3-20 | 2-4-20 |
| Disseny | | | |
| Investigació i avaluació de tecnologies | Cristian | 23-3-20 | 28-3-20 |
| Disseny de l'aplicació (MockUps) | Cristian | 7-4-20 | 12-4-20 |
| Arquitectura de la aplicació (Software) | Cristian | 12-4-20 | 15-4-20 |
| Validació de MockUps | Cristian i Juan Carles | 12-4-20 | 14-4-20 |
| Generació de prototips | Cristian | 12-4-20 | 15-4-20 |
| Definició de l'estructura de la BBDD | Cristian | 15-4-20 | 18-4-20 |
| Implementació | | | |
| Creació de la BBDD | Cristian | 18-4-20 | 23-4-20 |
| Instal·lació de l'entorn (DevOps) | Cristian | 24-4-20 | 28-4-20 |
| Instal·lació de paquets | Cristian | 29-4-20 | 4-5-20 |
| Implementació de l'arquitectura | Cristian | 5-5-20 | 9-5-20 |
| Integració dels mòduls | Cristian | 24-4-20 | 28-4-20 |
| Validació de prototips amb el client | Cristian i Juan Carles | 28-4-20 | 3-5-20 |
| Implementació del testing | Cristian | 29-4-20 | 19-5-20 |
| Verificació | | | |
| Cerca de bugs | Cristian | 19-5-20 | 26-5-20 |
| Validació Final | Cristian , Juan Carles i Yolanda | 26-5-20 | 2-6-20 |
| Traspàs al servidor de l'aplicació | Cristian i SID UAB | 2-6-20 | 16-6-20 |
| Entrega de manuals d'ús i tutorial | Cristian i Juan Carles | 2-6-20 | 16-6-20 |
| Presentació del projecte | Cristian | 8-7-20 | 9-7-20 |

Fig. 2: Planificació de les tasques

5 REQUISITS

Els requisits funcionals principals que ens vam marcar i que volíem complir són:

- L'alumne/usuari pugui accedir al servidor web, des de qualsevol dispositiu (web-app).
- L'aplicació sigui trilingüe.
- L'alumne/usuari pugui consultar el contingut de la pàgina web. Tipus de continguts: textos/posts, PDF, imatges i glossaris.
- L'alumne/usuari pugui realitzar consultes en el buscador d'animals.
- L'alumne/usuari pugui realitzar Quiz autoavaluatius per tal de veure el seu nivell.
- L'administrador pugui modificar el contingut de la pàgina web. Així com els textos, imatges, documents, afegir/modificar animals, crear/eliminar quiz, etc.

I els requisits no funcionals:

- Sistema eficient per tal d'optimitzar els recursos del servidor on s'allotjarà.
- Sistema segur: prevenció contra atacs/bugs i guardar la informació de forma segura.
- Usabilitat: Interfície *Userfriendly* i minimalista amb caràcter educatiu per una millor experiència d'usuari.
- Fàcil mantenibilitat: Orienta l'aplicació al màxim perquè sigui molt fàcil de realitzar el manteniment d'aquesta.
- Que totes les dades emmagatzemades siguin legals amb la Llei orgànica de protecció de dades.

6 DISSENY

En aquesta secció explicarem com hem realitzat el disseny tant de la base de dades com UI/UX i de l'aplicació:

6.1 Disseny Base de dades

Per realitzar el disseny de la base de dades vam pensar en el contingut que cada pàgina hauria de descarregar de la base de dades. El plantejament de l'aplicatiu multilingüe està directament connectat amb la base de dades, ja que aquesta ha d'emmagatzemar el contingut dels literals en els 3 idiomes, i mitjançant l'API retornar el contingut a l'usuari final. N'hi ha molt poc contingut estàtic pràcticament tot és dinàmic a la base de dades perquè així el professorat mitjançant la pàgina d'administració puguin canviar pràcticament tot el contingut de la web i així eliminar el manteniment extern per part d'un informàtic. Això fa augmentar el nombre de trucades a la base de dades, però assumint els riscos.

Primerament la base de dades es va realitzar orientada a documents (JSON), ja que volíem utilitzar dades de bases no relacionals (MongoDB), però un cop vam parlar amb el departament del servidor ens van limitar a utilitzar MariaDB/MySQL. Per això podem observar que tot i ser una base de dades relacional no apareixen relacions entre les taules.

6.2 Disseny UI/UX

La idea principal del disseny visual de l'aplicació ha estat extreta del client, tenia bastant clara la forma que volia que tingués l'aplicatiu. Gràcies al seguiment periòdic amb aquest a poc a poc anàvem concretant cada vegada més el disseny de l'aplicatiu. Per tal de presentar els diferents prototips i mockups ens hem reforçat amb eines ja existents com App-Moqups o Draw.IO, però també hem utilitzat dissenys a paper per les primeres aproximacions.

**Tot el disseny de l'aplicatiu ha estat validat pel client final.

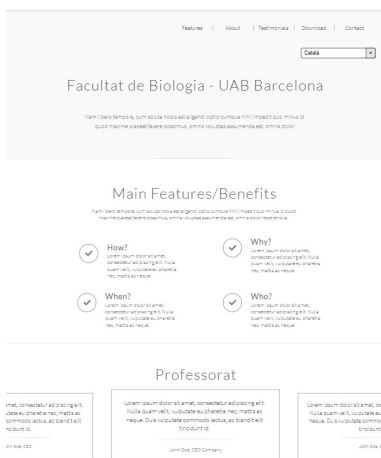


Fig. 3: Primera aproximació amb App Moqups

6.3 Disseny de l'aplicació

A continuació explicarem les diferents tecnologies utilitzades pel desenvolupament de l'aplicació i la seva funcionalitat:

- **Base de dades** : *MariaDB /MySQL*, encarrega d'emmagatzemar tots els textos /literals en els 3 idiomes.
- **Back End** : *NodeJS - Express*. Encarregat de desplegar l'aplicació (publica index.html) i aixecar l'API Rest (Rutes API).
- **Front End** : *HTML- CSS - JS - React - Material-UI - Babel - Webpack*, part visual de l'aplicatiu separada per vistes i components. Tota aquesta part es compila en un *bundle.js* que s'inserta al *index.html* (pàgina web publica).
- **Control de versions**: *Git*, per tenir controlades totes les versions de l'aplicatiu.

Podem observar el diagrama de funcionament de la web-app, així com interaccionen les diferents parts del sistema a un nivell alt d'abstracció a la Figura 4.

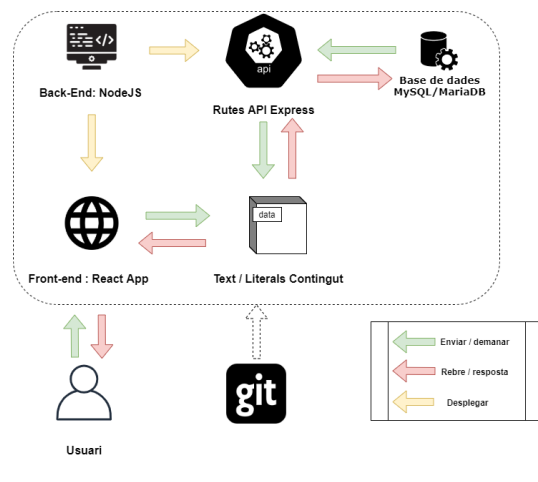


Fig. 4: Diagrama general del funcionament de l'aplicació

7 DESENVOLUPAMENT

7.1 Estructura de carpetes

Diferencien clarament el nostre projecte en dos de diferents, front end (part visual) i el back end (part del servidor), dintre d'aquestes carpetes trobem:

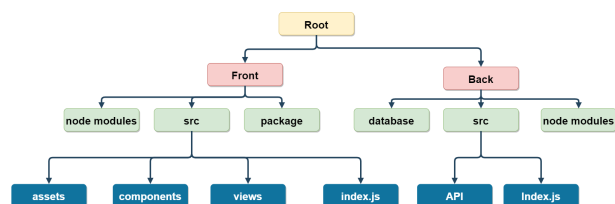


Fig. 5: Estructura de carpetes Appompics

- **Node modules** : tots els mòduls importats al projecte. Exemples: *Material-ui, Express, Webpack*, etcètera.

- **Assets** : Carpeta on guardem totes les imatges i logos de l'aplicació.
- **Views** : Pàgines principals de l'aplicació.(Combinació de components).
- **Components** : Elements individuals amb els seus atributs. Un exemple de component seria un Peu de pàgina, que es reutilitza en diverses pàgines i està format per una imatge y un text.
- **API** : Definició de tots els End-Points (Rutes d'accés). Un exemple de ruta seria: *GET "/language"* que ens retorna el llenguatge actual.
- **Bases de dades** : Conte dos fitxers de creació i inserció de les dades.

7.2 API Rest

L'API ens permet establir la connexió entre l'aplicació i la base de dades. L'aplicació consumeix l'API que aquesta l'hi retorna contingut o bé l'usuari envia dades cap a la base de dades, en el nostre cas l'únic usuari que envia contingut seria el professor a través del seu portal, la resta d'usuari demanen el contingut.

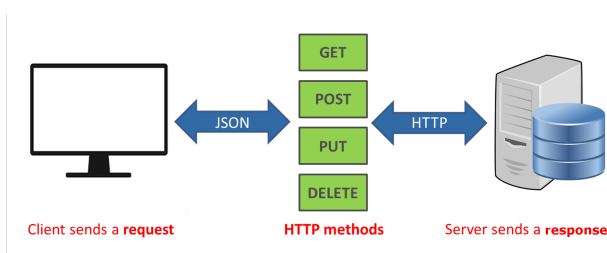


Fig. 6: Funcionament general d'una API

A cada ruta l'assignem un mètode *HTTP* d'aquest llistat, segon la necessitat:

- **GET** : Retorna informació.
- **POST** : Enviar informació al servidor.
- **PUT** : Guarda informació en una localització.
- **PDELETE** : Elimina informació.

| End-Points API Appomics | | |
|-------------------------|------------------|--------------------------------|
| Mètode HTTP | URL | Funció |
| GET | /language | Retorna el llenguatge actual |
| DELETE | /deletePost | Elimina el post seleccionat |
| POST | /insertPaleoTech | Insereix tècnica paleogenètica |

TAULA 1: EXEMPLES DE RUTES DE L'API DE L'APLICACIÓ.

8 RESULTATS

Com a resultat final podem trobar les següents pàgines a la web:

- **Pàgina de destinació (Landing Page)** : Pàgina de benvinguda, enllaça amb la pàgina principal, la pàgina de la UAB i la pàgina d'informació del professorat encarregat de la web (Team).



Fig. 7: Landing Page - APPOMICS

- **Pàgina Principal** : On trobaràs tots els mòduls de la web-app, aquesta pàgina és multi idioma i també enllaça amb la pàgina d'informació en el peu de pàgina.
 - Com: Les tècniques.
 - Per què: Les Espècies.
 - Quan: Paleogenomes.
 - Qui: Els Projectes.
 - Què: Els conceptes.



Fig. 8: Pàgina Principal - APPOMICS

- **Pàgina Com - Les tècniques** : Pàgina on es mostren totes les tècniques genòmiques. Conte un títol, subtítol, imatge, peu de foto i es pot agregar enllaços de referència per a cada post. (multi idioma)
- **Pàgina Per què - Les espècies** : Buscador d'animals, amb tots els animals registrats i la seva corresponent



Fig. 9: Pàgina Animal: Elefant exemple

informació (multi idioma). Conté informació del medi, genoma (característiques i gens d'interès) i fenoma, també es poden inserir imatges i enllaços de referència.

- **Pàgina Quan - Paleogenome** : Pàgina on es mostren tota la informació del paleogenomes. Està dividida en tres apartats (component Tab): Espècies que conte els problemes de l'ADN Fòssil i els genomes fòssils registrats (enllaça amb la pàgina de l'animal), Tècniques que conté totes les tècniques paleogèniques registrades i els Problemes de la desextinció.



Fig. 10: Pàgina Quan - Paleogenome - Desextinció Tab

- **Pàgina Qui - Els Projectes** : Pàgina on podem trobar tota la informació relacionada amb:
 - **Projectes** : Informació dels projectes genòmics i les investigacions de genomes.

- **Investigació** : La perspectiva de gènere en els models genòmics i la perspectiva de gènere en les investigacions.



Fig. 11: Pàgina Qui: Els Projectes

- **Pàgina Què - Els conceptes** : Aquí podrà trobar:
 - **Glossari**: Catàleg alfabeticat de les paraules i expressions.
 - **Preguntes freqüents** : Aquí podrà trobar les preguntes més freqüents sobre tècniques o conceptes i les seves corresponents respostes.
 - **Mòduls de pràctica** : Mòduls d'autoavaluació format per una temàtica i 10 preguntes amb 4 opcions possibles cada una.
- **Pàgina Informació** : Guia de navegació i utilització de la web i informació del professorat encarregat de la pàgina web.
- **Pàgina Administració** : Pàgina accessible pels professors per tal de modificar el contingut de la web, per accedir has de passar un *User/Login* que utilitza un xifrat bcrypt hash enllaçat amb la base de dades.

En aquesta pàgina el professorat pot modificar els següents atributs de qualsevol pàgina de la web:

- **Afegir** : Creació de nou contingut. Exemple: Inserir Animal, inserir nova tècnica, creació d'un quiz, inserta paraula glossari, etcètera.
- **Modificar** : Modificar qualsevol contingut prèviament inserit.
- **Eliminar** : Eliminar qualsevol contingut prèviament inserit.
- **Puja Imatges** : Inserta/Modifica les imatges de la web.

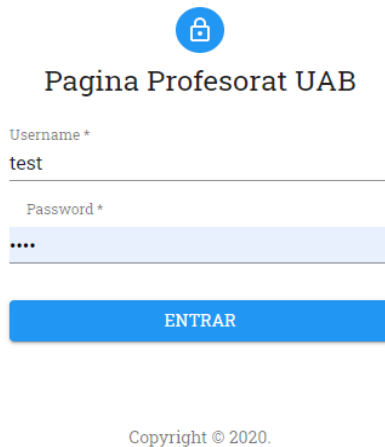


Fig. 12: Pàgina Login Professorat UAB

9 TEST/PROVES

Per tal de garantir una alta disponibilitat, hem hagut de capturar tots els casos d'ús i garantir un correcte funcionament sota errors, així com controlar totes les excepcions, missatges d'errors i combinacions de totes les casuístiques per tal de trobar bugs.

Les proves que s'han realitzat per l'aplicació desenvolupada han sigut realitzades tant en el servidor com en frontal de l'aplicatiu. Principalment tenim dues categories diferents de proves.

- **Proves individuals per cada component** : Aquestes proves busquen testear que els elements que es mostren en la vista funcionen de la manera esperada, una part fonamental d'aquest testeig és comprovar l'alineament/posicionament en els diferents dispositius mòbil així com la seva propietat responsiva.
- **Proves d'integració** : Un cop testejat els components de manera individual revisem com es mostren els elements a la vista quan els agrupem en una mateixa pàgina, comprovem també la seva propietat responsiva per tots els dispositius.
- **Proves API** : Comprovar totes les casuístiques, com les excepcions: quan retorna un valor buit, quan enviem un overflow (valor molt gran), un camp requerit com a buit, etcètera.

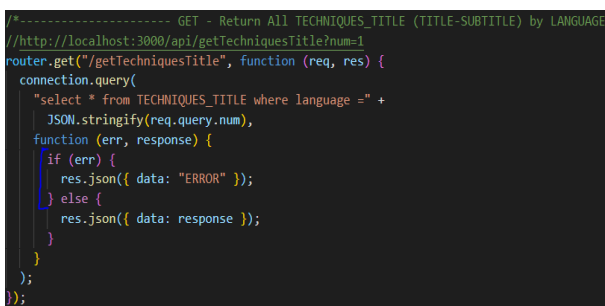


Fig. 13: Captura d'excepcions dintre de les rutes de l'API

- Proves amb la BBDD: Comprovar que es controla correctament una caiguda de la base de dades (la pàgina continua funcionant però sense contingut) i també la casuística de què la BBDD no contingui dades.

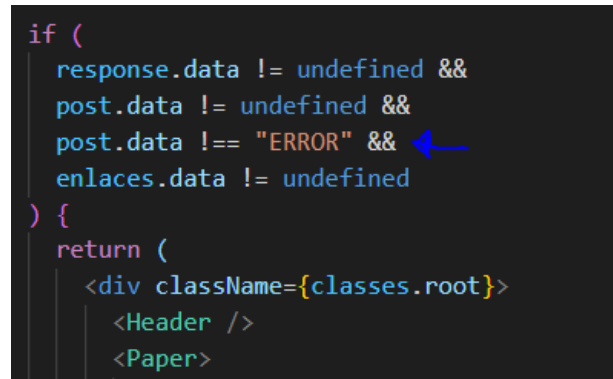


Fig. 14: Tractament d'events segons el contingut

10 CONCLUSIONS

Hem pogut arribar a un producte d'alta qualitat minimalista i plenament funcional / educatiu el qual és una eina molt pràctica / ràpida, ja que es basa en una tecnologia web-app fàcilment accessible. Amb tot això aconseguirem divulgar i forma en el coneixement de la biologia animal en un format més actualitzat.

A nivell personal aquest projecte ha suposat un gran repte, ja que per una part suposava a enfrontar-se a un projecte real (amb client final i product owner) aplicant tots els rols en el desenvolupament software (*DevOps / Front End / Back End*) i per altra banda suposava enfrontar-se a un stack tecnològic amb molt poca experiència prèvia (*MySQL/MariaDB – Express – React - NodeJS*) el qual ha suposat una formació autodidàctica basada en projectes. Ha estat tot una experiència molt enriquidora que m'aporta una visió com a freelance i tota la part de tractar amb el client directament, així com l'extracció de requisits i la validació constant de les necessitats a mida.

11 LÍNIES OBERTES

En aquesta secció s'exposen tots els punts del projecte que es considera que necessiten alguna refactorització o alguna nova implementació de cara al futur. Malauradament com el projecte està limitat pel temps i el procés de creació és una cosa que es va refinant dia rere dia, ens hagués agradat realitzar les següents millores:

- Implementació de seguretat (*JSON Web Token*), per garantir una completa seguretat de l'API i totes les sol·licituds.
- Traspàs a *MongoDB* , ja que l'aplicació es va pensar primerament amb una base de dades no relacional (Formada per documents textit JSON).
- Optimització del codi reduint la lògica del programa reutilitzant components i així disminuir al màxim les línies de codi.

AGRAÏMENTS

En primer lloc, agrair sincerament a la Universitat Autònoma de Barcelona per confiar en mi per portar endavant aquest projecte i per incentivar aquest tipus de propostes, que crec que són molt importants, ja que has de combinar tot el coneixement aplicat en projectes reals. En segon lloc a Juan Carlos Balasch (Professor de la facultat de Biologia) per la fàcil integració amb ell, sempre facilitant i ajudant en tot amb la mesura del possible. En tercer lloc i no per això més importat a la tutora Yolanda Benitez, que m'ha ajudat moltíssim amb el tema de la planificació del temps, tasques i entregues. També agrair a la meva família pel suport rebut i comprensió.

REFERÈNCIES

- [1] <http://en.wikibooks.org/wiki/LaTeX>
- [2] Node.js - Informació de NodeJS. [En línia]. Disponible a: <https://nodejs.org/es/about> Consulta: 14 de Juny del 2020.
- [3] React – Biblioteca de JavaScript . [En línia]. Disponible a: <https://es.reactjs.org> Consulta: 14 de Juny del 2020.
- [4] Material-UI – ReactUI Framework . [En línia]. Disponible a: <https://material-ui.com/> Consulta: 14 de Juny del 2020.
- [5] Javascript Information . [En línia]. Disponible a: <https://developer.mozilla.org/ca/> Consulta: 14 de Juny del 2020.
- [6] Trello - Scrum Board . [En línia]. Disponible a: <https://trello.com> Consulta: 14 de Juny del 2020.
- [7] Draw.IO - Diagram Software . [En línia]. Disponible a: <https://app.diagrams.net/> Consulta: 14 de Juny del 2020.
- [8] AdobeXD - UX/UI Design Tools . [En línia]. Disponible a: <https://www.adobe.com/products/xd.html/> Consulta: 14 de Juny del 2020.

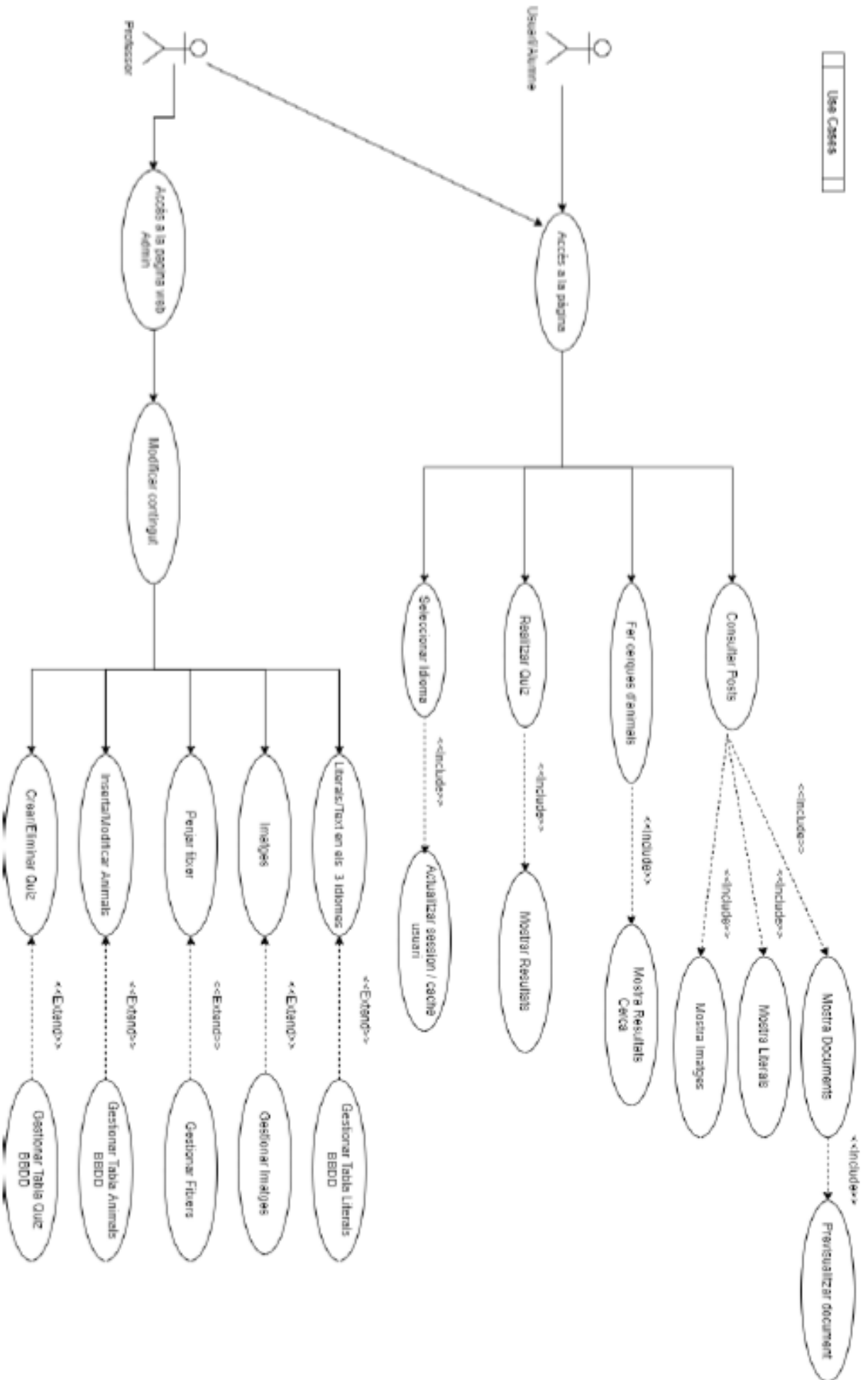


Fig. 16: Diagrama de casos d'ús

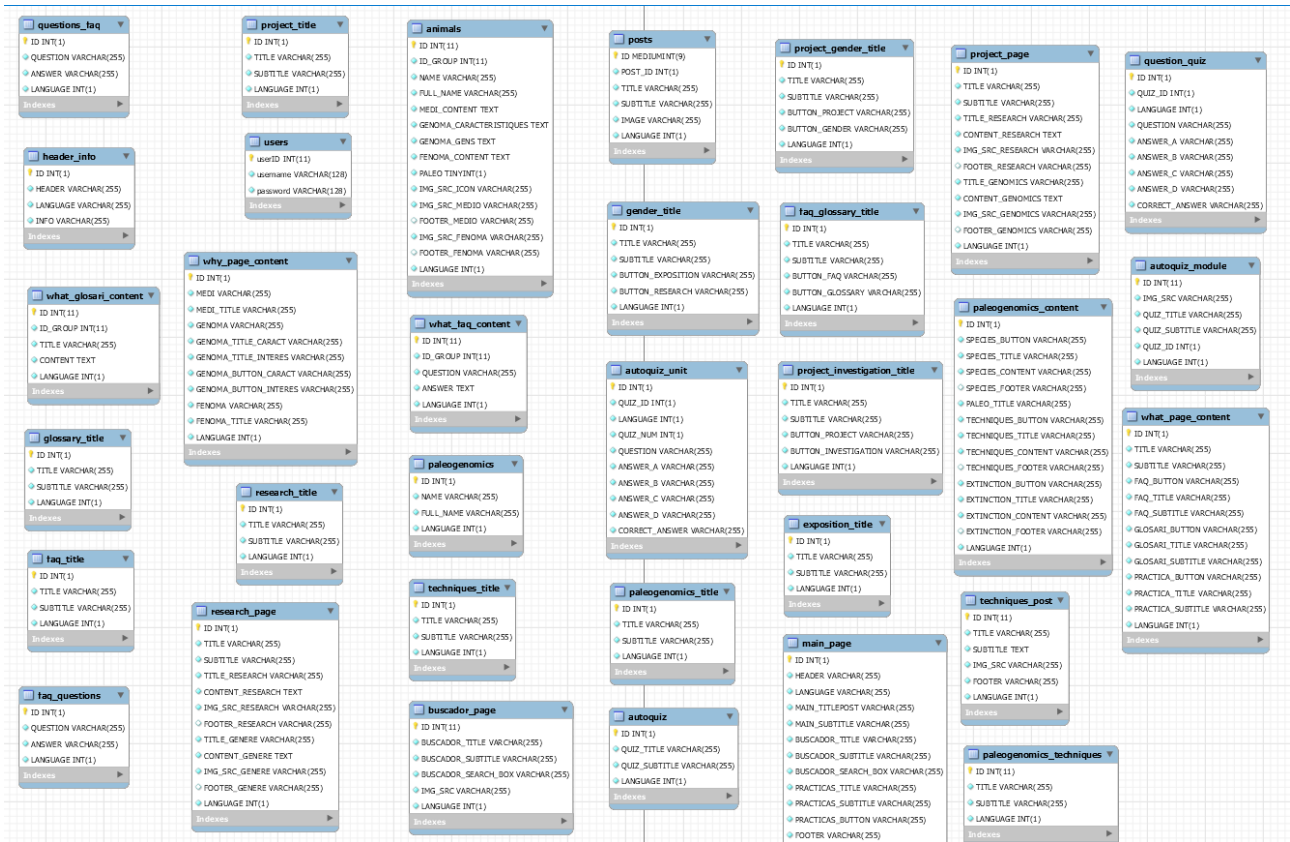


Fig. 17: Schema Base de dades

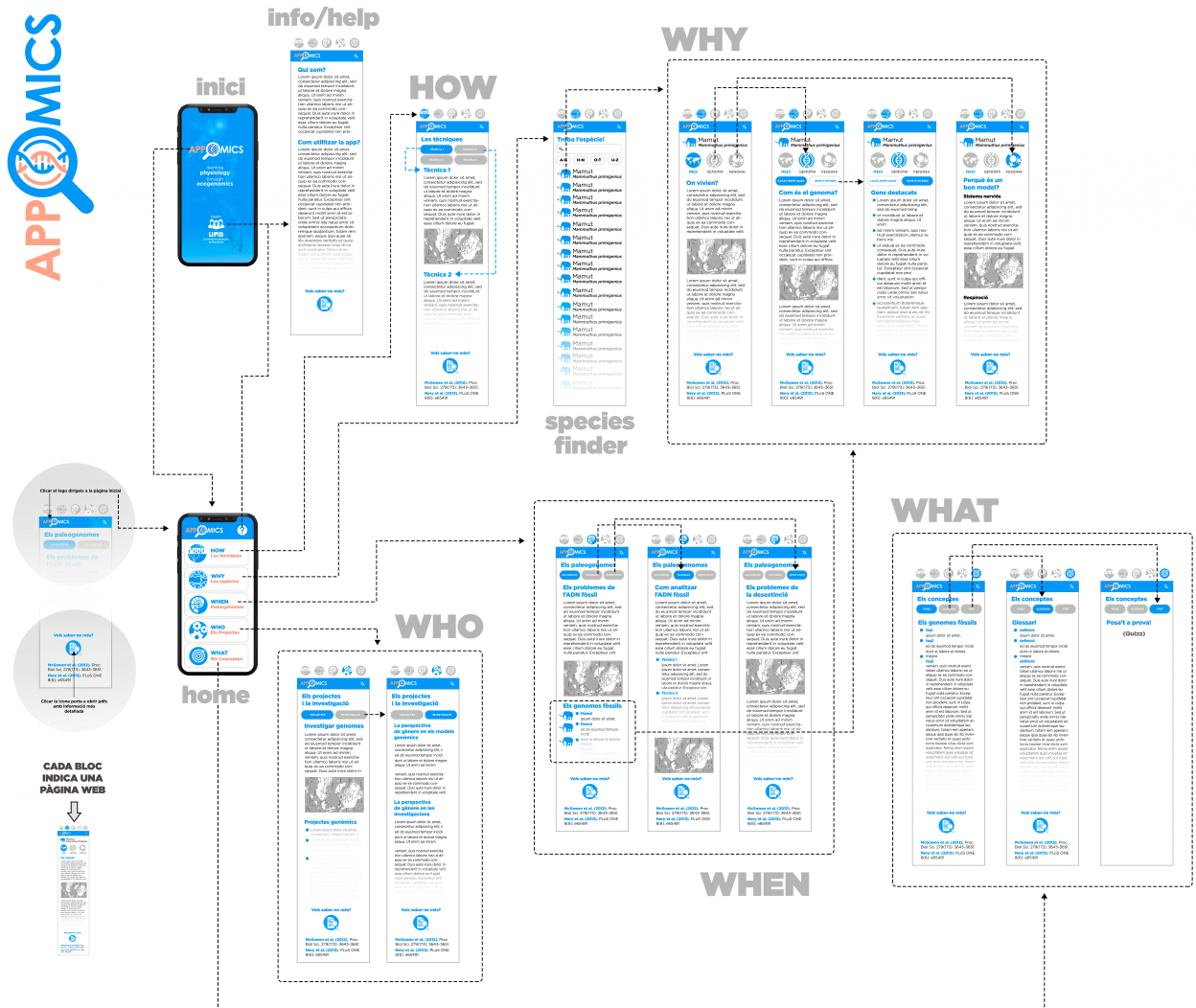


Fig. 18: Estructura APPOMICS

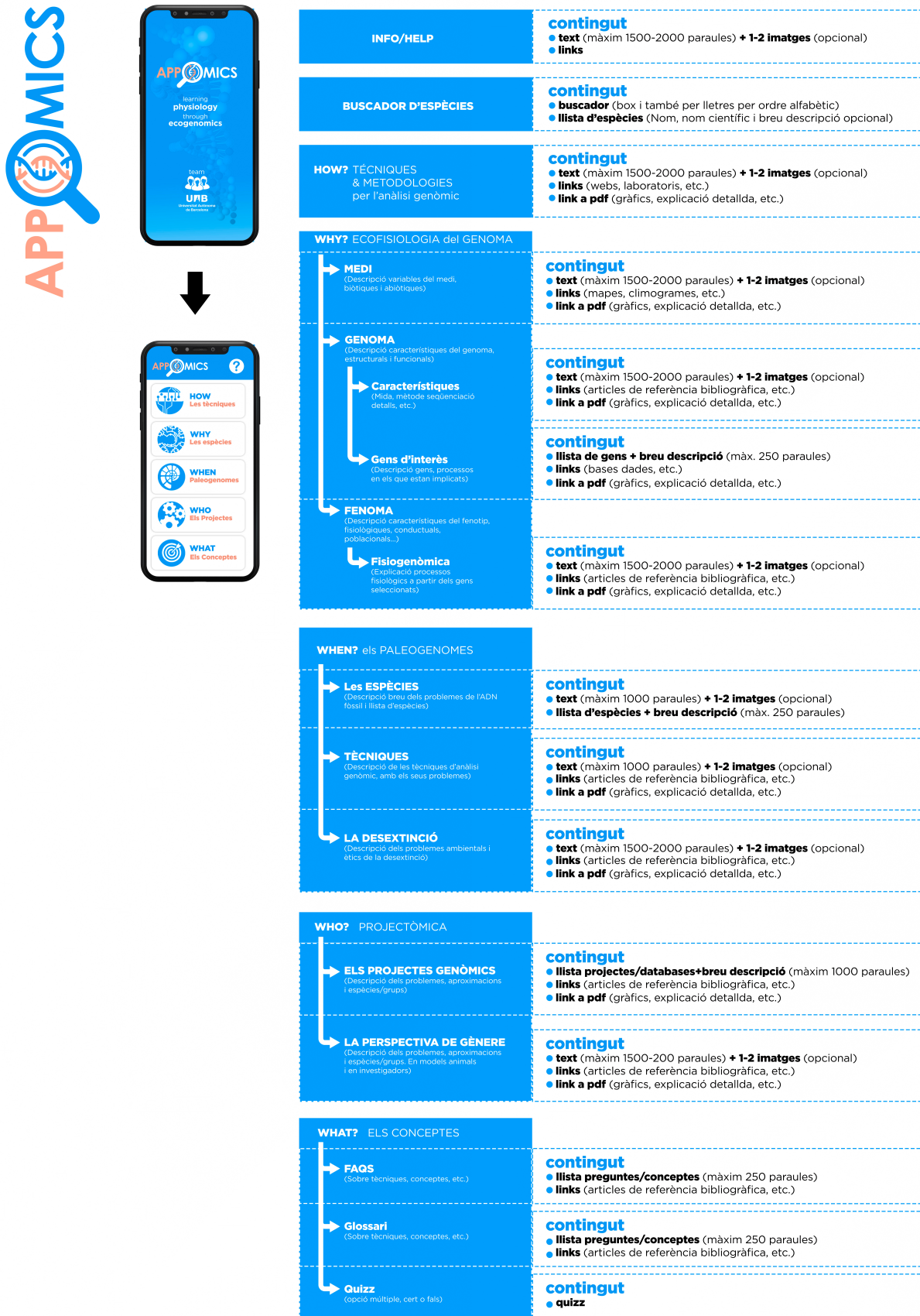


Fig. 19: CONTINGUT APPOMICS