



UNIVERSITAT DE  
BARCELONA

**Treball de Fi de Grau**

**GRAU D'ENGINYERIA INFORMÀTICA**

**Facultat de Matemàtiques i Informàtica  
Universitat de Barcelona**

---

**Disseny d'un sistema informàtic per a la  
digitalització i consulta de corpus de varietats  
funcionals de la llengua catalana**

---

**YI SHEN**

Director: Eloi Puertas  
Realitzat a: Departament de  
Matemàtiques i Informàtica  
Barcelona, 25 de juny de 2019

## **Resum**

Aquest projecte se situa principalment en l'àmbit de les bases de dades, on s'implementaràn mètodes per poder afegir, modificar i borrar les dades i en l'àmbit del desenvolupament backend, que s'interacciona amb el servidor a través del llenguatge PHP. Concretament es basa en la creació d'una pàgina web que serveix per fer cerques que localitzi i mostri el context de la forma lingüística de que s'hagi cercat en la base de dades dels corpora de varietats funcionals de la llengua catalana contemporània.

## **Resumen**

Este proyecto se ubica principalmente en el ámbito de las bases de datos, donde se implementan métodos para añadir, modificar y borrar los datos y en el ámbito del desarrollo backend, que interacciona con el servidor a través del lenguaje PHP . Concretamente se basa en la creación de una página web que sirve para realizar búsquedas que localice y muestre el contexto de la forma lingüística de que se haya buscado en la base de datos de los corpora de variedades funcionales de la lengua catalana contemporánea.

## **Abstract**

This project is mainly located in the field of databases, where methods will be implemented to add, modify and erase data and in the field of backend development, which interacts with the server through the PHP language . Specifically, it is based on the creation of a web page that serves to search to locate and show the context of the linguistic form that has been searched in the database of the corpora of functional varieties of the contemporary Catalan language.

# ÍNDEX

<b>1. Introducció</b> .....	<b>5</b>
1.1 Objectius .....	6
<b>2. Planificació</b> .....	<b>7</b>
2.1 Planificació temporal .....	8
2.2 Diagrama Gantt .....	8
<b>3. Anàlisi del sistema</b> .....	<b>9</b>
3.1 Requisits funcionals.....	9
3.2 Requisits no funcionals .....	9
3.3 Rols .....	9
3.4 Casos d'usos.....	10
3.4.1 Usuari anònim.....	10
3.4.2 Usuari logat .....	12
3.4.3 Administrador.....	14
3.5 Diagrama de seqüències .....	16
3.5.1 Afegir dades.....	16
3.5.2 Sign up .....	17
3.5.3 Editar dades .....	18
3.5.4 Eliminar dades .....	20
<b>4. Disseny</b> .....	<b>21</b>
4.1 Tecnologies utilitzades .....	21
4.2 Base de dades .....	24

4.3 Model Relacional.....	26
4.5 Taules.....	27
<b>5.Implementació .....</b>	<b>32</b>
5.1 Creació de pàgina web a partir de PHP y MySQL.....	32
5.2 Connexions.....	32
5.3 Sessions i restriccions.....	33
5.4 Control flux.....	34
5.5 Extracció de dades.....	34
5.6 Inserció i modificació de dades.....	36
5.7 Eliminació de dades.....	38
5.8 Validació .....	39
5.9 Ajax.....	39
5.10 Exportar dades a excel .....	40
5.11 Autocompletar.....	41
<b>6.Proves i resultats .....</b>	<b>42</b>
<b>7.Conclusions .....</b>	<b>43</b>
<b>8.Bibliografia .....</b>	<b>44</b>
<b>9.Annex. Manual d'usuari .....</b>	<b>45</b>

## 1.Introducció

El corpus de català contemporani és un arxiu de corpus de llengua catalana oral contemporània que ha estat confegit per a l'estudi de la variació dialectal (COD: Corpus Oral Dialectal), on estudien i analitzen els diferents processos fònics que tenen lloc en els dialectes catalans. Social (COS: Corpus de Varietats Socials) i funcional (COC: Corpus Oral de Conversa Col.loquial, COR: Corpus Oral de Registres), que estudia la variació funcional a partir dels fets de parla( conversa col.loquial, discurs polítics..)

Existeix una pàgina de publicacions que inclou els resultats dels diversos projectes d'investigació, on una de l'aportació a la investigació sobre la pragmàtica del català ha consistit a editar i posar a disposició dels investigadors diversos corpus de llengua catalana oral.

Totes aquestes informacions es poden trobar en un dipòsit digital, el CCCUB, que posa a disposició dels investigadors la transcripció i una part dels arxius sonors dels textos del corpus.

El pròposit d'aquest projecte és crear una pàgina web on permet a l'usuari fer cerques de paraules i mostrar el seu context en el corpus. La idea d'aquesta web sorgeix del grup GrEPAD ( Grup d'Estudis de Pragmàtica i Anàlisi del Discurs), que és un equip de treball interuniversitari on està integrat part per professors de la UB del departament de Filologia Catalana i Lingüística General. Que tenen com a objectiu l'estudi de la variació lingüística.

## 1.1 Objectius

L'objectiu principal d'aquest projecte és crear una aplicació web que s'encarregui de gestionar corpora de varietats funcionals de la llengua catalana contemporània.

On permet a l'usuari fer cerques que localitzi i mostri el context de la forma lingüística que s'hagi cercat. I digitalitzar els corpus de varietats funcionals construint una base de dades que permet afegir, eliminar o modificar aquests dades.

Per tal de poder assolir aquest objectiu, s'hauria d'entendre i poder manipular els següents conceptes:

- FrontEnd (HTML, CSS, Javascript)
  - Creació d'una pàgina web que permet a l'usuari introduir la paraula o el símbol a cercar, afegir nous dades a la base de dades i modificar-les.
- BackEnd ( en aquest cas, PHP), que s'encarrega d'interactuar amb la base de dades
- Base de dades
  - Creació de la base de dades amb les taules i camps necessaris.
  - Manipular la inserció i l'eliminació dels diferents camps.

Les eines per aconseguir els objectius desitjats són els següents:

-PHP

-Mysql

-Bootstrap i l'ús d'ajax per què la usabilitat sigui lleugera.

-FileZilla, per pujar el codi al servidor web.

## 1.2 Motivació

Un dels principals motivacions a l'hora d'escollir aquest projecte era perquè tenia interès sobre els diferents llenguatges de la xarxa, i posar en pràctica els conceptes que havíem vist durant les diferents assignatures de la carrera, com el llenguatge php, mysql... que són molt populars per al desenvolupament de la web.

## 2. Planificació

### 2.1 Fases de desenvolupament

#### 2.1.1 Primera fase

Es va confirmar les eines a utilitzar per desenvolupar el projecte. I ens van donar el username i passwords tant de servidor de la web com el de la base de dades.

Es va començar a preparar-se els entorns de treball necessaris per dur a terme el projecte, com la instal·lació del servidor en local MAMP, php, myapache i MySQL.

#### 2.1.2 Segona fase

Es va crear una primera versió de la base de dades i les funcionalitats de la pàgina web. On amb les reunions que vam tenir més o menys cada dos setmanes, es va anar modificant l'estructura de la base de dades i refinant les diferents opcions que ofereix la web.

#### 2.1.3 Tercera fase

Un cop que les funcionalitat de la pàgina web estiguin més o menys acabada, provar que tot funcioni de manera correcte, i fer petites modificacions si cal.

Es tenia pensat que durant el mes de juny, es dediqui el temps per fer proves del projecte i acabar la redacció de la documentació.

Per tant, es pot dir que la metodologia utilitzat per organitzar el projecte és el model incremental. On el projecte es va evolucionant amb cadascuna de les petites entregues fins que s'emmotlla al requerit.

## 2.2 Planificació temporal

Distribució del temps, amb les dades de les entregues

Title	Start Date	Due Date
▼ tfg	11/02/2019	27/06/2019
Planificació	11/02/2019	25/02/2019
Creació base de da...	25/02/2019	04/03/2019
Pàgina modificar, el...	05/03/2019	19/03/2019
Cerques a la base d...	25/03/2019	22/04/2019
Autocomplete, expo...	24/04/2019	06/05/2019
Modificació pàgina ...	08/05/2019	31/05/2019
Memòria i prova	03/06/2019	27/06/2019

FIGURA 1: Planificació temporal

## 2.3 Diagrama de Gantt

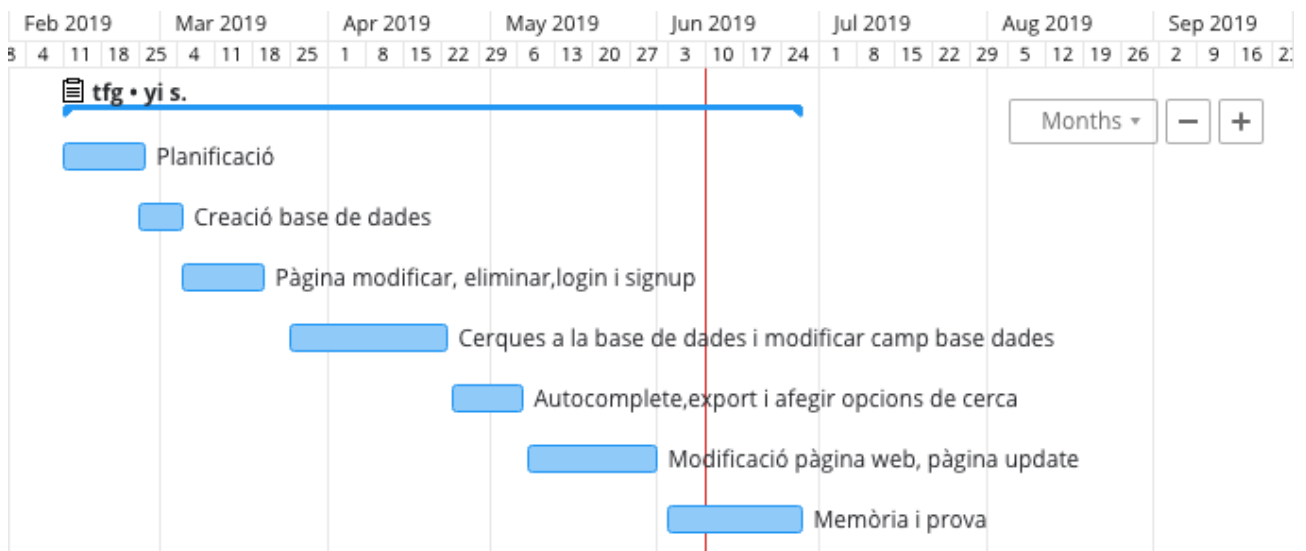


FIGURA 2: Diagrama de Gantt



## 3. Anàlisi del sistema

### 3.1 Requisits funcionals

Tots els usuaris que entren a la pàgina web tenen diferents objectius, com:

- Buscar informació introduïnt la paraula, el símbol..
- Exportar el resultat obtingut a csv, xls o text
- Insertar dades noves a base de dades a través d'un formulari en la pàgina d'afegir dades.
- Modificar camp d'informació del resultat cercat.
- Eliminar informació de la base de dades.

### 3.2 Requisits no funcionals

Que són aquelles funcions que no formen part dels requisits fundamentals del projecte, tals com el disseny, seguretat o la usabilitat..

La pàgina web es dotarà amb un estil minimalista i amb dissenys que permet a l'usuari accedir als diferents parts de la pàgina de manera fàcil i intuïtiva. I garantir que l'usuari serà capaç de completar els casos d'ús de manera ràpida i senzilla

-Requisits de seguretat: s'ha garantit la confidencialitat i integritat de les informacions. De forma que algunes accions només serà exclusius pels integrants del grup GrEPAD.

### 3.3 Rols

No tots els usuaris poden accedir a totes les funcionalitats de la web, per tant, existeix diferents rols.

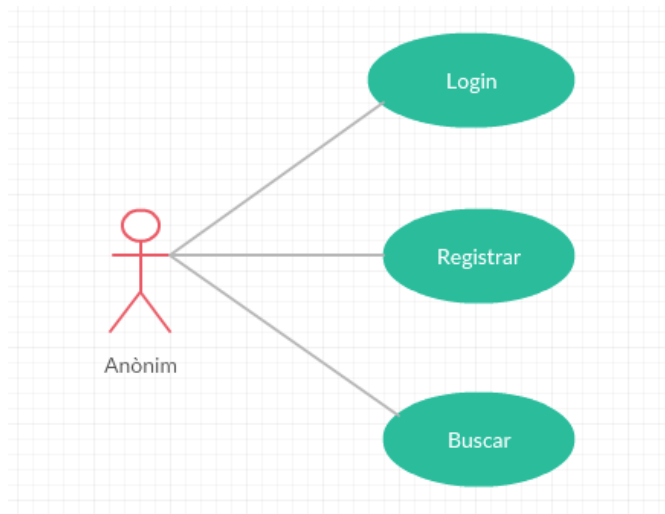
- Administrador: persona amb accés complet a totes les funcionalitats. Capaç de crear, modificar, eliminar, i buscar dades.
- Usuaris registrats: persona que además de buscar, pot afegir dades.
- Usuaris no registrats: persona que només pot accedir a la funció de cerca.

### 3.4 Casos d'usos

Una vegada ja tenim definit els diferents rols i els requeriments, es poden desenvolupar els diferents casos d'usos del sistema.

#### 3.4.1 Usuari anònim

L'usuari que no està logat, només pot realitzar la funció de la cerca en la pàgina principal. I una vegada obtingut els resultats, no són capaços de modificar o eliminar les dades trobades. En cas que es vol fer les altres accions, cal logar-se si ja està registrat, i sinó, fer el signup primer.



**FIGURA 3:**Casos d'ús usuari anònim

#### UC01: Log in

Use Case 1	LOG IN
Descripció:	El usuari que ja està registrat, pot fer el login, per poder accedir a més funcionalitats de la web

Use Case 1	LOG IN
Actor:	Usuari registrat
Precondició	El usuari està registrat i la base de dades té les dades introduïdes
Fluxe Principal	<ol style="list-style-type: none"> <li>1.L'usuari entra a la pàgina principal i fa clic sobre el botó Login</li> <li>2.L'usuari introdueix el email i el password de l'usuari registrat</li> <li>3.Un cop introduït les dades, fa clic sobre el botó log in</li> <li>4.El sistema valida les dades, i mostra el nom de l'usuari logat i un botó per fer el logout</li> </ol>
Fluxe alternari	4.a Si els valors introduïts no es troben en la base de dades, no es loga, apareixerà un pop-up dient que les dades són incorrectes i es redirigirà a la pàgina principal.
Postcondició	Un cop logat, s'activa la funció d'afegir dades

## UC02:Registrar usuari

Use Case 2	Registrar-se
Descripció:	El usuari vol registrar-se
Actor:	Usuari no registrat
Precondició	L'usuari no està registrat
Fluxe Principal	<ol style="list-style-type: none"> <li>1.L'usuari entra a la pàgina principal i fa clic sobre el botó sign up i es redirigeix a la pàgina de signup</li> <li>2.Envia el formulari amb el username, email i el password</li> </ol>

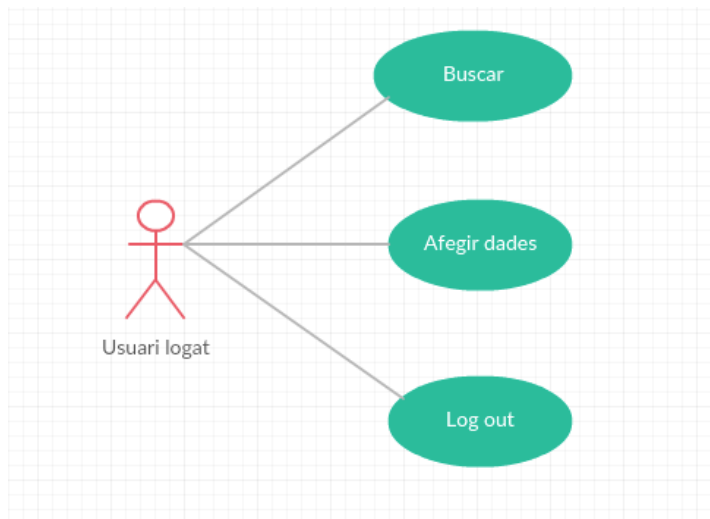
Use Case 2	Registrar-se
Fluxe alternari	2.a L'usuari no pot registrar-se si el email introduït ja es troba en la base de dades
Postcondició	L'usuari es queda registrat, i pot iniciar sessió

### UC03;Buscar dades en la base de dades

Use Case 3	Buscar dades
Descripció:	L'usuari vol fer una cerca de dades en la base de dades
Actor:	Usuari registrat o anònim
Precondició	Ninguna
Fluxe Principal	1.L'usuari entra a la pàgina principal i introdueix el valor que vol buscar en el camp d'entrada, selecciona la forma que vol fer la cerca ( paraula complerta, paraules que conté el valor,etc..) 2.Fer clic sobre el botó de forma o lema dependent de que vol buscar.
Fluxe alternari	2.a No es troba el valor introduït.
Postcondició	Si el valor introduït es troba en la base de dades, llavors, es mostra una taula amb els diferents camps que correspon.

#### 3.4.2 Usuari logat

En cas de que l'usuari està registrat i un cop fet el login, Ara además de buscar dades, pot accedir a la pàgina d'afegir dades, que està composta per un formulari on permet a l'usuari insertar dades a la base de dades.



**FIGURA 4:** Casos d'ús usuari logat

#### UC04: Afegir dades

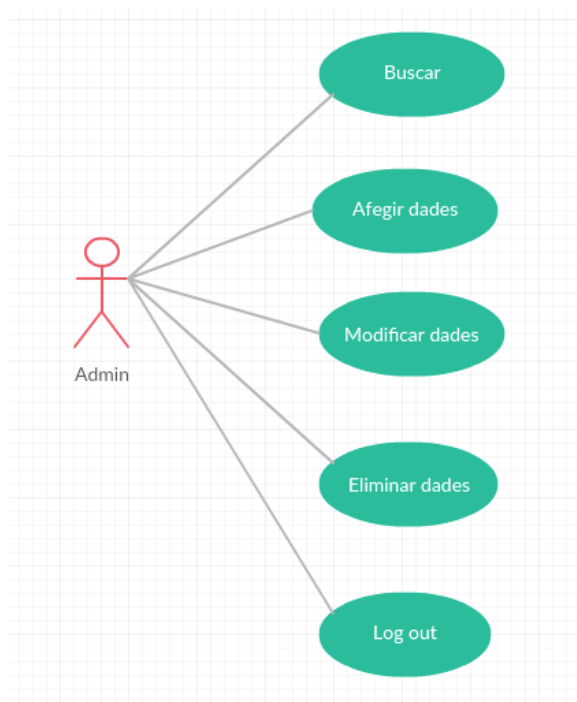
Use Case 4	Afegir dades
Descripció:	L'usuari omple el formulari de la pàgina d'afegir dades per insertar dades.
Actor:	Usuari logat o admin
Precondició:	L'usuari està logat
Fluxe Principal	<ol style="list-style-type: none"> <li>1.L'usuari entra a la pàgina principal i en el cas de que estigui logat, apareixerà un botó d'afegir dades que es redirigeix a una altra pàgina amb un formulari per introduir tots els camps necessaris per poder insertar.</li> <li>3.L'usuari omple el formulari i fa el submit</li> </ol>
Fluxe alternari	<ol style="list-style-type: none"> <li>2.a Algun dels camps estan buits, s'ha d'omplir tots els camps perquè es puguin fer la inserció de dades.</li> </ol>
Postcondició	El sistema guarda la paraula amb tots els camps vinculats.

## UC05: Tancar sessió

Use Case 5	LOG OUT
Descripció:	Permet a l'usuari tancar la sessió.
Actor:	Usuari logat
Precondició	L'actor han d'estar logats
Fluxe Principal	1.L'usuari fa clic sobre el botó logout situat en el header en la pàgina. 3.El sistema tanca la sessió.
Postcondició	Es redirigeix a la pàgina principal.

### 3.4.3 Casos d'usos d'Administrador

Existeix un rol, que a diferència dels usuaris normals logats, además de les accions que aquests poden fer, un administrador pot modificar i eliminar dades.



**FIGURA 5:** Casos d'ús administrador

## UC06: Modificar dades

Use Case 6	Modificar dades
Descripció:	Permet a l'usuari modificar els diferents camps de la dada trobada.
Actor:	administrador
Precondició	<ol style="list-style-type: none"> <li>1.L'actor ha d'estar logat, i el tipus d'usuari ha de ser l'administrador.</li> <li>2.La cerca de dades no es null.</li> </ol>
Fluxe Principal	<ol style="list-style-type: none"> <li>1.L'usuari busca dada, on el resultat es mostra en una taula.</li> <li>2.Selecciona una de ellas, i fer clic sobre el icona de edit</li> <li>3.El sistema mostra la informació completa en un formulari on l'usuari pot modificar els valors.</li> <li>4.L'actor actualitza els valors i fa clic sobre el botó submit.</li> <li>5.El sistema actualitza les dades</li> </ol>
Fluxe alternari	<ol style="list-style-type: none"> <li>4.a El sistema comprova que els camps requerits s'omplen.</li> </ol>
Postcondició	El sistema actualitza la base de dades amb els nous valors.

## UC07: Borrar dades

Use Case 7	Borrar dades
Descripció:	Permet a l'usuari eliminar
Actor:	administrador
Precondició	<ol style="list-style-type: none"> <li>1.L'actor ha d'estar logat, i el tipus d'usuari ha de ser l'administrador.</li> <li>2.La cerca de dades no es null.</li> </ol>

Use Case 7	Borrar dades
Fluxe Principal	1.L'usuari busca dades, on el resultat es mostra en una taula. 2.Selecciona una de ellas, i fer clic sobre el icona d'eliminar.
Postcondició	El sistema elimina la dada seleccionada del sistema.

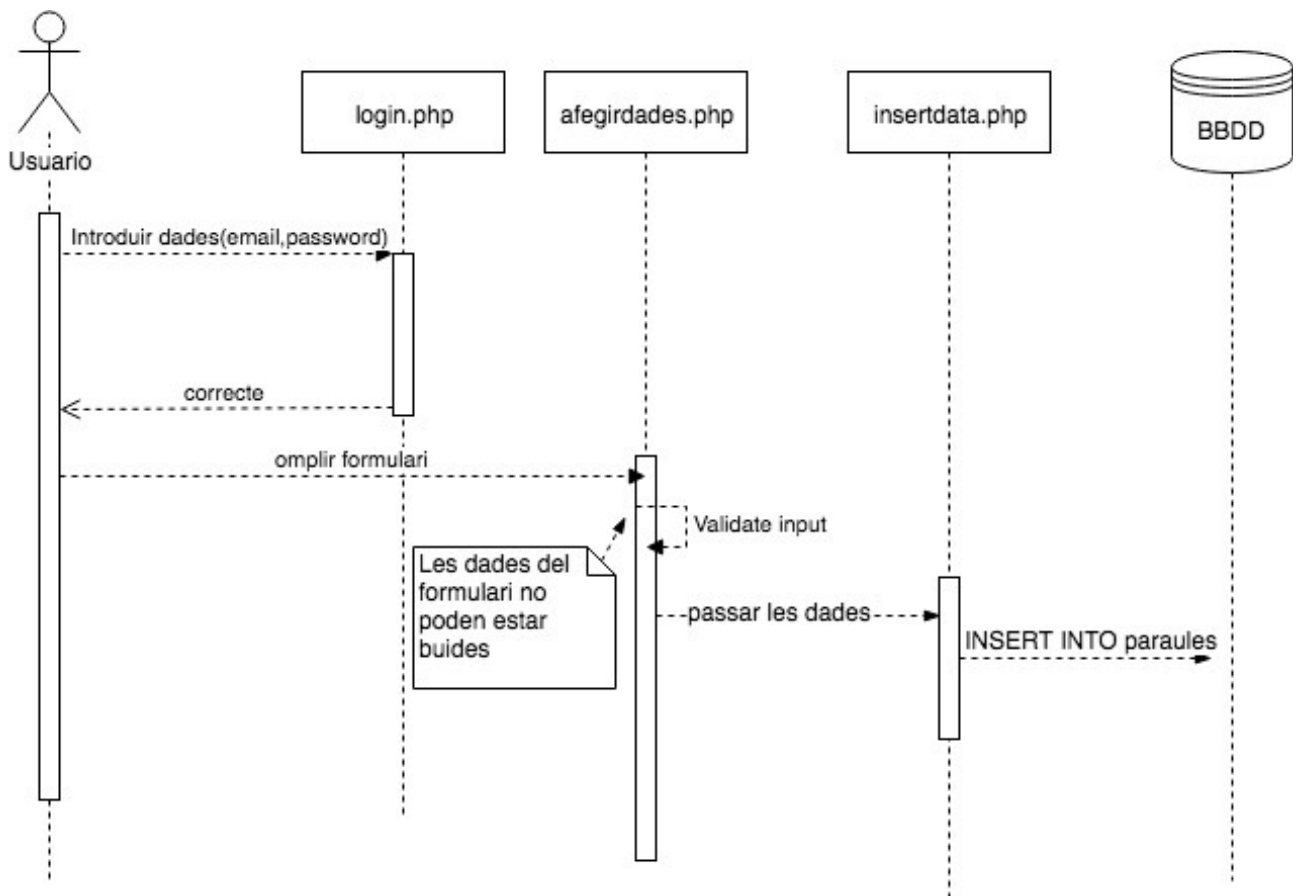
## 4.2 Diagrama de seqüències

Seguidament, mostraré les diagrames de seqüències de les funciones més importants de la web. Els diagrames están adaptats a pàgines PHP que s'ha fet servir per realitzar les funcions.

### 4.2.1 Afegir dades

Fem el login primer per poder insertar dades a la base de dadas. Un cop logat correctament, accedim a la pàgina afegirdades, i omplim el formulari, en cas de que existeix camps buits, la inserció fallarà, i mostrarà un text en color vermell al baix del camp. Un cop que totes les dades s'hagin validat correctament, i fem el submit, es passen a la pàgina insertdata.php, on es recupera els valors passat i els guarden en variables. Fa una connexió amb la base de dades i s'inserta les dades amb INSERT INTO.



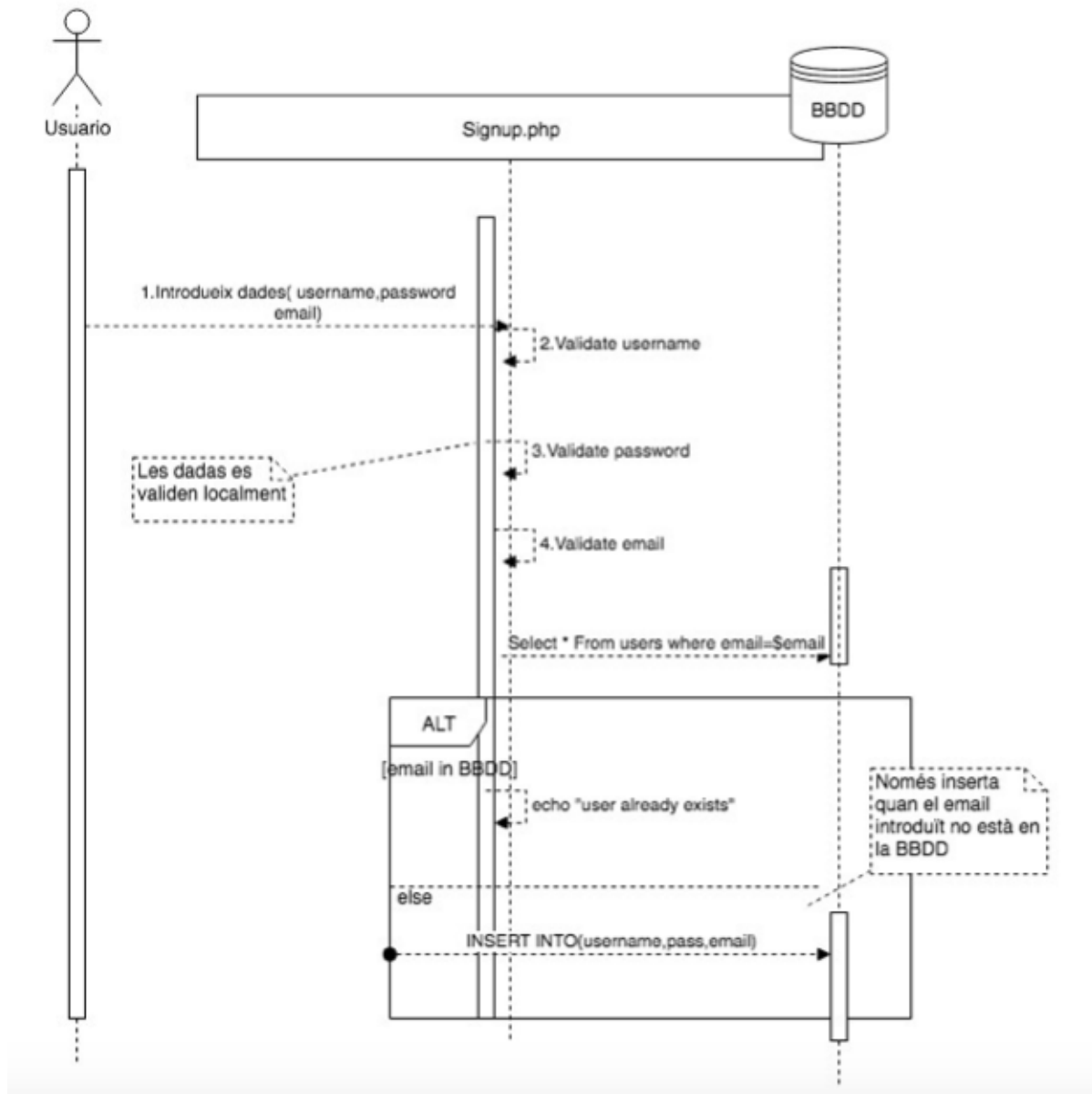


**FIGURA 6:**Diagrama seqüència per afegir dades

#### 4.2.2 Sign up

On abans d'insertar les dades a la tabla users de la base de dades, fa una sèrie de comprovacions. Ja que no tindria sentit si per exemple email té un format incorrecte. I només s'inserta si tots els camps compleixen les condicions.

A més a més, abans d'insertar, es comprova que el email introduït no està registrat, ja que l'usuari per fer el login, s'identifica mitjançant el correu registrat i la seva contrasenya, per tant, un email només pot aparèixer una sola vegada en la base de dades.



**FIGURA 7:**Diagrama seqüència per registrar-se

### 4.2.3 Editar dades

Per poder editar dades, l'usuari hauria d'estar logat en primer lloc com un administrador. Un cop logat correctament, introduïm la unitat a cercar en el camp d'entrada situat a la pàgina principal. Clicant sobre el botó, aquest li passa al fitxer ajax.php totes les variables necessàries, com la unitat a cercar, si es vol buscar en forma o lema, la categoria( que pot estar buida), i el tipus de la cerca(paraula completa, paraula que comencen amb...). Totes aquestes variables hauria de ser identificada perquè depenent d'un valor o altre, es farà un SELECT o un altre. Per exemple, si

volem fer una cerca de paraules que comencen amb la lletra "h", utilitzaríem una expressió regular REGEXP '^"h"'.

Una vegada trobat els resultats, són retornats en format JSON. I els mostrem en una taula, i seleccionem la paraula que volem modificar, passant com a paràmetre id de la paraula al fixer edit.php. Que bàsicament la pàgina consisteix del mateix formulari que el que tenim en la pàgina d'afegir dades, en excepció de que mostren els valors corresponents a l'id de la paraula passat per tal de poder modificar els camps més ràpidament.

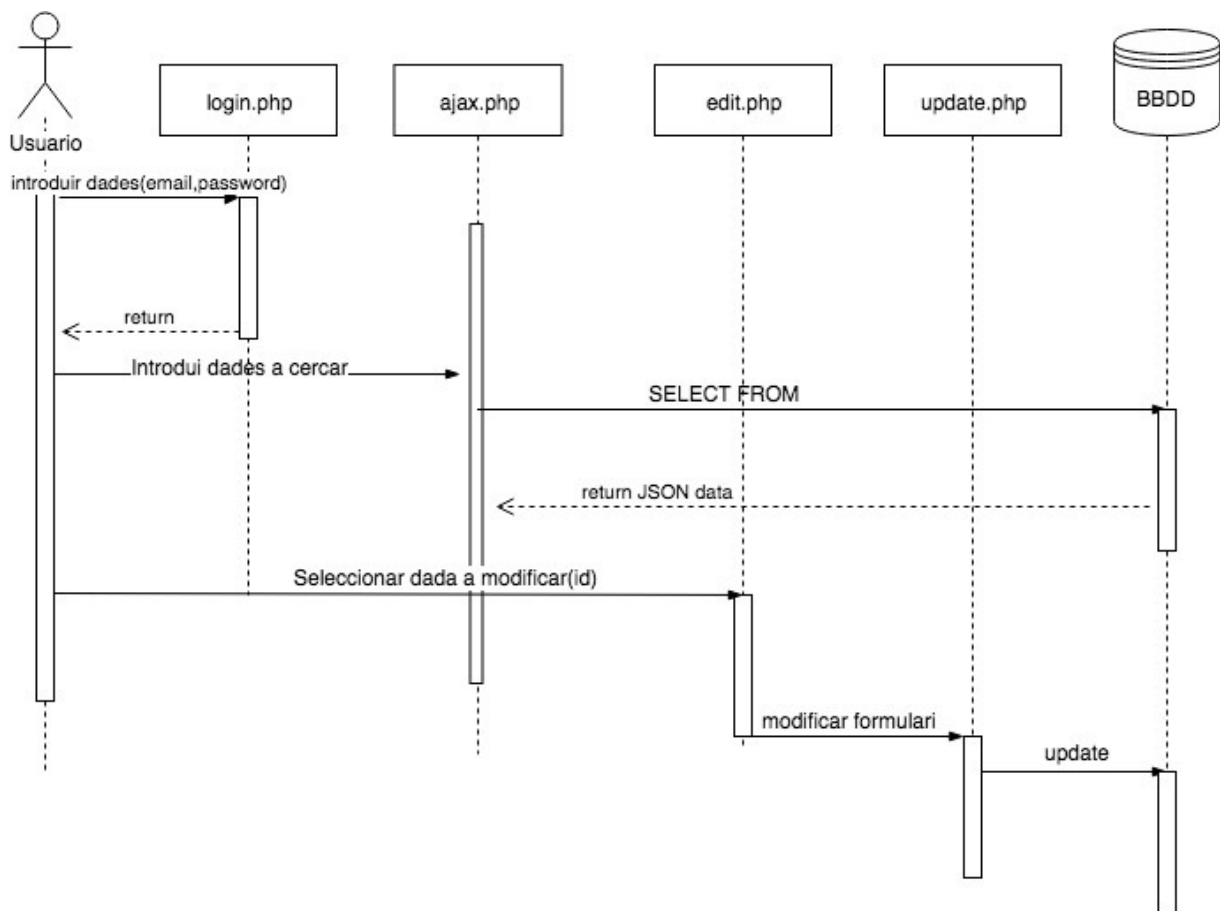
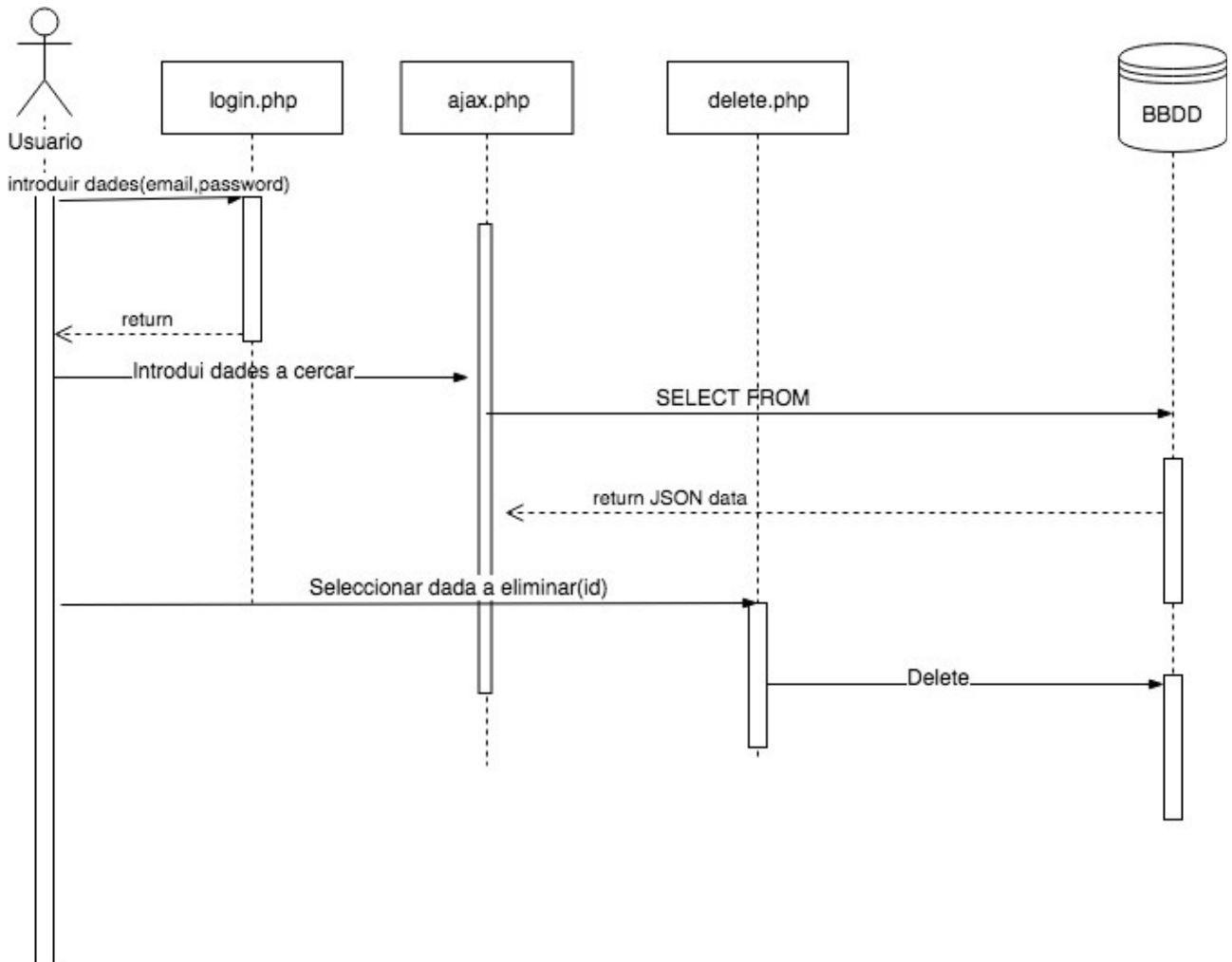


FIGURA 8: Diagrama seqüència per editar dades

#### 4.2.4 Eliminar dades

Per eliminar dades, bàsicament seguim els mateixos passos que per modificadar dades, i ara un cop tenim els resultats mostrats, seleccionem la paraula a eliminar, i es redirigeix a la pàgina delete.php, on allà processarem les funcions necessàries per eliminar les totes les dades corresponents de la base de dades.



**FIGURA 8:**Diagrama seqüència per eliminar dades

## 4. Disseny

### 4.1 Tecnologies utilitzades

Després d'analitzar els diferents requeriments s'ha decidit utilitzar el llenguatge de codi PHP, ja que és un codi obert molt popular que serveix per desenvolupar pàgines webs que funciona en la majoria dels servidors. I pot combinar-se amb mysql per treballar amb base de dades.

I a més degut a la seva popularitat, seria molt fàcil trobar solucions a problemes que es podrien sorgir durant el procés del desenvolupament.

Per la persistència de les dades es farà servir el SQL ( Structured Query Language), i el motor de base de dades MySQL. Que és un sistema de base de dades relacional, multifil, i multiusuari, que es basa en un model client-servidor. Que està disponible com a programa independent per utilitzar-lo en un entorn en xarxa client-servidor i com a biblioteca, que pot ser incrustada (o enllaçada) a aplicacions separades.

Ha esdevingut molt popular gràcies a la seva velocitat en executar consultes i en el nostre cas, com que volem fer en principi una pàgina web que té la funcionalitat principal, fer consultes, ens aniria bé utilitzar aquesta tecnologia.

I es farà servir el llenguatge de Javascript amb la llibreria JQuery, utilitzant la tecnologia de AJAX, que és una tecnologia que ens permet realitzar accions en una pàgina web que necessiten resposta del servidor sense recarregar. I amb això aconseguiríem que la web sigui més dinàmica i que permet una major interactivitat i una millora en la usabilitat.

En quan a disseny i l'aparença de la pàgina web, s'ha fet servir CSS, que és el llenguatge que descriu l'estil d'un document HTML, i indica com es mostraran els elements. Permet al desenvolupador canviarlos localment, i està dissenyat principalment per marcar la separació del contingut del document i la forma de presentació d'aquest, característiques com ara les capes o layouts, els colors i les fuentes.

Además dels CSS, s'ha utilitzat també el framework Bootstrap, per utilitzar algunes de les plantilles que ens ofereix basats en HTML i CSS.

Per tal de manipular els llenguatges mencionat anteriorment, s'ha fet servir un conjunt de software per fer la implementació.

Per part del servidor s'ha haver instal·lat els següents components:

- Servidor HTTP
- Mòdul PHP
- MYSQL

Per tal de posar en marxa , al l'entorn del desenvolupament local s'ha fet servir:

### MAMP

És un entorn de servidor local que potser instal·lat sota macOS i Windows. On l'arònim MAMP es refereix al conjunt de programes software per desenvolupar llocs webs dinàmics.

- Mac OS X: Sistema operatiu
- Apache: Servidor
- MySql: sistema gestor base de dades
- PHP, Perl o Python

Es pot utilitzar per desenvolupar llocs webs sense la necessitat d'un servidor web separat.

### Integrated Development Environment - IntelliJ IDEA

És un entorn de desenvolupament integrat per al desenvolupament de programes informàtics. Durant la carrera, en diferents assignatures, vam utilitzar diferents IDE per escriure codi en JAVA, com Netbeans, Android Studio, etc.. i com que en aquest projecte és té que manipular sobre llenguatge PHP, i amb el email de la universitat, podem obtenir licència de la versió Ultimate d'aquest software, vaig optar d'utilitzar-lo com a eina de desenvolupament del codi PHP.

## Mysqlworkbench

És una eina visual de disseny de bases de dades que integra desenvolupament de programari, administració de bases de dades, disseny de bases de dades, gestió i manteniment per al sistema de base de dades MySQL.

Entrant a la pàgina principal del servidor local que proporciona el MAMP, també tenim l'opció de manipular la base de dades a través de la web phpMyadmin. Tot i que ho vaig trobar el mysqlworkbench més fàcil d'utilitzar.

## FileZilla

Aquest software ens permet la connexió amb el nostre servidor host de forma ràpida i còmode. Com podem veure en la pàgina principal del softwarem, una vegada connectada, podem afegir fitxers arrastrantlos del lloc local fins a lloc remot.

Actualitzant-se el projecte al moment en la nostra web.

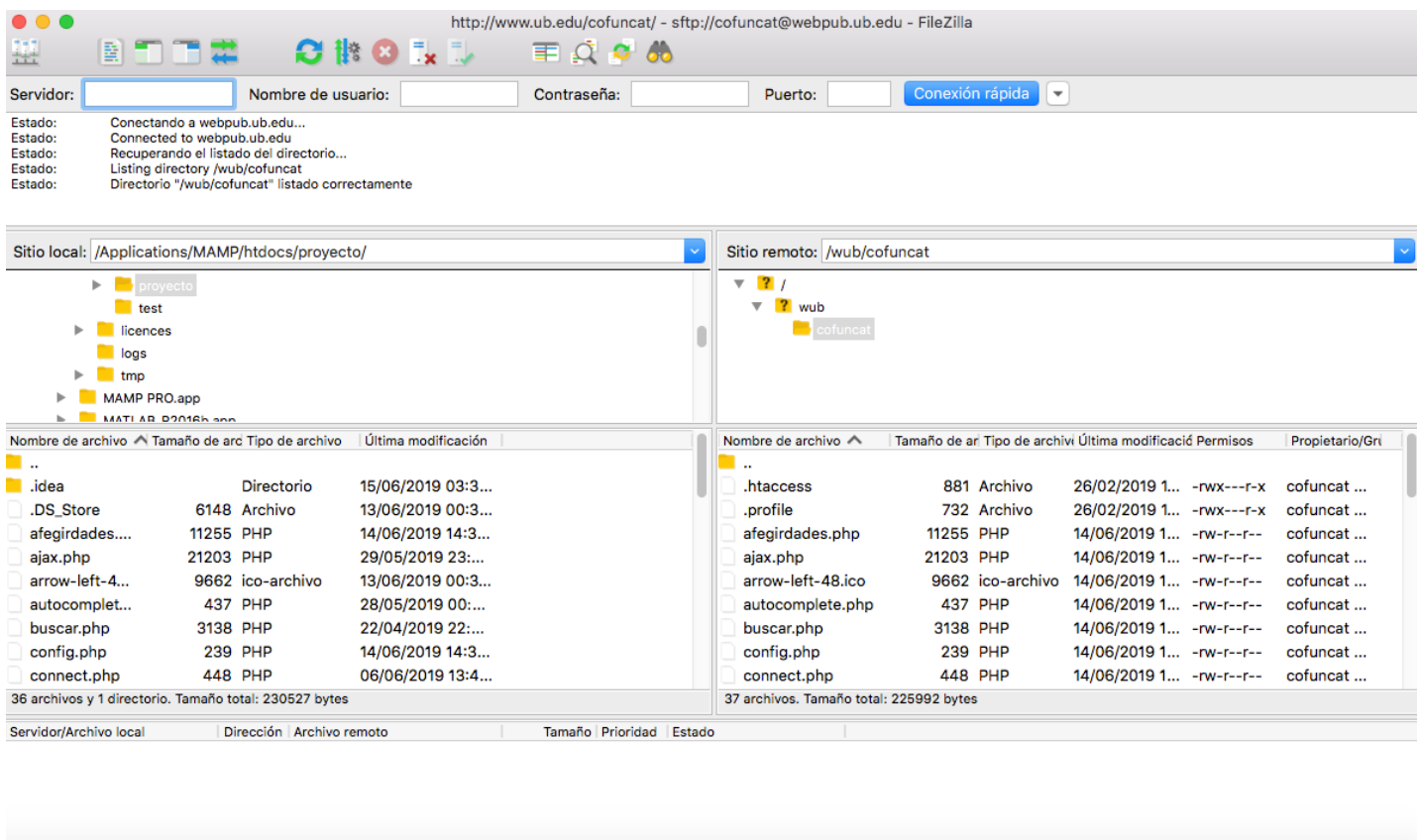


FIGURA 9: Pantalla inicial per pujar fitxers al servidor de FileZilla

## 4.2 Base de dades

En aquest projecte, s'ha optat a utilitzar un sistema de base de dades relacional, que funciona segons el model relacional.

Tal com el seu nom indica, funciona mitjançant relacions, on es té creat una sèrie de taules que es relacionen entre sí. On cada taula, està formada per files i columnes.

Cada fila és única, que s'identifica amb un identificador únic, que representa una única instància d'una entitat, que l'anomenem com a Primary Key, Com que s'identifica a través d'un identificador, l'ordre en que están posicionants les files no importa. Dins de cada fila, es troba un conjunt de columnes, on cadascuna d'elles representa un atribut únic de l'entitat.

### 4.2.1 Model entitat relació

Primer de tot, desenvolupem uns models d'entitat- relació, en referència a la base de dades que hauria de tenir el projecte. On cada entitat representa les diferents taules de la base de dades, i els atributs són els camps on es guarden les informacion. Els atributs que están subrallats representa el Primary key d'aquella taula, és a dir, és el valor que identifica cada taula.

Primer de tot s'ha desenvolupat un model d'entitat- relació per a un usuari, on el projecte, en aquest moment, l'usuari només serveix per identificar-se per tal de poder realitzar unes certes accions privilegiades depenent del tipus. Per tant, està fet de forma en que no està relacionat a cap altre entitat. I s'identifica a través del seu id.

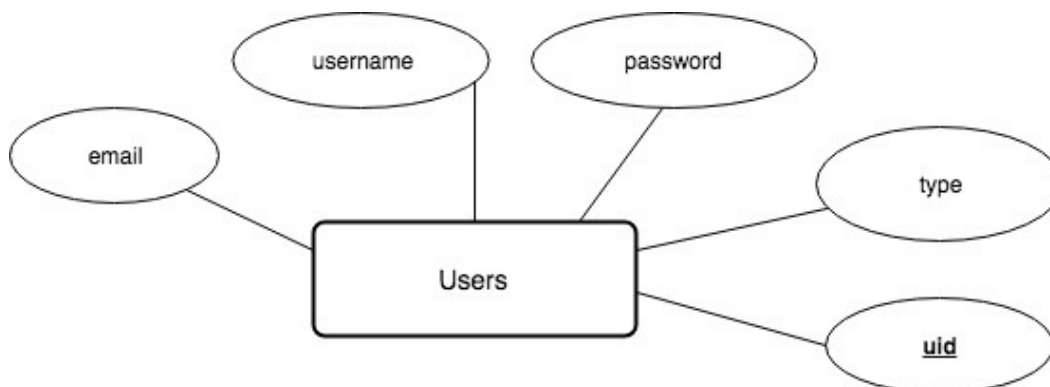
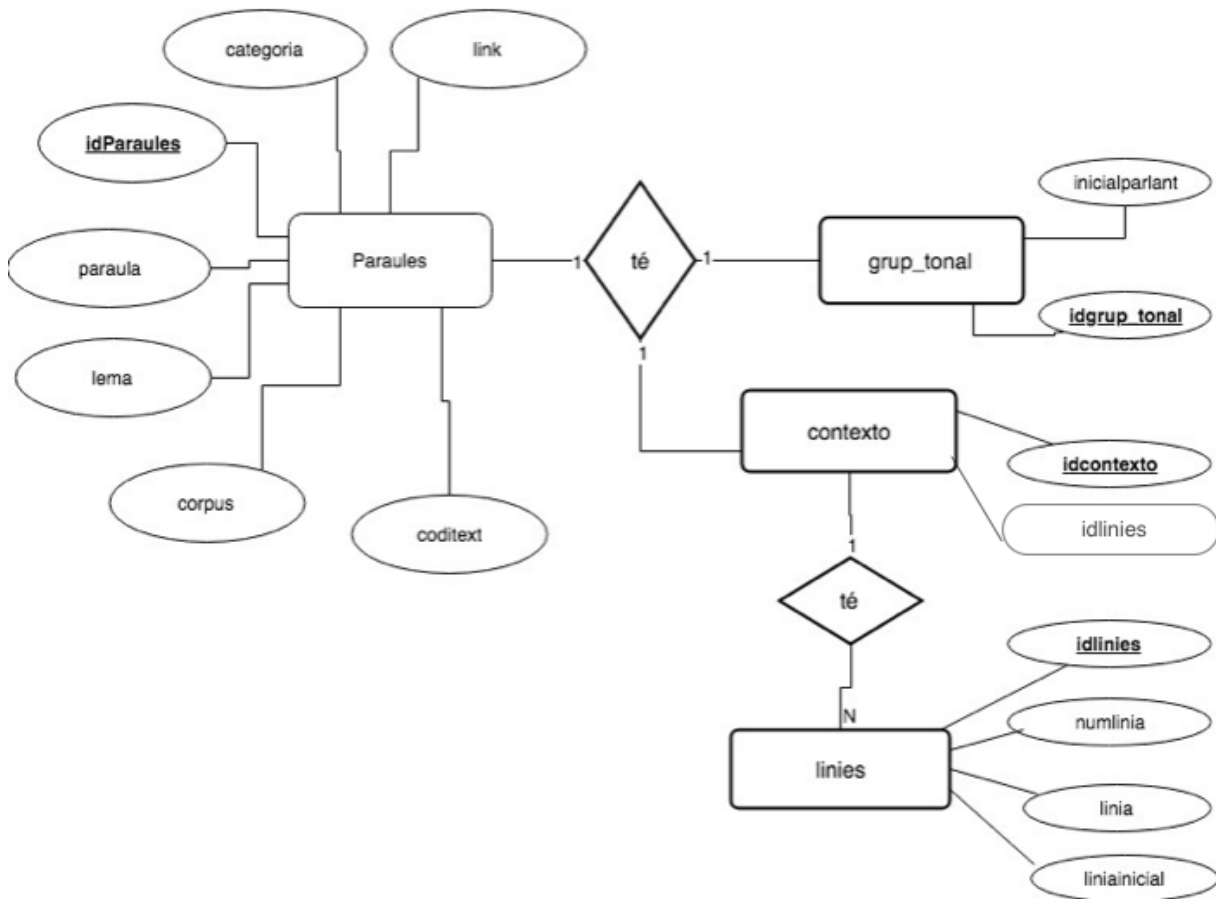


FIGURA 10: Model entitat-relació d'usuari



La paraula té unes característiques fixes, com la categoria, la seva forma i lema, etc, Per a cada paraula que guardem, aquest es troba en un context, per tant, cada paraula està relacionat a un grup tonal, que defineix l'inicial del parlant on apareix la unitat cercada.

I per cada paraula cercada, aquesta es troba dins d'un torn de parla, que especifica el grup de línies que conformen una intervenció concreta del parlant en la conversa. Per tant s'ha decidit crear una taula complementaria que s'anomena "contexto". Permet que cada paraula tingui varies línies amb la mateixa idlinies.



**FIGURA 11:** Model entitat-relació

### 4.3 Model relacional

A continuació s'ensenyen la relació entre les taules de la base de dades, on podem veure que les taules es corresponen amb les diferents entitats del model entitat-relació. On a cada fila de les taules es guarda el registre amb els atributs corresponents.

Cada taula disposa d'una clau primària que la identifica, que en la imatge es mostra amb un icona de clau en color groc com el idParaules, idgrup\_tonal, idcontexto. I per tal d'establir relació entre les dues taules, s'utilitza els foreign keys, que consisteix d'un icona de rombe, que fa referència a la clau primària d'altra taula. En la taula de Paraules, tenim 2 foreign keys, idgrup i idcontext que apunta a la taula grup\_tonal i contexto. I en la taula de línies tenim l'atribut idlinies que apunta al idcontexto de la taula contexto.

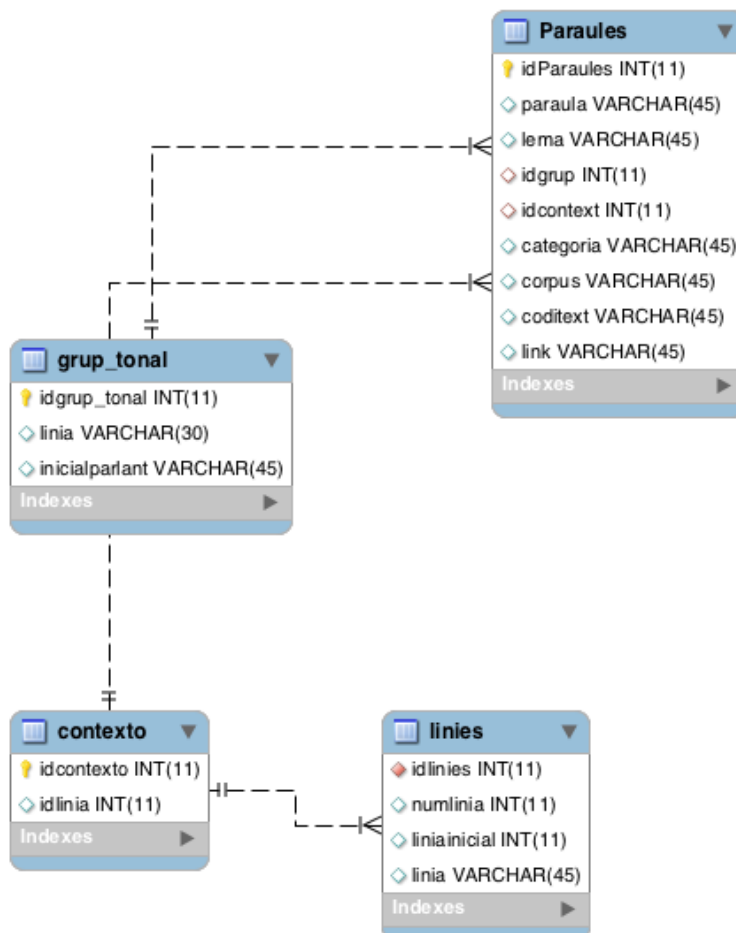


FIGURA 12: Model Relació

## 5.4 Taules

A continuació, es detallarà el contingut de les diferents taules.

### 5.4.1 Taula de users



**FIGURA 13:**Taula d'usuari

Taula que guarda la informació bàsica dels usuari. Conté les següents columnes:

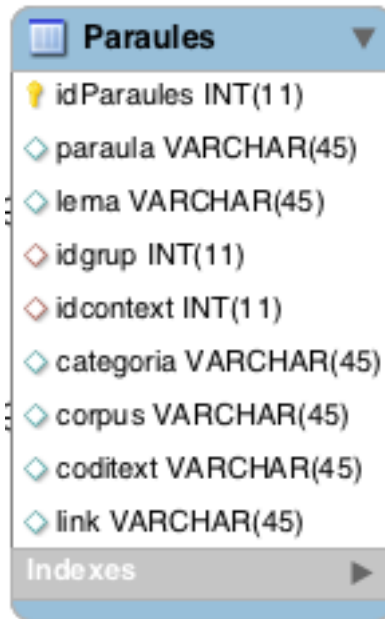
- uid: És la clau primària de l'usuari, està fet que sigui autoincrement, és a dir, el valor s'incrementa 1 cada vegada que afegim un nou usuari.
- user: representa el nom d'usuari registrat.
- pass: la contrasenya per poder identificar-se.
- email: email d'usuari, és el camp que serveix per després fer el login i també comprovar l'usuari existents.
- type: indica el tipus d'usuari, per defecte, quan tu registres un nou usuari, el tipus d'usuari serà 0, que significa que és un usuari normal. I en cas de que type=1 significa que es tracta d'un usuari administrador.

### 5.4.2 Taula de Paraules

És la taula que guarda la informació bàsica de la paraula, relaciona amb un grup tonal i el context en que es troba la paraula. Conté les següents columnes:

- idParaules: clau primària de la taula, està codificat com a autoincrement.
- paraula: forma de la paraula.
- lema: lema de la paraula.
- id\_grup: foreign key que apunta a la taula grup\_tonals.
- id\_context: foreign key que apunta a la taula contexto.
- categoria: categoria gramatical de la paraula, que pot ser nom, verb, pronom...
- corpus: nom corpus que pertany la paraula.

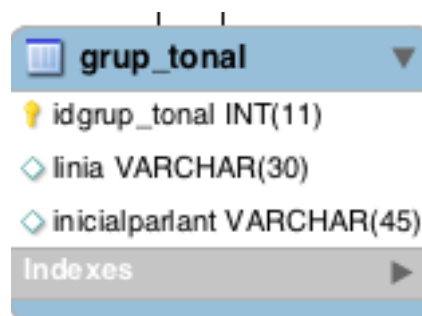
- coditext: nom text del corpus.
- link: enllaç on es troben els corpus.



**FIGURA 14:** Taula de Paraules

#### 5.4.3 Taula de grup tonal

Taula que guarda la informació sobre el grup tonal on apareix la unitat cercada.



**FIGURA 15:** Taula grup\_tonals

- idgrup\_tonal: clau primària que serveix de identificador, està codificat a autoincrement. És el valor que es relaciona amb la taula de Paraules.

- inicialparlant: valor que indica el parlant en la transcripció.

#### 5.4.4 Taula de context

Taula que serveix per relacionar la paraula amb el conjunt de línies de la transcripció.



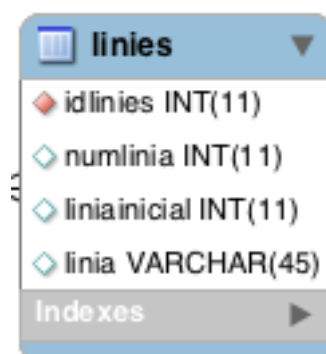
FIGURA 16: Taula de context

- idcontexto: clau primària que identifica la taula i que serveix per establir la relació amb la taula de la paraula.
- idlinia: id de les línies.

#### 5.4.5 Taula de línies

Taula que conté el conjunt de línies que formen part d'un torn de parla.

- idlinies: clau primària que identifica la taula i que relaciona amb la taula de contexte.
- numlinia: nombre de la línia en que es troba en la transcripció.
- liniainicial: nombre de la línia en la qual es comença el torn de la parla per un determinat parlant.
- linia: text de la transcripció.



**FIGURA 17:**Taula de línies

#### 5.4.6 Principals funcions de les taules.

Les funcions més conegudes per a aquest tipus de base de dades, i també les funcions que farem servir per la implementació són les de la lectura, creació, eliminació, modificació.

Funcions	Descripció
SELECT	selecciona les línies de la base de dades que compleix la condició i recupera dades de la taula.
INSERT	Insereix dades a una taula.
DELETE	Esborra tot el registre d'una taula.
UPDATE	Actualizar dada existent de la taula.

Podem crear i modificar l'estructura dels objectes de la base de dades utilitzant les següents funcions.

Funcions	Descripció
CREATE	Crea objectes a la base de dades.
ALTER	Alterna els objectes de la base de dades.
DROP	Esborra els objectes de la base de dades.

Per exemple, per connectar a la base de dades del servidor local, i crear una schema amb el nom de sys, farem les següents implementacions. On li passem el nom del servidor com a localhost, i username i password amb el valor de root. I fem la connexió

```

<?php
$servername = "localhost";
$username = "root";
$password = "root";

// Create connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

```

FIGURA 18:Codi d'exemple per fer la connexió a base de dades

Tot seguit, si es connecta correctament, amb la funció CREATE, cream un schema amb el nom de sys.

```

$sql = "CREATE DATABASE sys";
if ($conn->query($sql) === TRUE) {
    echo "Database created successfully";
} else {
    echo "Error creating database: " . $conn->error;
}

$conn->close();

```

FIGURA 19:Codi d'exemple per crear un schema

I un cop creat la base de dades, podem insertar-li taules, que veurem la creació d'una de les taules com a exemple. Es connecta a la base de dades amb els mateixos codis que abans.

I amb la funció CREATE TABLE, es crea una nova taula, on per a columna, se li assigna el tipus de valor que es guardarà, i tots els primary key i foreign keys necessaris.

```

$sql2="CREATE TABLE `Paraules` (
  `idParaules` int(11) NOT NULL AUTO_INCREMENT,
  `paraula` varchar(45),
  `lema` varchar(45) collate utf8_bin,
  `idgrup` int(11),
  `idcontext` int(11),
  `categoria` varchar(45),
  `corpus` varchar(45),
  `coditext` varchar(45),
  `link` varchar(45),

  PRIMARY KEY (`idParaules`),
  KEY `fk_1_idx` (`idgrup`),
  KEY `fk_2_idx` (`idcontext`),
  CONSTRAINT `fk_1` FOREIGN KEY (`idgrup`) REFERENCES `grup_tonal` (`idgrup_tonal`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_2` FOREIGN KEY (`idcontext`) REFERENCES `contexto` (`idcontexto`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB";

```

FIGURA 20:Codi d'exemple crear una taula

## 5.Implementació

### 5.1 Creació de pàgina web a partir de PHP y MySQL

Utilitzarem llenguatge HTML per crear les vistes de la pàgina web, a dins d'aquesta incrustem el llenguatge PHP per fer la programació de la pàgina, així com codificar les funcionalitats i controlar o establir una lògica de control en la codificació.

El llenguatge PHP ens permet cridar funcions MySQL, que estableixi una connexió amb la base de dades que tenim creada, per fer consultar o gestionar les dades.

De seguida, se explica en detall les funcionalitats més importants per a poder efectuar les funcionalitats de l'aplicació web.

### 5.2 Connexions

Per tal d'establir la connexió entre la BDD i el nostre PHP, primer de tot, tenim creat un fitxer config.php, que s'encarrega d'establir la connexió entre la BDD i el PHP, i el incrustem a tots aquells fitxers dels quals necessiten connectar-se amb la BDD per poder realitzar alguna acció.

S'ha fet dos connexions, una per treballar amb el local host, que és la base de dades que s'ha utilitzat en primer lloc per fer totes les proves en el nostre ordinador localment i l'altra per connectar-se a la base de dades online que ens ha proporcionat. On en la de servidor online, li passem com a paràmetre els valors que ens ha donat per poder fer la connexió.

Hem utilitzat les següents variables per establir la connexió

Servidor: localhost

. Nom de la base de dades : sys

. Nom de l'usuari administrador: root

. Contrasenya: root

En cas de que es tracta d'un servidor online, doncs amb els seus valors corresponents.



### 5.3 Sessions i restriccions

Com que l'adreça HTTP no manté l'estat, i cada cop que canviem de pàgina, les informacions es perden.

Per tal de solucionar aquest problema, utilitzarem les variables de sessió PHP que permet emmagatzemar les informacions necessàries que s'utilitzarà a diverses pàgines. Guarda les dades al servidor, i per a cada sessió es crea un identificador únic que s'utilitza per enllaçar l'usuari amb la seva pròpia informació.

Amb la funció `session_start()` s'inicialitza la sessió i es crea l'identificador. Cada cop que fem el login, guardem les variables com el nom d'usuari, el seu id, i el tipus d'usuari ( que potser 0 si es un usuari normal o 1 si es tracta d'un administrador) en la variable `$_SESSION[]`. Que són accessibles durant tota la vida útil d'una sessió.

Com hem vist abans, no tots els usuaris tenen accés a totes les funcionalitats disponibles, depenent del tipus d'usuari logat, alguns elements es mostrarà o no. S'ha de comprovar l'atribut corresponent de la `$_SESSION()` cada cop que entrem a una nova secció de l'aplicació per tal de limitar l'accés a les diferents zones de la pàgina. En cas de que un usuari sense privilegi, té la direcció de la web de la pàgina que no li correspon, a l'inici de cada fitxer s'ha posat un filtre per comprovar el tipus d'usuari per deixar la pàgina només visible a aquells usuaris que compleix la restricció. I en cas negatiu, es redirigeix directament a la pàgina principal.

Per exemple es podria ocórrer el cas de que un usuari que no està logat accedeix a la pàgina amb URL [ub.edu/cofuncat/afegirdades.php](http://ub.edu/cofuncat/afegirdades.php), aquest no tindria permís per accedir i tindria que ser redirigida a la pàgina principal. Per tant cada vegada que tu entres a una pàgina nova restringida, s'ha de comprovar primer si estàs logat o no. En aquest cas concret, quan tu entres a la pàgina d'afegir dades, comprova que `$_SESSION['user_id']` és True, i en cas negatiu, es redirigeix a la pàgina principal amb la funció `header("Location: ./index.php");`

En el cas on tu vols modificar o eliminar dades, aquest només són accessible pels usuaris admistradors ( usuari amb tipus=1).

Per això, una vegada fet la cerca, i tenim els resultats mostrats en una taula, només l'administrador pot veure el botó d'eliminar i modificar. Abans de mostrar la taula, es comprova primer el tipus d'usuari amb la variable `$_SESSION["type"]`, i en el cas de que el valor sigui 1, li afegim 1 columna a la taula amb el icone d'editar que conté l'enllaç amb l'adreça `edit.php?id=idParaula` i l'altra columna amb el icone d'eliminar amb l'enllaç `delete.php?id=idParaula`.

## 5.4 Control flux

Per tal d'enviar els paràmetres d'un fitxer php a un altre, es fa mitjançant la funció POST a través d'un formulari o mitjançant la funció GET. En el cas d'un formulari, fent clic sobre el botó submit del formulari, crida la funció corresponent a aquell botó, i executa el seu fragment del codi, passant com a paràmetre tots els camps que es troben dins del formulari.

Per exemple, per afegir la dada, un cop hem omplert el formulari, i clicant sobre el botó submit, aquest crida una funció i un cop verificat que tots els camps estigui omplerts, passa tots els camps al fitxer insertdata.php, on allà es processarà l'acció d'insertar.

## 5.5 Extracció de dades

Per extreure dades de la BDD, podem utilitzar la funció `mysqli_query()` de PHP, i si existeix més d'una fila que compleix la condició, per obtenir totes les dades seleccionades, es pot fer servir la funció `mysqli_fetch_array()`, que retorna la fila com una matriu associativa, o una matriu numèrica.

S'utilitza el bucle `while` per recórrer sobre tots els resultats i guardar o mostrar les dades de les columnes desitjades.

Quan volem extreure dades de la BDD, com que tenim una sèrie d'opcions que podem fer quan fem la cerca, com la cerca per paraula, per símbol, paraules començades amb... , agrupats en radiobuttons, la funció de selecció es canviarà depedent del radiobutton seleccionat. Per tant tenim que passar un paràmetre que indica el tipus del radiobutton, el camp d'entrada de paraula i de la categoria gramatical.

En el fitxer `ajax.php` , es farà la funció `SELECT` amb condicions diferents depedent de la opció seleccionada i del valor introduït.

De manera que primer de tot, comprova amb `$_GET[]` que el valor introduït no es null, després mira si es tracta d'una cerca per lema o per forma i veure el radiobutton seleccionat, i finalment comprovar si tenim que fer la cerca depenent de la categoria gramatical.

El pseudocodi tindria el següent format:

```
if text != null:
    if lema:
        if tipus=radiobutton seleccionat:
            if categoria:
                SELECT atr1,atr... FROM ... WHERE lema=text( or lema
                REGEXP dependent del tipus) and categoria=tipus
            else:
                SELECT atr1,atr... FROM ... WHERE lema=text (or lema
                REGEXP
        else:
            difents tipus de radiobutton
            ..
            ..
    else if forma:
        if tipus=radiobutton seleccionat:
            if categoria:
                SELECT atr1,atr... FROM ... WHERE forma=text( or lema
                REGEXP dependent del tipus) and categoria=tipus

            else
                SELECT atr1,atr... FROM ... WHERE forma=text (or lema
                REGEXP
        else:
            difents tipus de radiobutton
            ..
            ..
else si text=null:
    if tipus:
        SELECT atr1,atr... FROM ... WHERE categoria=tipus
```

Si en la base de dades, es disposa d'un nombre significatiu de registres, fer consultes requeriria un gran esforç de la CPU i l'ús de memòria per poder executar-se.

Per exemple, si fem la següent cerca:

```
SELECT * FROM Paraules WHERE lema="hola"
```

i no existeix ningú ordre en el nostre dades, llegirà tots els registres de la taula Paraules i efectuarà una comparació entre la columna lema i la cadena "hola" i a mida que la mida de la base de dades incrementarà, això cada cop requerirà més temps per fer la consulta.

Per tal d'optimitzar el rendiment de les consultes, convendria crear indexos que ajuda a indexar el contingut de diverses columnes.

Per exemple, està creat un index pel registre lema de la taula Paraules, que executem la següent comanda:

```
ALTER TABLE Paraules ADD INDEX ('lema')
```

D'aquesta forma senzilla, es genera una llista ordenada de la columna lema de la taula Paraules. Que això d'alguna forma funciona com una guia de telefòn, que si tenim volem buscar una persona amb el cognom zapatero, ens aniríem directament al final de la guia.

Per poder notar la diferència considerablement, s'ha de comptar amb un nombre de registres significatius de la taula.

En el projecte, com que volem fer cerques de forma o lema de la paraula introduïda cream indexos per les columnes de lema i paraula de la taula Paraules.

Com que volem mostrar varies registres que están situats a taules diferents, utilitzem la tècnica de INNER JOIN, que permet seleccionar dades de diferents taules en funció de les funcions de join.

## 5.6 Inserció i modificació de dades

Per tal de guardar les dades en la base de dades, omplim el formulari de la pàgina d'afegir dades, on un cop tots els camps estàn omplerts, i clicant sobre el botó Submit, passem tots els valors introduïts com a paràmetre a la pàgina insertdata.php, mitjançant una funció AJAX, com que són diversos camps d'entrada, per no tenir que passar els valors un per un, s'ha fet servir la funció serialize de JQuery, que converteix d'una manera eficient els valors del formulari en una cadena de text.

I un cop recuperat els variables, utilitzem la funció `mysqli_real_escape_string()` per assegurar que la cadena SQL sigui legal escapant de caràcters especials.

I amb el mètode `INSERT`, podem insertar directament els valors a la taula "Paraules", "contexto" i "grup\_tonal" ja que la inserció es tracta d'un sol valor. Excepte per la taula de línies, com que per cada paraula, aquesta es troba dins d'un torn de parla on conté varies línies, per tant, no n'hi hauria prou amb només fer un sol `INSERT`.

Per tal d'afegir línies es va crear un responsive table on clicant sobre el botó add, es va afegint columnes a la taula i les línies están guardades en un array.

Per saber el nombre de línies que hi han, s'ha fet :

```
$number = count($_POST["name"]);
```

Per tant a l'hora de fer la inserció per les línies, s'hauria d'iterar sobre totes les línies introduïdes, per tal d'aconseguir això, s'ha fet un bucle for. De manera que si volem insertar 3 línies, es cridaria al mètode `INSERT` tres vegades.

Per a modificar les dades, bàsicament enlloc de `INSERT`, ara per actualitzar les dades, fem servir el `UPDATE`, que igual que abans, per les taules "Paraules", "contexto", i "grup\_tonal", podem fer l'actualització directament. Per exemple, per actualitzar la columna "paraula" fem servir la següent funció.

```
UPDATE Paraules SET paraula='$paraula' WHERE idParaules='$id'
```

El mateix problema sorgeix amb les línies, que ademés de modificar el text de les línies que es troben en la base de dades, podem afegir més línies o eliminar la línia existent.

Això provocaria un canvi en el nombre de la línia. Per exemple, si teníem 5 línies amb línia inicial=10, i eliminem la línia amb numlinia=13, el numlínia de les línies 14 i 15 canviaria. I el seu nou valor seria 13 i 14.

	idlinies	numlinia	liniainicial	linia
▶	1	2	2	sds
	1	3	2	lalla
	<del>2</del>	<del>12</del>	<del>12</del>	<del>aeadae</del>
	2	13	12	sdfds
	2	14	12	aaa



FIGURA 21: exemple canvi línia taula quan s'elimina una columna

O en el cas d'afegir més línies a la base de dades, tindríem que saber el nombre de línies existents per a poder assignar el valor de numlinia correcte a la nova línia. Per tant a l'hora d'actualitzar la taula de la línia s'hauria de tenir en compte totes aquestes condicions.

En cas de que el nombre de línies no canvia, doncs igual que abans, fem un bucle for i iterem sobre totes les línies que hi ha introduïdes, i fem el UPDATE per a cada línia.

I si hi ha un canvi en el nombre de línies, com que involucra canvis en la línia inicial, nombre de línies, i el text de la línia, he optat de fer primer eliminació de totes les línies amb el id corresponent:

```
DELETE FROM linies WHERE idlinies= '$id'.
```

I després iterar sobre les línies i fer una inserció.

## 5.7 Eliminació de dades

Com hem dit abans, seleccionem l'element que volem eliminar, un cop tenim el resultat mostrat, i clicant sobre el icone delete de la pàgina, aquest passa el id d'element seleccionat per tal de poder identificar l'element, i guardem aquest valor en una variable. Com que tenim 4 taules que están relacionades, i quan eliminem una paraula de la taula, tenim que eliminar també totes aquelles columnes com les línies del torn de parla en que es troba la paraula, el inicial del parlant, etc., S'ha fet 4 eliminacions utilitzant la declaració DELETE amb la clàusula WHERE per eliminar tots aquells registres que coincideix amb la condició especificada. Per exemple, per eliminar la taula de Paraula s'ha fet la següent funció:

```
DELETE FROM Paraules WHERE idParaules = '$id';
```

I després executem aquesta consulta passant per la `mysqli_query ()` funció PHP.

## 5.8 Validació

Abans d'enviar el formulari, es comprova que tots els camps estigui omplerts amb la funció `validate` del jQuery, ja que una columna buida en la taula no tindria sentit, i per tal de manipular això, dins de la funció javascript, amb el `submitHandler` processem les dades del formulari.

Dins de la funció `validate`, tots els camps del formulari es posa com a `required`. En cas de que algun dels camps están buits, en clicar el botó Submit, no es processarà la inserció de les dades, i apareixerà una línia en color vermell al baix d'aquelles camps d'inputs indicant l'absència de valors.

En quan fem la funció de sign up, es comprova primer els valors abans de fer l'INSERT.

En primer lloc, per a nom d'usuari, es comprova que el valor introduïts nòmes conté lletres mínuscules o majúscules. S'ha fet servir la funció de PHP `preg_match()`, que ens permet comparar cadenes amb expressions regulars. La sintaxi d'aquesta funció és:

```
entero preg_match (exp_regular , cadena [, coincidencias[, flags [, comienzo ]]])
```

En el nostre cas, `exp_regular` seria `"/^[a-zA-Z ]+$/"`, i la cadena a comparar el valor del camp d'input del nom.

Per validar el correu, s'ha utilitzat el filtre `FILTER_VALIDATE_EMAIL()`, que valida una adreça de correu automàticament.

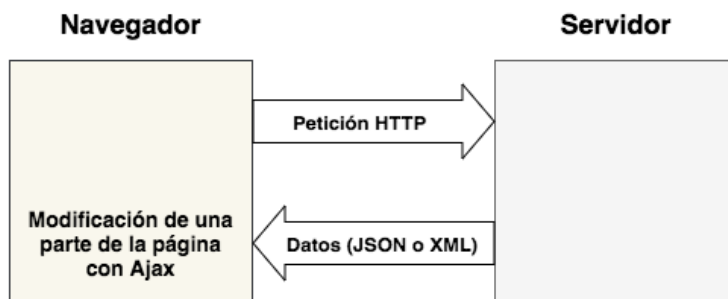
I finalment pel assegurar que el password tingui una seguretat mínima, amb la funció `strlen( $password)` es comprova que la longitud de string introduït té com a mínim 6 caràcters.

## 5.9 AJAX

Ajax és una tècnica de desenvolupament que permet actualitzar certes parts d'una pàgina web, millorant l'experiència i la interactivitat de l'usuari.

Permet només recarregar part de la pàgina, sense interrompre la navegació. On per exemple nosaltres per fer cerques diferents, això només modificarà la part de les taules sense alterar els altres elements de la pàgina.

La figura següent mostra el funcionament bàsic d'Ajax.



Vam a realitzar crides utilitzant JQuery, on per seleccionar entre els diferents opcions, utilitzem un objecte configurable, que serveix per configurar les necessitats del nostre petició AJAX.

On url indica l'adreça a la qual enviem la petició, que li posem el fitxer php on están els codis per processar les variables i interactuar amb la base de dades. Method es posa el tipus de la petició, que en aquest cas, com que volem obtenir dades del servidor, s'ha fet servir el mètode GET. Data són les dades que enviem al servidor per poder processar la cerca. I DataType, el tipus de data retornat, que seria JSON.

S'executa la funció success, quan s'obté amb èxit una resposta. Un cop rebut la resposta, es processa les dades retornades amb un bucle each, que recorre sobre tots els elements del array data obtingut com a resposta.

Insertem cada un dels elements del array a la taula creat anteriorment, amb el mètode append i fem visible la taula, que en un principi està posat amb display="none". Per a les accions d'enviar dades al servidor i guardarles, es fa ús del mètode POST d'Ajax.

### [5.10 Exportar dades a excel](#)

Per tal de exportar els resultats retornats i que están mostrats en una taula HTML a un fitxer, s'ha fet mitjançant el plugin de JQuery, table2excel, que permet exportar



taules HTML a fulls de càlcul Microsoft Excel, dades tabulars i text sense format. Per tant, cada cop que fem la cerca, apart de la taula de resultats, també es mostrarà tres botons per fer l'exportació.

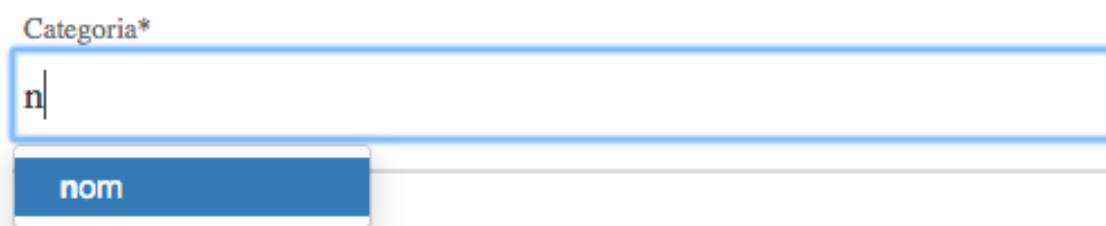
Désprés afegir tots els elements del array a la taula, clicant sobre el botó segons el tipus de fitxer que vols importar, això cridarà a la funció `table2excel` corresponent.

Dins de la funció, es poden passar propietats addicionals per personalitzar la característica de la taula. com el `exclude_links`, que elimina els enllaços.

### 5.11 Autocompletar

La tècnica d'autocompletar proporcionen a l'usuari suggeriments mentre escriu al camp, i seleccionar ràpidament el valor si aquest ja existeix en la base de dades.

Ho fem servir perquè a l'hora d'insertar paraules a la base de dades, podem trobar valors que es repeteixen amb alta freqüència, com per exemple la categoria de la paraula, o el corpus, i autocompletar ens permetria accerelar el procés d'escriptura, ja que si escrivim la lletra n en el camp de Categoria, si ja havíem fet la inserció amb categoria=nom, ens apareixerà l'opció i així no tindrim que escriure cada vegada la paraula completa.



**Figura 22:** exemple autocomplete del camp categoria

Per mostrar les opcions, s'ha importat el plugin `typeahead`, que farà una crida AJAX per omplir el desplegable.

En l'apartat del camp de link, com que aquest depèn de corpus, és a dir els enllaços són les mateixes per al mateix corpus, en seleccionar un corpus que ja existeix en la base de dades, el camp d'enllaç es completarà automàticament.

## 6. Proves i Resultats

Désprés d'acabar la fase d'implementació, es va dur a terme la prova de l'aplicació web per la professora de la facultat de filologia, col·laboradora del projecte.

Les proves principals han estat comprovar que les dades s'afegeixen, s'editen, s'eliminen correctament a o de la base de dades.

On a partir d'unes proves fetes, es va adonar-se d'alguns problemes existents de la web, algunes funcionalitats que no funcionaven de tot bé o que faltava coses com per exemple, poder recuperar la contrasenya en cas si ens l'oblidem. La qual cosa, a respota d'aquesta sugerència es va afegir una nova pàgina, on introduïnt el correu per fer el login, s'envia un email amb les informacions del compte d'usuari.

Creia que faltava un formulari per poder afegir símbols per poder observar coses com entonacions, les rialles..

Un cop arreglat els possibles fallos, i superat aquestes proves exitosament, es va proposar algunes millores tant funcionals com d'estil que es podria fer, com per exemple buscar una forma de poder duplicar la fitxa per canviar només el que és diferent.

## 6. Conclusions

Els objectius descrits en la part d'introducció han estat complerts gràcies als coneixements adquirits en les diferents assignatures del grau. Es va investigar les eines vitals per construir la web, i durant el transcurs del desenvolupament del projecte, s'ha anat investigant i dominant les diferents tecnologies i entorns.

Pel que fa a l'aplicació web i els objectius generals, el projecte ha estat enfocat en crear una pàgina web útil pels col·laboradors del grup GrEPAD per guardar els diferents arxius de corpus en una base de dades, i posteriorment, fer la consulta d'informació.

Un cop es tenia l'esquema del projecte, es va desenvolupar el disseny i l'implementació seguint els punts d'anàlisi fets.

S'ha perseguit la idea de crear una web amb un estil minimalista que permet a l'usuari aconseguir els objectius de forma més fàcil i eficient possible.

### Treballs futurs

El projecte funciona com una versió inicial amb funcionalitats que es podrien millorar per poder integrar aquesta aplicació en un entorn real i funcional.

Primer de tot, es podria perfeccionar el camp d'informació afegint les metadades, com l'època a que pertany, gèneres del discurs, etc..

Seria convenient si l'aplicació pot fer la lematització automàtica. Ja que ara l'aplicació funciona de la manera en que nosaltres tenim que introduir la forma i la lema corresponent manualment. La lematització comprèn la identificació i classificació tant de les variants morfològiques com de les ortogràfiques d'un mateix mot aparegut en un corpus textual. Per tant, trobar una forma de construir per exemple un diccionari per tal de guardar les informacions.

## 7. Bibliografia

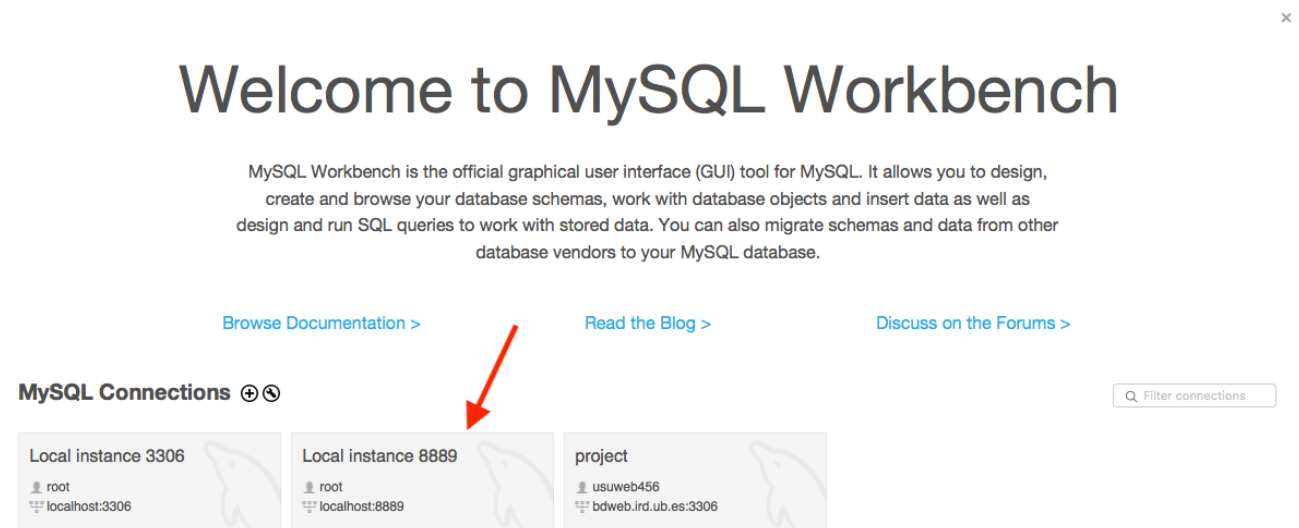
- [1] w3schools. <https://www.w3schools.com>
- [2] <https://www.tutorialrepublic.com>
- [3] <https://stackoverflow.com/>
- [4] <https://www.w3resource.com>
- [5] <https://www.anerbarrena.com>
- [6] Bootstrap <https://getbootstrap.com>
- [7] PHP Manual <https://php.net/manual/en/index.php>
- [8] Corpus de Català Contemporani de la Universitat de Barcelona  
<http://www.ub.edu/ccub/>
- [9] Dipòsit digital universitat de Barcelona  
<http://diposit.ub.edu/dspace/handle/2445/10410>

## 8. Annex. Manual d'usuari

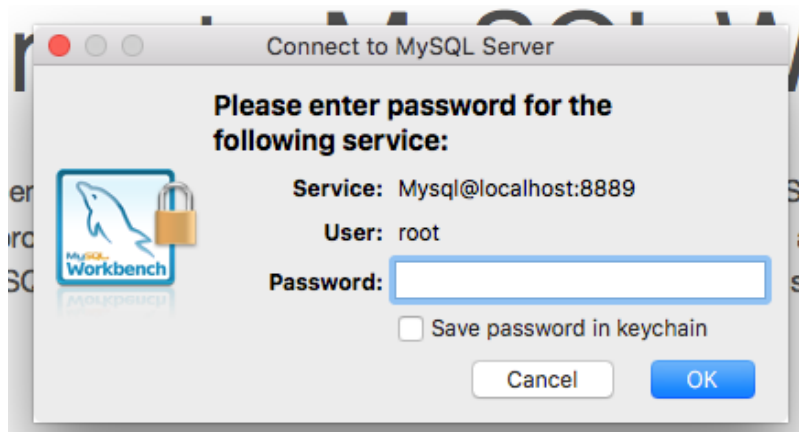
### 8.1 MySQL Workbench

Per tal de crear i connectar a la base de dades, es fàcil fer-lo a través del MySQL Workbench.

On per defecte ens connectem al servidor local amb port 8889



Un cop seleccionat el MySQL a connectar, posem com a contrasenya "root"



Si volem crear una nova connexió, cliquem sobre el botó add situat al costat de

## MySQL Connections. I omplim el següent

Setup New Connection

Connection Name:  Type a name for the connection

Connection Method:  Method to use to connect to the RDBMS

**Parameters** SSL Advanced

Hostname:  Port:  Name or IP address of the server host - and TCP/IP port.

Username:  Name of the user to connect with.

Password:    The user's password. Will be requested later if it's not set.

Default Schema:  The schema to use as default schema. Leave blank to select it later.

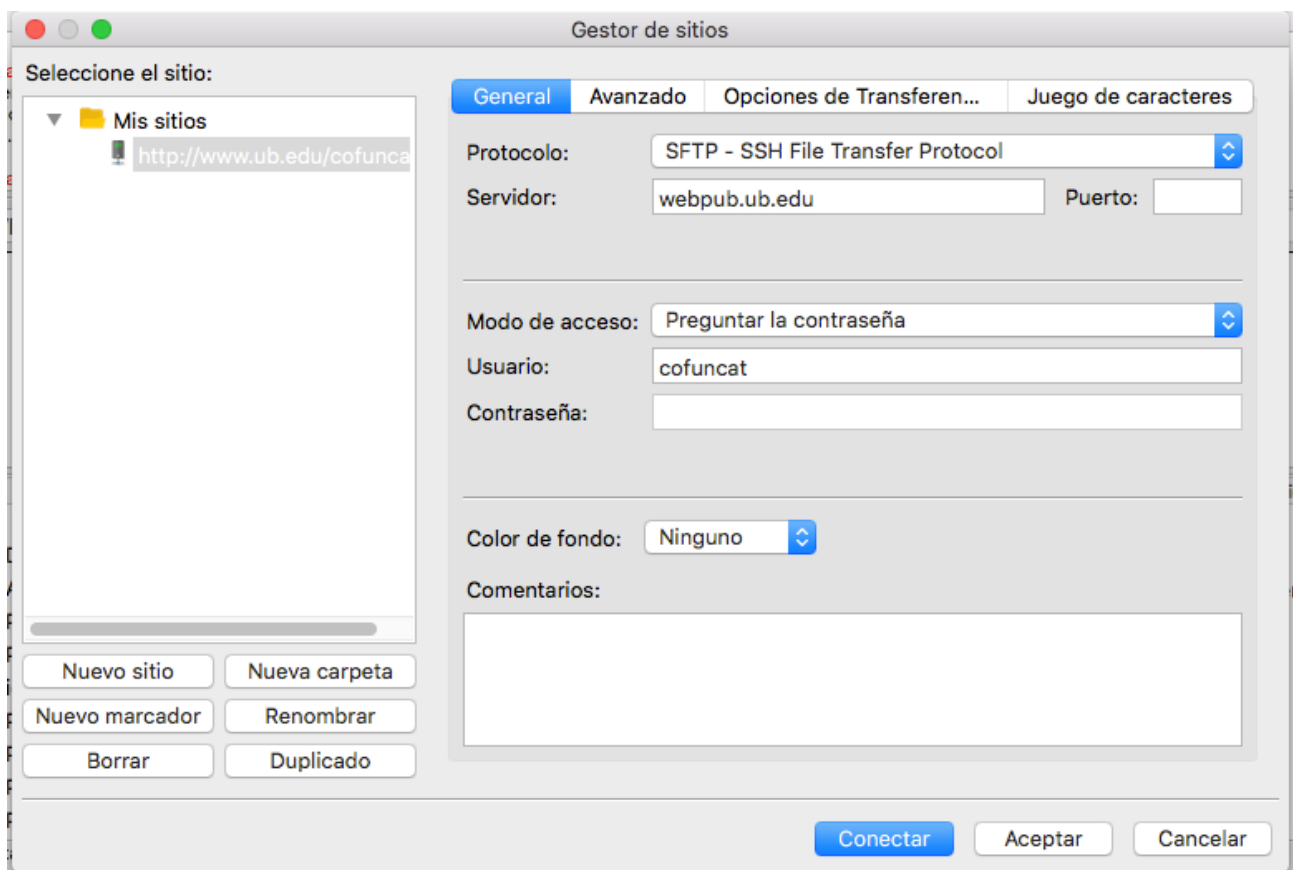
Tipus base dades: MySQL

- . Servidor: [bdweb.ird.ub.es](http://bdweb.ird.ub.es)
- . Nom de la base de dades : usuweb456
- . Nom de l'usuari administrador: usuweb456
- . Contrasenya: xxxxxxx

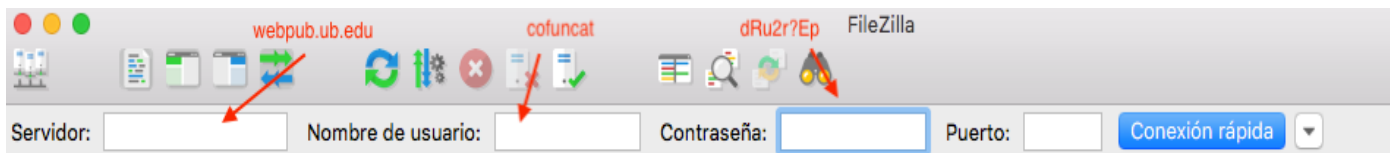
### 8.2 Connexió amb servidor

Utilitzant FileZilla és molt fàcil de configurar un cop sabem els paràmetres del lloc on tenim allotjada. El l'apartat "Archivo" -> "Gestor de sitios"  
Selecció com a protocol SFTP-SSH File Transfer Protocol.  
I omplim els diferents apartats amb les següents informacions

- . Servidor/host: [webpub.ub.edu](http://webpub.ub.edu)
- . Usuari: cofuncat
- . Contrasenya: xxxxxxx



O podem fer la connexió mitjançant la barra de ràpid accés



- Pas 1: Escriure el nostre domini
- Pas 2: El nostre nom de usuari assignat al host
- Pas 3: Escriu la seva contrasenya
- Pas 4: Prémer "Connexió ràpida"



Accedim a la pàgina web a través de la URL <http://www.ub.edu/cofuncat/>

Entrada a cercar

Buscar per categoria gr

Lema    Forma

- Coincident
- Començada per
- Acabada en
- En qualsevol posició
- No començada en
- No acabada en
- Que no contingui
- Símbol

I entrem a la pàgina principal

Ara per poder accedir a totes les funcionalitats, fem el log in amb l'usuari de tipus administrador. Clicant sobre el botó Login, apareixerà un formulari.

Login    Sign up

**Login**

Enter Email

Enter Password

Login

Close

[Forgot password?](#)

Si ens hem oblidat la contrasenya, i volem que ens enviïn un email per recuperar-la, fa clic sobre el link "Forgot password?" I es redirigeix a aquesta pàgina. On introduïm el email

Introdueix el correu per recuperar la contrasenya

Enviar

En cas de que volem fer la registració d'un usuari normal, premem el botó Sign up. I ens redirigeix a aquesta pàgina.

Per fer la cerca, introduir la paraula o conjunt de lletres en el camp "Entrada a cercar", i el segon camp és opcional, dependent de si vols restringir el resultat a un sol tipus de categoria gramatical.

Nom usuari

Correu

Contrasenya

Sign Up

I prémer el botó "Lema" o "Forma" per iniciar la cerca.

Exemple cerca amb paraula que comencen per lletra "h". El segon camp de text es pot deixar en blanc, cosa que fa una búsqueda de totes les categories.

Resultat retornat:

Resultat mostrat per un usuari no administrador

Paraula	Lema	Inicial Parlant	Categoria gramatical	Corpus	Text	Línia	Context	Enllaç
hola	hola	ecc	nom	cac	civ4	5	asdf	<θ>
						6	test	<θ>
						7	tesf	<θ>
						8	aaaa	<θ>
ho	ho	l05	pron	cap	C1CA05CC	1	1 l05: (... 1.42) (INH)	<θ>
						2	2 és que no ho sé_	<θ>
						3	3 p(e)rò::_	<θ>
						4	4 a ve(u)re\	<θ>
						5	5 et diré l'últim llibre\	<θ>
						6	6 {(P) que m'he [[legit]]}	<θ>

Pantalla inicial un cop logat. Es mostra el botó per afegir dades.

Afegir Dades

Entrada a cercar

Buscar per categoria gr

- Coincident
- Començada per
- Acabada en
- En qualsevol posició
- No començada en
- No acabada en
- Que no contingui
- Símbol

I si fem una prova de cerca ara, el resultat apareix 2 botons extras que amb un usuari logat no apareixia.

Paraula	Lema	Inicial Parlant	Categoria gramatical	Corpus	Text	Línia	Context	Enllaç
hola	hola	ecc	nom	cac	civ4	5	asdf	<a href="#">&lt;θ&gt;</a> <input type="button" value="✎"/> <input type="button" value="🗑️"/>
						6	test	<a href="#">&lt;θ&gt;</a> <input type="button" value="✎"/> <input type="button" value="🗑️"/>
						7	tesf	<a href="#">&lt;θ&gt;</a> <input type="button" value="✎"/> <input type="button" value="🗑️"/>
						8	aaaa	<a href="#">&lt;θ&gt;</a> <input type="button" value="✎"/> <input type="button" value="🗑️"/>

Pantalla d'afegir dades

Podem prémer la paraula CCCUB per retornar a la pàgina principal.

Redirigeix a la pàgina principal

# Afegir Dades

Buscar Dades

Forma \*

Introdueix la forma

Lema \*

Introdueix la lema

Categoria\*

Introdueix la categoria

Parlant \*

Introdueix el codi del parlant

Corpus \*

Introdueix el codi del corpus

Codi del text \*

Introdueix el codi del text

Enllaç \*

Introdueix l'enllaç al text

Torn de parla o paràgraf \*

Linia Inicial \*

Introdueix una frase entonativa o una clàusula

Afegeix

Submit →

En la part per insertar línies, podem afegir o borrar camps d'entrada amb els botons.

Torn de parla o paràgraf \*

Linia Inicial \*

Introdueix una frase entonativa o una clàusula

Afegeix

Introdueix una frase entonativa o una clàusula

X

Borrar línia

Introdueix una frase entonativa o una clàusula

X

Pantalla per modificar dades, bàsicament és igual que la d'afegir dades, a diferència de que els camps d'entrada mostren els valors



Go Back

## Modificar Dades

\* required field

Forma \*

hola

Lema \*

hola

Categoria\*

nom

Parlant \*

Parlant \*

ecc

Corpus \*

cac

Codi del text \*

civ4

Enllaç \*

www.google.es

Torn de para o paràgraf \*

Línia Inicial \* 2

Afegeix

sds

X

fdf

X