

Treball de Fi de Grau/Màster

## **Grau en Enginyeria en Tecnologies Industrials**

**Recomanador de matrícula en format web dirigit  
als estudiants de la UPC-ETSEIB, basat en el  
mètode estadístic de l'anàlisi del veí més pròxim**

### **MEMÒRIA**

**Autor:** Guillermo Amado Balaguer  
**Director:** Lluís Solano Albajes  
**Convocatòria:** Juny 2021



Escola Tècnica Superior  
d'Enginyeria Industrial de Barcelona





A l'Alèxia, per haver-me fet suport al llarg de tota la carrera i per incentivar-me a sempre donar el millor de mi.

Al Lluís, per brindar-me l'oportunitat de començar aquest projecte de nou i acceptar tutoritzar-me'l.





## RESUM I PARAULES CLAU

Aquest Treball de Fi de Grau ha estat creat amb la finalitat d'ajudar els actuals i futurs estudiants del Grau en Enginyeria en Tecnologies Industrials a la Universitat Politècnica de Catalunya a Barcelona (ETSEIB).

Més concretament, ha consistit en el desenvolupament d'una eina que els facilitarà el complex procés de matriculació d'assignatures durant la fase inicial del grau (primer i segon quadrimestre). Per tal d'aconseguir-ho, he intentat combinar una vessant teòrica amb una de pràctica.

En primer lloc, he fet un anàlisi i un tractament de les dades extens, combinant una base estadística amb el llenguatge Python i la seva llibreria Pandas. D'aquesta manera he aconseguit, no només estructurar i filtrar les dades necessàries per aquest projecte, sinó també elaborar un codi que em facilités la part pràctica (i, per tant, el producte final real) del treball.

A partir d'aquí, la part pràctica es recolza en la creació d'una pàgina web dinàmica, intuïtiva, senzilla i accessible. Per tal de fer-ho possible he combinat tres llenguatges de programació web: HTML5, CSS3 i JavaScript.

Finalment, he aconseguit crear un producte final que considero prou realista i que es podia ampliar molt més en un futur i arribar a ser de gran utilitat per als estudiants i docents de l'escola ETSEIB de la Universitat Politècnica de Catalunya.

**Paraules clau:** recomanador de matrícula, anàlisi de dades, Python, Pandas, HTML5, CSS3, JavaScript, fase inicial

## ABSTRACT AND KEY WORDS

This Final Degree Project has been created with the aim of helping current and future students of the Degree in Engineering in Industrial Technologies at the Polytechnic University of Catalunya in Barcelona (ETSEIB).

More specifically, it has consisted of the development of a tool that will facilitate the complex process of enrollment of subjects during the initial phase of the degree (first and second semester). In furtherance of that goal, I have tried to combine a theoretical slope with a practical one.

In the first place, I did extensive data analysis and processing, combining a statistical basis with the Python language and its Pandas library. By doing so, I have managed not only to structure and filter the data needed for this project, but also to develop a code that facilitates the practical part (and, therefore, the real final product) of the project.

From here, the practical part is supported by creating a dynamic, intuitive, simple, and accessible website. To make this possible, I have combined three web programming languages: HTML5, CSS3 and JavaScript.

Finally, I have managed to create a final product that I consider realistic enough and that could be expanded much more in the future and become very useful for students and teachers of the ETSEIB school of the Polytechnic University of Catalunya.

**Key words:** enrollment recommender, data analysis, Python, Pandas, HTML5, CSS3, JavaScript, initial phase

## ÍNDEX

ÍNDEX .....	5
1. GLOSSARI.....	9
2. PREFACI.....	11
2.1. Origen del treball .....	11
2.2. Motivació.....	11
2.3. Requeriments previs .....	12
3. INTORUDCCIÓ .....	13
3.1. Objectius del treball.....	13
3.2. Abast del treball.....	14
4. CONTEXT .....	15
5. PROBLEMES A RESOLDRE .....	17
6. BASE DE DADES ORIGINAL .....	18
7. APLICACIONS I REQUERIMENTS.....	20
7.1. Filtre.....	20
7.1.1. Filtre inicial .....	20
7.1.2. Alumnes amb assignatures suspeses.....	20
7.1.3. Codificació d'assignatures.....	21
7.2. Localitzar els veïns .....	21
7.3. Tractament de dades previ a anàlisi.....	22
7.4. Rendiment dels veïns.....	22

---

7.5.	Resultat .....	23
8.	EINES .....	24
9.	DESCRIPCIÓ DE LES EINES.....	25
9.1.	Llenguatges de programació.....	25
9.1.1.	Python .....	25
9.1.2.	HTML 5 .....	26
9.1.3.	CSS 3.....	26
9.1.4.	JavaScript .....	27
9.1.5.	Conjunt HTML + CSS + JavaScript.....	28
9.2.	Editors de text .....	28
9.2.1.	Jupyter Notebook.....	28
9.2.2.	Python y Python Shell.....	29
9.2.3.	Visual Studio Code .....	29
9.3.	Llibreries.....	30
9.3.1.	Pandas .....	30
9.3.1.	Flask .....	31
10.	IMPLEMENTACIÓ.....	34
10.1.	Funcions creades amb Python .....	34
10.1.1	Fixters <i>filtreQ1.py</i> i <i>filtreQ2.py</i> .....	34
10.1.1.1.	Funcions <i>filtrar_Q1 ( )</i> i <i>filtrar_Q2 ( )</i> .....	34
10.1.1.2.	Funcions <i>creafitxersQ2 ( )</i> i <i>crearfifxersQ1 ( )</i> .....	35

10.1.2.	Fitxer <i>programaQ1.py</i> .....	36
10.1.2.1.	Funció <i>veins(l)</i> .....	36
10.1.2.2.	Funció <i>analitza_veins ( l )</i> .....	36
10.1.2.3.	Funció <i>exit ( )</i> .....	38
10.1.2.4.	Funció <i>addicio ( l )</i> .....	39
10.1.2.5.	Funció <i>resposta ( l )</i> .....	40
10.1.3.	Fitxer <i>filtres_q1_q2_q3.py</i> .....	41
10.1.3.1.	Funció <i>mercat ( )</i> .....	41
10.1.4.	Fitxer <i>programaQ_altres.py</i> .....	42
10.1.4.1.	Funció <i>vecinos ( ls )</i> .....	42
10.1.4.2.	Funció <i>df_q3 ( )</i> .....	43
10.1.4.3.	Funció <i>analiza_vecinos (la, ls)</i> .....	43
10.1.4.4.	Funció <i>addicion (la, ls)</i> .....	44
10.1.4.5.	Funció <i>respuesta (la, ls)</i> .....	44
10.2.	Lloc Web .....	44
10.2.1	Pàgina principal .....	45
10.2.1.1.	Formulari per l'opció Q1 .....	45
10.2.1.2.	Formulari de Q1 i/o Q2 .....	46
10.2.2.	Pàgina amb la recomanació .....	48
10.2.3.	Aplicació de l'HTML .....	50
10.2.4.	Aplicació del CSS .....	51

---

10.2.5 Aplicació del JavaScript.....	52
11.    COSTOS, PLANIFICACIÓ I IMPACTE AMBIENTAL .....	55
11.1.    Costos.....	55
11.2.    Planificació.....	56
11.3.    Impacte Ambiental.....	57
12.    CONCLUSIONS.....	58
13.    BIBLIOGRAFIA.....	59

## 1. GLOSSARI

**ERP:** són les sigles de Enterprise Resource Planning. Concepte que representa l'automatització de l'empresa a partir d'un software, que s'encarrega de organitzar, tramitar i gestionar tots els tràmits o processos (com poden ser comandes, pagaments, stock, personal, etc) d'aquesta.

**Stackoverflow:** fòrum on es realitzen consultes relacionades amb temes de programació de tot tipus.

**Tag:** és la forma amb la que s'indiquen les etiquetes en el llenguatge HTML. Solen tenir una obertura <tag> i un tancament </tag>. Cadascuna de les etiquetes representa un element que s'introdueix en el codi de la teva pàgina web.

**If:** sentència que permet, per exemple en Python i JavaScript, imposar una condició en el codi.

**For:** sentència que permetrà iterar un element que emmagatzemi informació (llistes, diccionaris, DataFrames, etc).

**Bug:** en llenguatge informàtic, es tracta d'un error, incidència en el codi que produeix un problema en la màquina i no retorna el que s'espera de cada programa i/o funció.

**Debugging:** estratègia que es fa servir per identificar el bug o problema que existeix en el codi

**Open-source:** fa referència al software que ha estat implementat de per a que es pugui distribuir de forma gratuïta i pugui estar modificat

**Command Window / Anaconda Prompt:** és el terminal de l'ordinador, des d'on es pot accedir i organitzar els directoris de l'ordinador. També permet executar programes dins del directori convenient.

**DataFrame:** element essencial en el tractament de dades amb la llibreria Pandas. Esta compostat per un conjunt de columnes (cada columna representa un tipus de dada) i files enumerades per un índex.

**Array/llista:** tipus d'estructura que emmagatzema informació de diferents tipus de manera ordenada.

**Veïns:** estudiants que es troben en la mateixa situació, pel que fa a assignatures supeses i aprovades, que l'usuari.

**Checkbox:** tipus de input que s'especifica en el codi HTML. Disposa de dos estats, quan es fa clic sobre el boto el seu estat passa a ser seleccionat i, quan es torna a fer clic passa a estar

deseleccionat.

**Form:** és una etiqueta (<form></form>) HTML que indica la secció d'un document web compost per elements (botons, checkboxes, espai de text, etc) que permet recavar informació de l'usuari i enviar-la al servidor per un futur tractament.

**Script:** document que conté codi de programació.

**Diccionari de Python:** és una estructura que permetrà emmagatzemar informació de manera no ordenada. Aquesta informació ve donada per parelles claus-valors. Dins d'un diccionari les claus són úniques i es solen fer servir per accedir a la informació emmagatzemada.

## 2. PREFACI

### 2.1. Origen del treball

Gràcies a haver cursat Projecte I, vaig conèixer la gran utilitat les eines que es fan servir per tractar i analitzar correctament un conjunt extens de dades. Va ser el moment en el que em vaig donar compte del gran interès que em despertava el camp de la informàtica.

Aquest cas pràctic tractava de desenvolupar un recomanador de matrícula pels estudiants de la fase inicial de l'escola. Per tant, havent fet una primera presa de contacte amb el projecte, el plantejament que es proposa és situar-se altre cop en el punt de partida i implementar una nova estructura que permeti corregir, millorar i ampliar el projecte que més interès m'ha despertat com a estudiant.

### 2.2. Motivació

Gràcies a l'estructura del grau d'Enginyeria en Tecnologies Industrial, com a estudiants hem estat capaços d'adquirir coneixements i formar-nos en diversos sectors (electrònica, termodinàmica, química, informàtica, etc) i, per tant, poder avaluar quin dels camps estudiats s'ajusta més als nostres interessos o objectius professionals. En el meu cas, les assignatures que s'han cursat d'informàtica han estat les que més interès m'han despertat però, malauradament, són poques i es concentren majoritàriament al començament de la carrera. Per aquest motiu, sempre he aprofitat les oportunitats que m'ofereixen assignatures com Projecte I, Projecte II i altres del bloc optatiu per aprofundir més en aquest àmbit.

Addicionalment, he tingut la sort de realitzar aquest treball alhora que he cursat les pràctiques curriculars del grau en una consultoria tecnològica, on he anat a parar en un dels departaments que treballen amb el software SAP. SAP és un dels programes més utilitzats dins de la família d'eines ERP (*Enterprise Resource Planning*), que permet emmagatzemar un volum de dades immens per després tractar-les de la forma més convenient. Per tant, és un molt bon exemple del protagonisme que ha adquirit l'anàlisi o el tractament de dades en el món actual, i em demostra com d'aplicable es un projecte així en la vida quotidiana.

Poder ampliar la meva experiència en el tractament de dades i fer ús de les habilitats adquirides en programació al llarg de la carrera, ha fet que em decantés per desenvolupar una eina que pot ser de gran interès i utilitat de cara als estudiants que estan començant el grau i que òbviament, no tenen l'experiència dels alumnes més veterans per fer la millor tria possible en la matrícula del proper quadrimestre que cursaran.

### 2.3. Requeriments previs

Pel correcte desenvolupament del Treball de Fi de Grau, serà necessari tenir una sèrie de coneixements previs en diferents llenguatges informàtics com són, Python, HTML, CSS i JavaScript. Els coneixements Python seran els més rellevants, ja que serà el llenguatge amb el que tractaré les dades i per tant em serà essencial estar familiaritzat amb l'ús de diccionaris, llistes, condicionals o estructures iteratives. Pel que fa als demés llenguatges, és recomanable tenir-ne nocions, ja que facilitarà molt la creació de la pàgina web. Concretament faré servir HTML per crear l'estructura i l'organització de la web, CSS per estilitzar els components de la web, i per últim Javascript, per crear dinamisme i interaccions amb l'usuari.

## 3. INTORUDCCIÓ

### 3.1. Objectius del treball

L'objectiu final d'aquest treball és poder donar un suport a l'estudiant que comença la seva etapa universitària a l'ETSEIB, proporcionant-li una eina que permeti trobar dins de totes les possibilitats existents, una combinació d'assignatures que ofereixi el major èxit possible de cara a la matrícula del proper quadrimestre que cursarà.

El principal objectiu d'aquest Treball de Fi de Grau és analitzar el comportament dels estudiants del grau en Enginyeria en Tecnologies Industrials de l'Escola Tècnica Superior d'Enginyeria Industrial de Barcelona al passar la fase inicial, és a dir, en el pas del primer al segon curs.

Per tal de poder complir amb aquest objectiu, primerament s'han de complir un seguit d'altres objectius més petits però igualment importants:

- Aprofundir en els coneixements dels llenguatges esmentats prèviament.
- Estudiar els diferents processos que es poden fer servir amb Python per aconseguir el millor rendiment possible i disminuir al màxim el temps d'execució.
- Dominar la llibreria de *Pandas* per aconseguir tractar correctament un gran volum de dades de forma còmode i eficaç.
- Analitzar el comportament dels estudiants que s'han trobat en la mateixa situació que l'usuari.
- Determinar un criteri de puntuació de manera que es tingui en compte tants factors com sigui possible i, per tant, sigui el màxim realista.
- Familiaritzar-se amb el funcionament de Flask i descobrir què ens pot aportar.
- Fer recerca sobre la manera com es comuniquen els tres llenguatges usats en la creació de la pàgina web (HTML, CSS i JavaScript), per tal de proporcionar-hi accés als estudiants que en vulguin fer ús.
- La pàgina web esmentada ha de ser senzilla, intuïtiva i dinàmica per tal que l'usuari pugui obtenir la seva recomanació de la manera més fàcil i còmode possible. En addició, s'haurà d'adaptar a diferents mides i orientacions de pantalles.

### 3.2. Abast del treball

La proposta que s'ha realitzat per a dur a terme aquest treball, és bastant completa, ja que s'ha de pensar i planificar la manera com es tractaran les dades i tenir en compte els coneixements necessaris per a dur-ho a terme. Les dades tractades s'han de manipular de manera que s'obtingui el resultat desitjat i, per últim, preparar una interfície per que permeti a l'usuari accedir-hi.

A més, degut a que aquest treball es focalitza en ajudar els estudiants que estan cursant la fase inicial, es pot ampliar fàcilment per la cua, en un futur, i donar suport a estudiants que es troben en un punt més avançats del grau.

En base a l'abast d'aquesta primera proposta del projecte, s'han podria considerar la utilitat i viabilitat per futurs estudis, i fins i tot es podria extrapolar a altres sectors professionals.

## 4. CONTEXT

En un món dominat per la tecnologia digital, cada dia cobren més força conceptes com globalització, *big data* (bases de dades) i llenguatges informàtics.

En primer lloc, vull parlar de la globalització i la digitalització, ja que són fets que van molt lligats i als que cada vegada estem més exposats i condicionats. Actualment, gairebé tots els aspectes del nostre dia a dia se'n veuen cada vegada més influenciats i, òbviament, cal que ens hi anem adaptant ja que també provoquen canvis en les necessitats de les persones i, en general, canvis en el món que ens envolta.

Entenem com globalització (o mundialització) el procés pel qual els sistemes socials, naturals i tecnològics convergeixen fins el punt d'afectar el conjunt de la Terra, la societat i els individus de tot el món.

Aquest procés de globalització s'ha vist afavorit, en gran part, per l'exponencial desenvolupament tecnològic que ha tingut lloc els darrers anys. Podem dividir aquesta evolució en 3 onades de transformacions de la tecnologia de la informació<sup>1</sup>: la tercera és la que estem vivint en l'actualitat, i que es caracteritza per introduir la tecnologia de la informació com a part integral del propi producte, pel que comporta molta innovació, guanys de productivitat i creixement econòmic.

Per altra banda, la digitalització és el fet de convertir una magnitud física, un text o una senyal analògica en una representació digital<sup>2</sup>: registrar dades de manera digital.

En definitiva, les noves tecnologies ens donen l'oportunitat de saber què passa arreu del món, aprendre, investigar, i fins i tot crear negocis i vendre sense haver de sortir de casa.

Les bases de dades (o *big data*) són definides a l'Enciclopèdia Catalana com el conjunt total de dades de què pot disposar l'ordinador o els seus usuaris. També es poden definir com dades de gran mida, típicament el grau de manipulació de les quals presenta reptes lògics<sup>3</sup>. Per tant, és clar veure que es tracta d'un concepte molt variant alhora que subjectiu, cosa que converteix en un repte qualsevol tractament que se'n vulgui fer.

---

<sup>1</sup> Porter, M. i Heppelmann J. (2014). "How Smart, Connected Products Are Transforming Competition", *Harvard Business Review*

<sup>2</sup> Oxford Languages

<sup>3</sup> Oxford English Dictionary

Les dades es generen contínuament a partir de pràcticament tots els dispositius electrònics que avui en dia acompanyen a la societat. El món que ens envolta gira al voltant de la informació i l'anàlisi de la mateixa. Les grans empreses líders estan tan ben posicionades perquè, entre d'altres, utilitzen la immensitat de dades que generen, ja sigui per estudiar els seus potencials clients, com per vendre aquestes dades a altres empreses perquè les puguin analitzar i aplicar als seus respectius sectors. Així, és fàcil veure l'arma de doble fil que suposen: la dualitat que tenen com a eina de control, i alhora com a eina de gran ajuda per entendre millor com són els seus clients i què volen. Empreses com Google, Facebook i Apple són clars exemples d'aquest fenomen; han observat que generen un gran volum de dades i han sabut analitzar-les per treure'n profit. Si bé és cert que el *big data* afavoreix a les grans empreses, ja que són les que poden generar més volum de dades, també es pot aplicar en d'altres més petites i, inclús en entitats públiques, universitats, etc. Sense anar més lluny, al llarg del treball es demostrarà com l'ús de les bases de dades de les diferents matrícules dels estudiants de l'ETSEIB m'han fet possible desenvolupar aquest projecte. Així doncs, un dels principals objectius és buscar la optimització dels processos per tal de reduir els temps d'execució dels programes i de les funcions.

Finalment, crec necessari fer menció als llenguatges informàtics o de programació per introduir el meu treball. Definites com els mitjans utilitzats pels programadors per crear un programa d'ordinador<sup>4</sup>, s'han convertit en les noves llengües del món actual, i de gran importància per aquest Treball Fi de Grau que presento.

---

<sup>4</sup> [https://es.wikipedia.org/wiki/Lenguaje\\_informático](https://es.wikipedia.org/wiki/Lenguaje_informático)

## 5. PROBLEMES A RESOLDRE

El grau d'Enginyeria en Tecnologies Industrial esta dividit en dues parts: les assignatures que formen part de la fase inicial, i la resta d'assignatures que s'han de cursar fins el final de la carrera. Durant aquesta fase inicial és quan es produeix el famós primer filtre ja que, a l'estudiant, per a continuar en el grau se li exigeix, primerament, que al llarg d'un any acadèmic hagi superat un mínim de dotze crèdits i que, per altre banda, finalitzi aquesta fase inicial (que està composta per deu assignatures) en un termini màxim de dos anys.

Un dels majors problemes en els que es troben els estudiants és la manca d'experiència que tenen tant en el coneixement general de les assignatures com en el funcionament de la universitat, ja que és quelcom que els ve de nou. Per això, existeixen molts casos d'estudiants que, després d'haver realitzat un quadrimestre molt dolent, amb l'objectiu de remuntar el curs, es matriculen a un gran número d'assignatures. Els resultats no solen ser els esperats i es troben en una situació pitjor que la inicial.

És per això que, amb aquest treball, pretenc oferir una eina que pugui ajudar-los a fer una bona tria d'assignatures de cara al quadrimestre següent, de manera que, des d'un començament, tinguin la probabilitat més alta d'assolir el màxim número d'assignatures aprovades com sigui possible.

Per altra banda, el principal repte que suposa dur a terme aquest treball, recau sobre el tractament de les dades que s'han proporcionat des de l'escola, ja que inicialment s'hauran de realitzar diferents filtrats per quedar-nos únicament amb aquelles dades que realment interessin per dur a terme l'estudi desitjat.

## 6. BASE DE DADES ORIGINAL

Des del servidor de PRISMA de la UPC es reben les dades que es disposaran per a realitzar el projecte. Aquestes dades, però, contenen molta informació innecessària per a dur a terme l'anàlisi i, per tant, se'ls hi d'aplicar algun filtre per poder-ne reduir el volum i quedar-nos només amb les que m'interessen. S'ha de tenir en compte que, com major sigui el volum de dades, major serà el temps de resposta i, per tant, interessa poder reduir-les al màxim.

El primer fitxer que es traçarà serà “*qfni2019.csv*”, que presenta la següent estructura:

	CODI_PROGRAMA	CODI_EXPEDIENT	CODI_UPC_UD	CREDITS	CURS	QUAD	SUPERA	NOTA_PROF	NOTA_NUM_AVAL	NOTA_NUM_DEF	GRUP_CLASSE
0	752	284326	240013	6.0	2013	2	S	5.0	5.0	5.0	NaN
1	752	284326	240023	6.0	2013	2	S	3.0	4.5	5.0	NaN
2	752	276894	240015	6.0	2013	1	S	9.0	9.0	9.0	NaN

Figura 1. Estructura i exemple d'entrades del fitxer *qfni2019.csv*. Font: elaboració pròpia 2021

Cada entrada d'aquest fitxer ens dona la informació tant dels resultats obtinguts per un estudiant (*CODI\_EXPEDIENT*) cursant una assignatura de la fase inicial (*CODI\_UPC\_UD*), com del curs (*CURS*) i del quadrimestre (*QUAD*) en que l'ha cursat, entre d'altres.

Per l'estudi que vull dur a terme, les dades que em seran de essencials són les que he destacat en color vermell:

- El *CODI\_EXPEDIENT* servirà per poder identificar en tot moment els alumnes.
- El *CODI\_UPC\_UD* servirà per identificar quina assignatura ha cursat l'alumne.
- El *CURS* indica l'any en què l'alumne ha cursat aquella assignatura.
- El *QUAD* indica el quadrimestre en què l'alumne a cursat l'assignatura.
- *SUPERA* indica si l'estudiant ha superat (S) o no (N) l'assignatura.
- La *NOTA\_NUM\_DEF* fa referència a la nota que ha tret l'alumne a l'assignatura.

Es pot observar que, de totes les dades, n'hi ha una que està subratllada en groc (*CODI\_PROGRAMA*). Tot i que no es consideri de gran importància de cara a l'estudi, sí que ajudarà molt a l'hora de fer el filtrat de dades ja que, imposant que tingui un valor de 752, es descartaran totes les entrades dels estudiants que no pertanyin a l'ETSEIB.

Per altra banda, també es disposa del fitxer *qfnoini2019.csv* que conté exactament la mateixa informació que el prèviament descrit però, en aquest cas, dels alumnes que han cursat la fase no inicial. Per tant, la informació que ens serà rellevant serà la del primer fitxer.

	CODI_PROGRAMA	CODI_EXPEDIENT	CODI_UPC_UD	CREDITS	CURS	QUAD	SUPERA	NOTA_PROF	NOTA_NUM_AVAL	NOTA_NUM_DEF	GRUP_CLASSE
0	752	244846	240501	3.0	2013	1	S	10.0	10.0	10.0	10
1	752	245663	240501	3.0	2013	1	S	7.5	7.5	7.5	10
2	752	245505	240501	3.0	2013	1	S	8.5	8.5	8.5	10

Figura 2. Estructura i exemple d'entrades del fitxer qfnoini2019.csv. Font: elaboració pròpia 2021

Les dades d'aquest fitxer es poden fer servir per estudiar el pas dels alumnes del primer any al segon. Més concretament, el pas del quadrimestre dos al quadrimestre tres. Així doncs, entre d'altres filtres que s'hauran d'aplicar per poder tractar aquestes dades, un dels més rellevants serà quedar-se només amb les assignatures que pertanyin al tercer quadrimestre del grau.

## 7. APLICACIONS I REQUERIMENTS

Abans d'introduir el funcionament del programa en si, que s'ha desenvolupat per arribar al resultat final d'aquest treball, és important descriure correctament i clarament amb quines tasques s'ha anat dividint el projecte i quins objectius s'han anat complint amb elles.

### 7.1. Filtre

#### 7.1.1. Filtre inicial

Prèviament ja s'han introduït quines són les dades amb les que s'ha de treballar, i s'ha pogut comprovar que, efectivament, hi ha dades innecessàries que faran que el programa tardi més temps en executar-se i, per tant, disminueixi el rendiment. Per tant, aquest primer filtre simplement servirà per fer que, dels dos fitxers esmentats, n'apareguin només les dades que interessin per l'estudi.

#### 7.1.2. Alumnes amb assignatures suspeses

Serà necessari tenir agrupats als estudiants en funció de les assignatures que hagin suspès, per poder localitzar als estudiants que s'hagin trobat en la mateixa situació que l'usuari i analitzar el seu rendiment en el següent quadrimestre. Per això s'elaborarà un segon filtre que agruparà els alumnes en diferents documents en funció de les assignatures suspeses. En aquests documents cada alumne tindrà una única entrada on apareixeran les assignatures que ha suspès. En el cas, per exemple, d'un alumne que hagi suspès en el primer quadrimestre Càlcul I (240012), Àlgebra (240011) i Química I (240014), es trobarà emmagatzemat en el fitxer amb tots els estudiants amb tres assignatures suspeses i es veurà representat de la següent manera:

CODEX	S1	S2	S3
<b>Codi alumne</b>	240011	240012	240014

*Taula 1. Exemple entrada d'un alumne que ha suspès Àlgebra, Càlcul I i Química I, al fitxer de tres assignatures suspeses.  
Font: elaboració pròpia 2021*

En els casos on l'usuari acabi de cursar per primer cop el primer quadrimestre, es creen cinc fitxers diferents, des dels alumnes que han suspès una assignatura fins als alumnes que n'han suspès cinc. Això és degut a que, en aquesta situació, l'alumne pot cursar cinc assignatures com a màxim.

Per als casos on l'usuari acabi de cursar per primer cop el segon quadrimestre del grau, es creen fins a set documents diferents ja que, en aquest cas, els alumnes poden cursar més de cinc assignatures.

### 7.1.3. Codificació d'assignatures

Per fer més fàcil el tractament de dades es decideix donar una nomenclatura pròpia a cada assignatura, de manera que no s'hagi de treballar constantment amb els codis predeterminats.

Així, durant tot el procés, les assignatures passaran a tenir els següents valors:

Assignatura	Codi Assignatura	Codi propi
Àlgebra	240011	0
Càlcul I	240012	1
Mecànica Fonamental	240013	2
Química I	240014	3
Fonaments d'Informàtica	240015	4
Geometria	240021	5
Càlcul II	240022	6
Termodinàmica Fonamental	240023	7
Química II	240024	8
Expressió Gràfica	240025	9
Informàtica	240132	10
Mecànica	240133	11
Equacions Diferencials	240131	12
Materials	240033	13
Electromagnetisme	240031	14
Mètodes Numèrics	240032	15

Taula 2. Nomenclatura establerta per a cada assignatura. Font: elaboració pròpia 2021

## 7.2. Localitzar els veïns

El resultat del projecte es basa en analitzar el rendiment *a posteriori* de totes les persones que s'han trobat en la mateixa situació que l'usuari, és a dir, s'ha d'analitzar a quines assignatures s'han matriculat aquestes persones en la següent matrícula, i veure els resultats obtinguts. En funció d'aquests resultats es recomanarà una alternativa o una altra.

Per tant, és imprescindible poder localitzar a aquests estudiants, que anomenarem veïns. Per fer-ho es voldrà crear una funció que rebi com a entrada una llista amb les assignatures que l'usuari

ha suspès i que retorni el DataFrame de tots els estudiants que s'han trobat en la mateixa situació.

La funció esmentada prèviament es farà servir en el cas on l'estudiant acabi de cursar per primera vegada el primer quadrimestre. En cas contrari, la funció ha de complir exactament el mateix objectiu, però amb la diferència que ara es rebran dues entrades: una llista amb les assignatures que l'usuari ja ha aprovat, i una altra llista amb les assignatures que l'usuari acaba de suspendre. Aquest canvi és degut a que, per donar una recomanació realista i coherent, si no es descartessin les assignatures que l'usuari ja ha aprovat, el programa possiblement faria una recomanació inútil per l'usuari ja que li recomanaria que es matriculés a assignatures que ja hauria cursat i aprovat.

### 7.3. Tractament de dades previ a anàlisi

Amb el veïns ja localitzats, abans d'iniciar el procés d'anàlisi, caldrà tenir les dades perfectament preparades i estructurades. Amb aquest objectiu present, es voldrà crear una funció que, rebent el DataFrame dels veïns esmentat a l'anterior punt i el DataFrame *qfini2019.csv* amb el primer filtre aplicat, aconseguixi els següents objectius:

- Obtenir el següent quadrimestre dels veïns.
- Aconseguir tota la informació rellevant sobre el següent quadrimestre dels veïns (assignatures matriculades, assignatures suspeses, assignatures aprovades, notes, etc).
- Plasmar tota la informació obtinguda en un nou DataFrame.

Per tant, com a resposta, aquesta funció donarà un DataFrame que servirà per poder avaluar correctament el rendiment de cada veí en el següent quadrimestre.

### 7.4. Rendiment dels veïns

Com ja s'ha esmentat prèviament es vol que, a l'hora d'avaluar el rendiment de cada veí, es tinguin en compte el màxim número de factors per poder fer més realista l'estudi. Per això es voldrà que en aquest estudi es tingui en compte el següent:

- El tant per cent de les assignatures aprovades respecte les matriculades.
- La nota obtinguda en les assignatures.
- La dificultat de les assignatures.
- Quines assignatures s'han aprovat i quines assignatures s'han suspès.

Tenint present que el primer punt esmentat ja vindrà incorporat en el DataFrame obtingut en l'última apartat, per incorporar els següents factors pretenc crear dues noves funcions diferents:

1. La primera funció servirà simplement per obtenir el tant per cent d'èxit de cadascuna de las assignatures.
2. La segona funció servirà per relacionar la nota obtinguda amb l'assignatura cursada, és a dir, es plantejarà un sistema de bonificacions i penalitzacions. Per tenir en compte els factors restants alhora es vol aplicar una bonificació als estudiants per assignatura aprovada. Aquesta bonificació consistirà en sumar-los la nota treta a l'assignatura multiplicada pel tant per cent d'èxit de la pròpia assignatura. En cas contrari, es vol aplicar una penalització seguint el mateix procediment.

## 7.5. Resultat

Per finalitzar tot el procediment, només queda establir com es voldrà donar la recomanació. L'idea és crear una nova funció que, tenint com a entrada els resultats de l'anàlisi comentat prèviament, doni fins a 8 possibles matrícules en funció del número d'assignatures a matricular. És possible que, en els casos extrems, el programa es trobi que no té suficients dades per analitzar o que la recomanació que se li faria a l'usuari sigui inviable. En aquests casos, la resposta hauria de ser que, per aquell número d'assignatures, no es pot donar una recomanació fiable. En el cas on sí que es pugui donar, la funció hauria de retornar el bloc d'assignatures que ha obtingut millor puntuació, així com la puntuació en qüestió.

## 8. EINES

Al llarg d'aquest treball s'ha fet ús de diferents eines que han permès realitzar les diferents parts del projecte. L'objectiu d'aquest apartat és introduir les següents eines i programari utilitzat, tant en el desenvolupament del programa com en el desenvolupament de la pàgina web.

En primer lloc, esmentar els diferents editors de text que he fet servir:

- Jupyter notebook
- Visual code
- Editor de text propi i la Shell propis de Python

Per altra banda, pel que fa als llenguatges:

- Python 3
- HTML 5
- CSS 3
- JavaScript

Per últim, destacar les llibreries de Python utilitzades:

- Pandas
- Flask

## 9. DESCRIPCIÓ DE LES EINES

En aquest apartat es descriuran les eines esmentades de manera més detallada.

### 9.1. Llenguatges de programació

Pel que fa als llenguatges de programació, cadascun està destinat a cobrir una necessitat al llarg del projecte, per això és important conèixer tant la seva metodologia com l'ús pel qual han estat creats.

#### 9.1.1. Python

Python és el llenguatge per excel·lència que s'ensenya avui en dia a les diferents universitats per introduir l'estudiant al món de la programació. Això és degut a la seva fàcil comprensió, ja que es tracta d'un llenguatge molt intuïtiu i que reflexa molt bé la lògica humana. Per tant, llegir un codi escrit amb Python no deixa de ser com llegir un text escrit en anglès (l'idioma de referència, degut a que és l'idioma amb les que estan definides les funcions per defecte).



Figura 3. Python. Font: <https://cosasdedevs.com/python/>

Es caracteritza per ser un llenguatge pensat per poder comunicar-se amb d'altres, i per tant resulta molt útil tant per realitzar projectes a petita escala o per simplement introduir-se en la programació (com s'ha fet a l'escola), com per realitzar projectes de gran envergadura on es necessita l'ús de més d'un llenguatge.

La seva versatilitat és una de les raons per les quals ha passat per damunt d'altres llenguatges, ja que permet fer una gran varietat de funcions: des de desenvolupar la teva pàgina web fent ús, per exemple, de Flask, Django o JSON, realitzar desenvolupaments científics o de tractament de dades amb les llibreries SciPy o Pandas; fins a programar videojocs amb la llibreria Pygame (feta servir a l'optativa de Jocs per a Computadors. Estructura i Desenvolupament).

Python, com molts programes, és un codi obert, fet que fa que, darrera d'ell, existeixi una gran comunitat que permeti una evolució i expansió increïble. A més, el fet que existeixi una comunitat significa que hi ha molta documentació que l'usuari pot fer servir en qualsevol moment per resoldre dubtes o descobrir noves funcionalitats de les extenses llibreries de les que disposa Python.

En el projecte realitzat, s'ha fet servir l'última versió disponible, Python 3.9.5. Ens ha proporcionat les eines necessàries per dur a terme tot el tractament de dades necessari per poder complir amb l'objectiu de proporcionar una recomanació acurada i realista.

### 9.1.2. HTML 5



Figura 4. HTML 5. Font:  
<https://www.manejandodatos.es/2013/07/plantillas-de-html5/>

HTML és el llenguatge de referència per a qualsevol pàgina web. Aquest aporta la part més organitzativa a l'hora de dissenyar la web, ja que ens permet introduir els elements (imatges, enllaços, text, formularis, botons) que conformaran cadascuna de les seccions, ja sigui la barra de navegació i el cos, entre d'altres.

El llenguatge HTML funciona mitjançant etiquetes o *tags* que defineixen el tipus d'element que estem introduint. Aquestes etiquetes sempre s'han d'obrir i tancar mitjançant el següent sistema: amb '< etiqueta >', representem que hem obert l'etiqueta, i un cop ja s'ha definit tot el que es requereix, s'ha de tancar fent servir l'expressió '</ etiqueta >'. És molt usual definir una classe o un identificador als elements per poder tenir-los referenciats i accedir-hi fàcilment. Si es vol definir una classe, dins de l'etiqueta s'escriurà `class = "nom de la classe"`; és a dir '< etiqueta class = "nom de la classe" >'; en el cas que es vulgui definir un identificador dins de l'etiqueta s'escriurà `id = "nom del identificador"`, és a dir '< etiqueta id = "nom de l'identificador" >'.

És molt usual definir una classe o un identificador als elements per poder tenir-los referenciats i accedir-hi fàcilment. Si es vol definir una classe, dins de l'etiqueta s'escriurà `class = "nom de la classe"`; és a dir '< etiqueta class = "nom de la classe" >'; en el cas que es vulgui definir un identificador dins de l'etiqueta s'escriurà `id = "nom del identificador"`, és a dir '< etiqueta id = "nom de l'identificador" >'.

### 9.1.3. CSS 3

CSS són les inicials de *Cascading Style sheets*. Aquest llenguatge és el que es fa servir per a aportar els estils al codi HTML d'una pàgina web. Per tant, es farà servir per fer la pàgina web visualment atractiva aplicant, per exemple, diferents estils de lletra, fons, animacions i efectes al passar per sobre d'enllaços o botons amb el ratolí, i molt més. Es pot observar, doncs, que si és un llenguatge fet servir per estilitzar, requereix tenir un model i, per això, sempre anirà lligat a un document HTML al qual podrà accedir fent



Figura 5. CSS3. Font:  
<https://rolandocaldas.com/php/css3-basico-1-php-paso-a-paso>

menció de les diferents classes definides.

Així doncs, aquest llenguatge es basa en cridar elements definits en el fitxer HTML i aplicar-los els atributs desitjats en funció de l'objectiu que es vulgui complir. El format estàndard del llenguatge seria el següent:

- Primer de tot, s'ha de fer referència a un element del document HTML. Si a aquest element se l'hi ha definit una classe (pot existir més d'un element amb la mateixa classe) simplement cal escriure `.'nom_de_la_classe'`; en el cas que, per exemple, s'hagi definit un identificador (només existeix un element amb aquest identificador) en l'element, se'l cridarà escrivint `.'#nom_de_l'identificador'`.
- Tots els atributs que es vulguin implementar per a l'element, han d'estar definits dins d'una clau (`{ }`). Si, per exemple, es volgués aplicar una mida a la lletra de 16px a la classe `arròs`, es faria:

```
.arròs{  
  font-size: 16px;  
}
```

#### 9.1.4. JavaScript



Figura 6. JavaScript. Font: [https://dev.to/azkar\\_moulana/javascript-what-happens-with-the-browser-3cdn](https://dev.to/azkar_moulana/javascript-what-happens-with-the-browser-3cdn)

Fins ara s'ha vist que existeixen dos llenguatges diferents per organitzar i per estilitzar una pàgina web, però ens falta un llenguatge que permeti crear cert dinamisme i fer que l'usuari pugui interactuar. És aquí on entra en joc el llenguatge de JavaScript, ja que ens

permet definir tot tipus d'esdeveniments i funcions que s'activaran quan l'usuari faci clic sobre un botó o premi una tecla.

Aquest llenguatge, altre cop, va molt lligat a l'HTML ja que, molts cops, el que es buscarà serà modificar parts del codi HTML en situacions determinades, per exemple, substituir una imatge o un text per un altre. També és molt comú fer que el codi d'un fitxer de JavaScript interaccioni amb el codi d'un document CSS ja que, per exemple, es podrà fer amagar o ensenyar una part de la web en funció del botó que es premi.

### 9.1.5. Conjunt HTML + CSS + JavaScript

Com s'ha pogut comprovar, per poder dissenyar un lloc web complet, agradable i atractiu, és necessari fer servir tots tres llenguatges, tot i tenir els tres una funció principal estipulada.



Figura 7. Combinació HTML5 + CSS3 + JavaScript. Font: <https://teorema-rd.com/producto/workshop-javascript-html-and-css-web-development>

## 9.2. Editors de text

Els editors de text són els programes que ens permeten, com bé diu el seu nom, poder escriure el codi del programa que s'estigui desenvolupant. El gran avantatge que tenen és que, generalment, amb un únic editor de text pots escriure qualsevol llenguatge i no has de fer servir diferents programes. A més, degut a que generalment es poden instal·lar diferents complements, disposen d'eines com el text predictiu o com la diferenciació dels events/funcions del codi.

Existeixen nombrosos editors de text i cadascun d'ells ofereix prestacions similars, però cadascú acaba decantant-se per aquell que funcionalment li resulta més còmode i fàcil de fer servir.

### 9.2.1. Jupyter Notebook

Es tracta d'una aplicació de codi obert online que ens permet desenvolupar codi en diferents llenguatges. En aquest cas s'ha fet únicament per l'ús de Python. Un dels motius principals pels que s'ha fet servir aquesta aplicació és que permet experimentar molt fàcilment amb petites parts del codi i, per tant, és útil a l'hora d'analitzar una part en concret que està donant errors.

Per altra banda, degut a que es pretén tractar DataFrames de gran



Figura 8. Jupyter Notebook. Font: [https://es.wikipedia.org/wiki/Proyecto\\_Jupyter](https://es.wikipedia.org/wiki/Proyecto_Jupyter)

dimensions, és important que el programador pugui visualitzar-los de manera clara per poder entendre com estan estructurades. Aquesta aplicació permet justament això: obtenir les respostes desitjades d'una manera molt visual i entenedora, com es podrà comprovar al llarg de la memòria.

### 9.2.2. Python y Python Shell

Python té el seu propi editor de text, el qual s'ha fet servir per desenvolupar la major part del codi del projecte referent a l'estudi de las dades. L'avantatge de treballar directament amb aquest editor de text és l'ús de la Shell de Python, on es poden executar directament les funcions creades en el fitxer. També serveix per a realitzar tot tipus de proves que permeten entendre molt millor el llenguatge en si. A part, és molt útil per buscar solucions als errors que puguin donar-se i per plantejar nous procediments que es volen aplicar al codi.



Figura 9. IDLE. Font: <https://morioh.com/p/3b2d2eb23f3>

### 9.2.3. Visual Studio Code

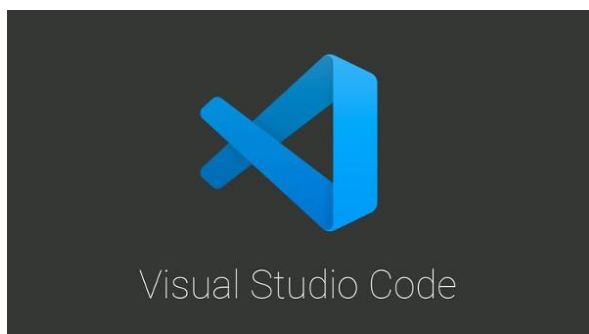


Figura 10. Visual Studio Code. Font: <https://www.ochobitshacenunbyte.com/2020/05/21/como-instalar-visual-studio-code-en-linux/>

Es tracta d'un dels editors de text més complet que es pot fer servir actualment. Per aquest projecte s'ha fet servir per desenvolupar tot el codi relacionat amb la pàgina web, és a dir, l'HTML, el CSS i el JavaScript s'han dut a terme fent servir aquesta aplicació.

És important tenir present el llenguatge que has de fer servir a l'hora de triar un editor de text, i en aquest cas, ha estat justament per això que he

escollit fer servir Visual Studio Code. De totes les característiques que el conformen, destacaria:

- **La gran quantitat d'extensions:** Funcionalment, tots els editors de text són iguals. El que els acaba diferenciant són les eines que es puguin afegir a més a més. Les extensions més destacades són:
  - *IntelliSense:* Es tracta d'una eina que autocompleta de forma intel·ligent, tenint en compte les funcions i els paràmetres ja definits, mentre escrius el codi.
  - *Prettier:* Serveix per mantenir una estructura coherent del codi. Un cop es guarda el document, aplica un estil d'organització coherent amb el codi escrit.

- *Bracket Pair Colorizer 2*: En llenguatges com HTML, CSS i, sobretot, JavaScript, és molt important tenir present l'obertura i tancaments d'etiquetes, parèntesis, claus o claudàtors. Fent servir aquesta eina es dona color a aquests elements per poder identificar ràpidament l'inici i el final dels diferents elements.
- **Debuguejar**: La forma més habitual que es fa servir per trobar errors en el codi és fer imprimir valors a la consola i analitzar-los. Amb Visual Studio Code, per defecte, es té l'opció d'aplicar *break points* o *call stacks* i, alhora, disposa d'una pròpia consola.
- **Identificar errors**: Analitza el codi per defecte, remarca en vermell aquelles parts del codi que no estan ben implementades i dona a la consola el motiu d'aquest error.

### 9.3. Llibreries

Per poder millorar el rendiment del codi que es crea, l'ús de llibreries és essencial. Un llibreria permet aplicar una funció en el teu codi fent servir una única sentència, en lloc d'haver primer de definir-la. És a dir, es tracta d'un codi extern que està preparat per aplicar funcions que facilitarà la creació del codi i també, li millorarà el rendiment.

La comanda bàsica per poder fer ús d'una llibreria és *import*. Aquesta comanda, permet importar qualsevol llibreria en el nostre codi, de manera que ens puguem beneficiar de les funcions definides en ella. Si es volgués, també es podria importar únicament una funció d'una llibreria fent, *from llibreria import function*.

#### 9.3.1. Pandas



Figura 11. Llibreria Pandas. Font: <https://www.adictosaltrabajo.com/2020/12/29/primeros-pasos-con-pandas/>

Es tracta d'una llibreria de Python que permet fer un tractament de dades molt eficaç en quant a temps d'execució, i evita al programador haver de desenvolupar línies de codi per llegir les dades mitjançant iteració.

L'objecte per defecte que es fa servir amb Pandas són els DataFrames. Això equivaldria a una taula indexada organitzada en files i columnes i que, per tant, serà imprescindible tenir un gran domini sobre aquests objectes. Els DataFrames emmagatzemen les dades de manera que el programador sempre tingui accés a elles mitjançant la crida de qualsevol de les sentències definides en la llibreria.

Tots els fitxer proporcionats que s'han descrit prèviament es llegeixen mitjançant una funció de Pandas que permet convertir diferents tipus de documents (en aquest cas `.csv`) en DataFrames amb la sentència `pd.read_csv('document.csv')`. Com es pot observar, s'ha utilitzat l'expressió `pd` per referir-se a la llibreria Pandas. Això és degut que, com a criteri general, s'ha establert que a l'hora d'importar la llibreria s'importi amb el nom de `pd`.

### 9.3.1. Flask

Flask és la llibreria de Python que ens permetrà poder connectar el lloc web amb el servidor. Es pot entendre doncs, que serà l'eina que farà d'intermediària entre l'HTML i Python, de manera que puguem rebre la informació de l'usuari, tractar-la amb els fitxer elaborats i, posteriorment, plasmar els resultats al lloc web.



Figura 12. Llibreria Flask. Font: <https://pythondiario.com/2018/08/construccion-de-una-api-de-prediccion.html>

Flask, es caracteritza per necessitar un tipus d'organització i estructura especial pel que fa als directoris. Tot ha d'estar contingut dins d'una carpeta que anomenarem carpeta mare (senyalada amb vermell a la Figura 3):

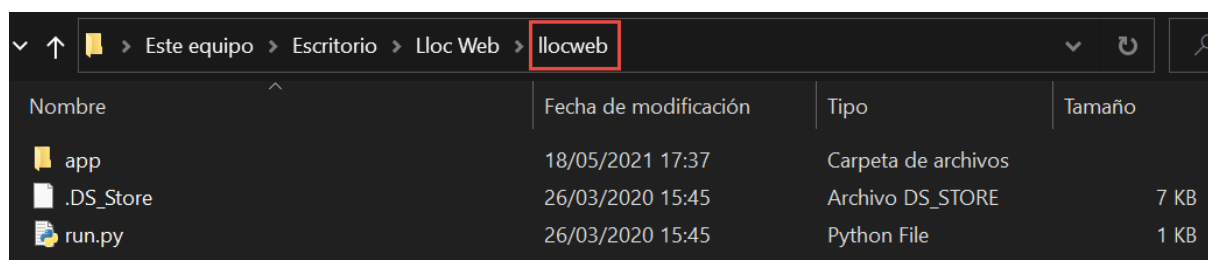


Figura 13. Carpeta mare. Font: elaboració pròpia 2021

Dins d'aquesta carpeta mare (`llocweb`), es destaquen:

- Carpeta `app`: En aquesta carpeta es troben tots els continguts que conformen la pàgina web. Des de els scripts de Python fins a les imatges que s'han de carregar al codi CSS.
- Fitxer `run.py`: Executant aquest script en el terminal o Command Window, és la manera

de poder accedir al lloc web i, alhora, a totes les funcions que es troben dins dels scripts a la carpeta *app*.

En la carpeta *app*, s'hauran de distribuir tots els arxius en subcarpetes depenent de la seva funcionalitat:

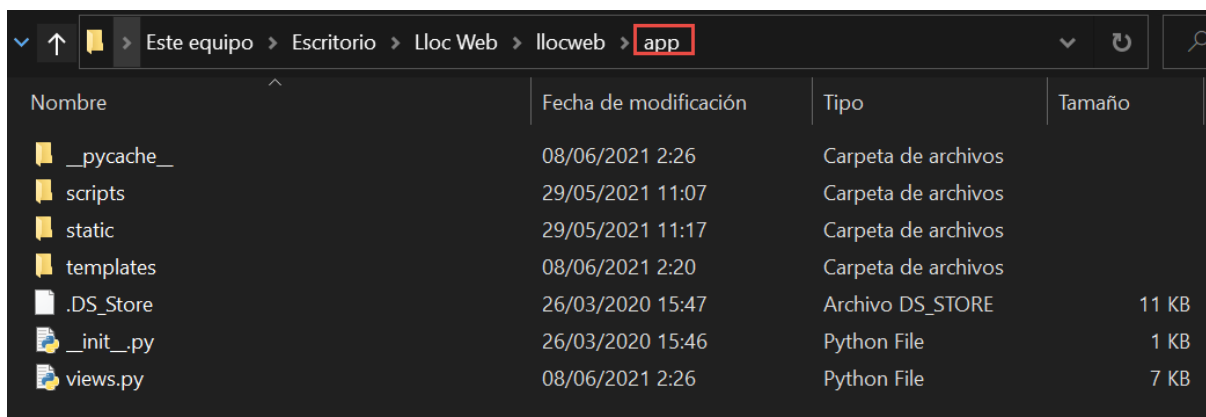


Figura 14. Distribució de la carpeta *app*. Font: elaboració pròpia 2021

- Carpeta *scripts*: En aquesta carpeta s'ubicaran tots els scripts creats amb Python que són necessaris per dur a terme l'estudi. En el nostre cas hi trobarem dos:

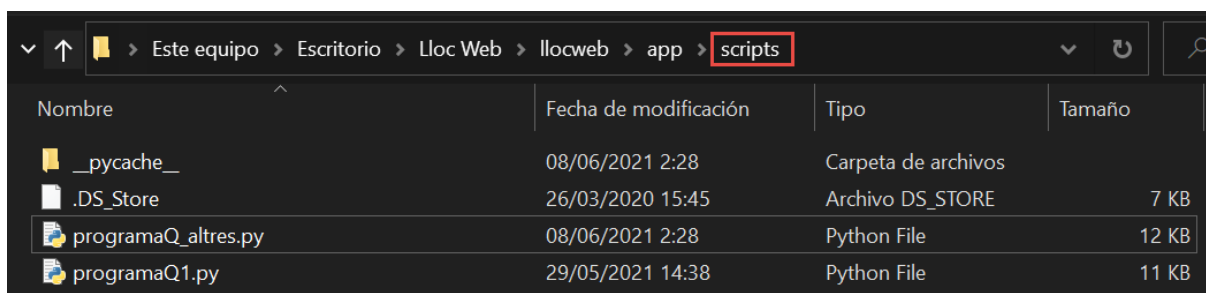


Figura 15. Carpeta *scripts*. Font: elaboració pròpia 2021

- Carpeta *statics*: En aquest directori es posaran totes les imatges, documents, etc, que es facin servir en algun dels fitxers Python o HTML.

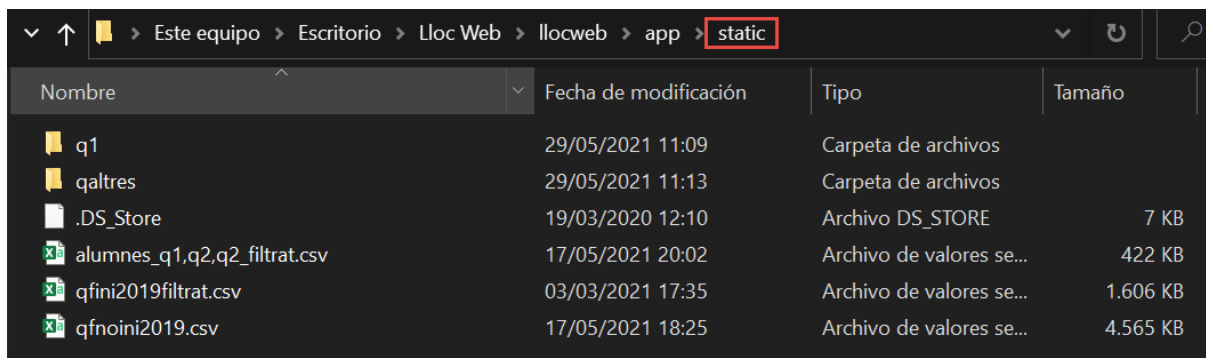


Figura 16. Carpeta *static*. Font: elaboració pròpia 2021

- Carpeta *templates*: És on s'ubicaran tots els fitxers relacionats amb la pàgina web. Tant el codi CSS com el JavaScript, es troben dins del document HTML.

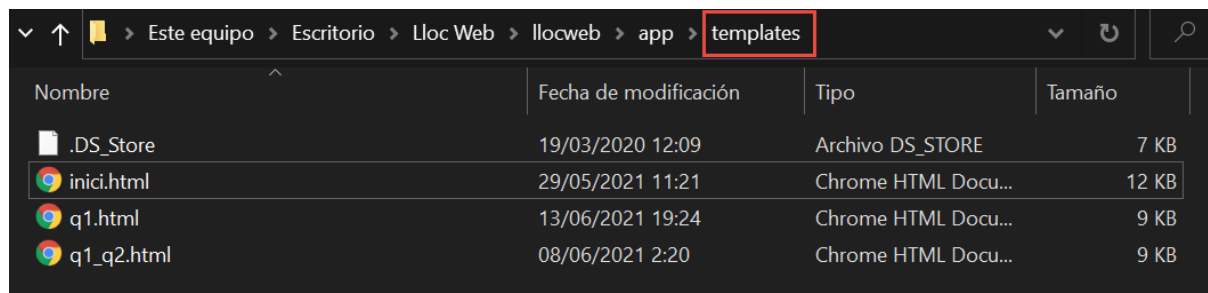


Figura 17. Carpeta *templates*. Font: elaboració pròpia 2021

- Fitxer `__init__.py`: Fitxer que es executa pel `run.py`. Té com a objectius la importació del fitxer `views.py` i iniciar l'execució del lloc web.
- Fitxer `views.py`: És el fitxer on es rebrà tota la informació de la web i, per tant, és on s'executaran les funcions dels scripts creats. Les respostes obtingudes, es podran enviar als fitxers HTML per a que es pugin tractar i ensenyar-se de la forma desitjada. A més, des d'aquest fitxer es definirà la manera de navegar entre les pàgines de la web.

## 10. IMPLEMENTACIÓ

L'objectiu d'aquest apartat és explicar com s'han cobert totes les necessitats explicades en l'apartat 8 a partir de funcions creades amb Python. Tot i que els fitxers de codi es troben adjuntats, és important explicar com s'han definit cadascuna de les funcions ja que, a vegades, entendre el codi escrit per una altra persona pot ser complicat. A més, també s'explicarà com s'ha desenvolupat el lloc web.

### 10.1. Funcions creades amb Python

Com ja s'ha explicat, Python és l'eina fonamental que s'ha fet servir per dur a terme tot l'estudi. En aquest subapartat s'analitzaran els diferents fitxers que s'han creat i les funcions que contenen.

#### 10.1.1 Fitxers *filtreQ1.py* i *filtreQ2.py*

Aquests fitxers contenen les funcions necessàries per aconseguir els documents *.csv* (en el cas de *filtreQ1.py* retornarà cinc documents *.csv*, i en el cas de *filtreQ2.py* en retornarà set) que serviran per ubicar a tots els estudiants que s'han trobat en la mateixa situació que l'usuari. Aquests documents, posteriorment, els llegirà el fitxer principal de Python per iniciar el tractament de les dades. Per tant, no es trobaran dins de la carpeta *Scripts* ja que únicament els necessitem executar un cop per aconseguir els documents. Dins de cadascun d'aquests fitxers es troben dues funcions.

##### 10.1.1.1. Funcions *filtrar\_Q1()* i *filtrar\_Q2()*

Aquestes funcions són les encarregades de llegir el document *qfni2019filtrat.csv* fent servir la comanda *pd.read\_csv(nom\_document.csv)*<sup>5</sup> i, d'aplicar-hi dos filtres:

- En el cas de *filtrar\_Q1()*, filtra el DataFrame per la columna que representa el número del quadrimestre i per la columna que fa referència als codis d'assignatura. Es quedarà, doncs, amb totes aquelles entrades que tinguin com a valor el quadrimestre 1 i que continguin alguna de les assignatures cursades en el primer quadrimestre.
- Pel que fa a *filtrar\_Q2()*, únicament filtrarà el quadrimestre ja que, els alumnes, en el segon quadrimestre, poden haver cursat el total d'assignatures que es troben a la fase inicial. Per tant, simplement es descartaran totes aquelles entrades que tinguin com a valor 1 en la columna de quadrimestre.

---

<sup>5</sup> Converteix els documents *.csv* en un Dataframe.

### 10.1.1.2. Funcions *creafitxersQ2* () i *crearfifxersQ1* ()

Aquestes funcions cridaran a les anteriors per poder treballar amb el DataFrame, obtingut prèviament. Té dos principals objectius: realitzar el canvi de nomenclatura, per als codis d'assignatura; i crear i omplir nous DataFrames que contindran als alumnes en funció del numero d'assignatures que hagin suspès.

El funcionament és exactament igual per a les dues funcions i, per tant, s'explicarà únicament de *crearfifxersQ2*. Primerament, iterant el DataFrame es crearà un diccionari que tindrà com a clau el 'CODEX' de l'estudiant i, com a valors, les diferents assignatures que l'estudiant hagi suspès. En la Figura 3 es pot veure un fragment del diccionari esmentat.

```
Out[5]: {228721: [1, 1, 1, 1, 1],
         228048: [1, 1, 1, 0, 1],
         232335: [0, 1, 1, 0, 0],
         228536: [1, 1, 0, 1, 0],
         227742: [0, 0, 1, 0, 1],
```

Figura 18. Diccionari amb els CODEX i assignatures suspeses. Font: elaboració pròpia 2021

Un cop creat aquest diccionari, es crearan cinc nous DataFrames buits, on, cadascun d'ells, farà referència als estudiants que hagin suspès una, dues, tres, quatre o cinc assignatures, respectivament. Tots tindran en la primera columna el 'CODEX' de l'estudiant i després, tantes columnes depenent del DataFrame que es tracti,

Per omplir-los, es recorrerà el diccionari creat i, tenint en compte la llargada de cada un dels valors d'aquest (la llargada ens indicarà el número d'assignatures que ha suspès un estudiant), s'aniran afegint entrades fent servir l'atribut *df.at[index, columna]*<sup>6</sup> al DataFrame corresponent.

	CODEX	S1	S2	S3
0	228536	0	1	3
1	228532	2	3	4
2	228600	1	3	4
3	228715	0	1	4
4	227406	0	2	3
...	...	...	...	...

Figura 19. Entrades del document d'estudiants amb tres assignatures suspeses. Font: elaboració pròpia 2021

Un cop ja s'hagin omplert tots, fent servir la comanda *df.to\_csv('nom del fitxer.csv')*<sup>7</sup> es crearan a la carpeta de treball els cinc documents esmentats. A la Figura 4 es pot veure la informació que

<sup>6</sup> Accedir a un valor d'una fila indicant l'*index* i la *columna* on es troba.

<sup>7</sup> És la comanda inversa a *pd.read\_csv()*, converteix el DataFrame *df* en un document *.csv* amb el nom del fitxer que s'imposi.

es trobaria en el document dels estudiants que han suspès tres assignatures.

Per últim, destacar que per la funció *crearfitxersq2* (), enlloc de cinc fitxers se'n crearan set ja que, en el segon quadrimestre, els alumnes poden cursar més de cinc assignatures i, per tant, també en poden suspendre més.

### 10.1.2. Fitxer *programaQ1.py*

En aquest fitxer és on es troben totes les funcions que duen a terme l'estudi per als usuaris que acabin de cursar per primer cop el primer quadrimestre.

#### 10.1.2.1. Funció *veins(l)*

El primer que s'ha de fer és trobar els veïns més pròxims de l'usuari, és a dir, els alumnes que hagin suspès exactament les mateixes assignatures que ell. Per fer-ho, es necessari conèixer les assignatures que ha suspès l'usuari i per tant, com a paràmetre d'entrada, requerirà una llista amb les matèries suspeses (introduïdes mitjançant el lloc web).

En funció de la llargada d'aquesta llista i, per tant, de les assignatures suspeses, és llegirà un dels documents *.csv* creats amb la funció *crearfitxersq1* (). Per assegurar-nos que només apareixeran les matèries suspeses, s'imposarà que només apareguin les entrades que continguin els mateixos codis d'assignatura que els de la llista d'entrada.

Així doncs, com es pot apreciar a la Figura 5, aquesta funció retornarà un DataFrame que contingui tots els veïns de l'usuari.

#### 10.1.2.2. Funció *analitza\_veins (l)*

Es tracta, segurament, de la funció més important de l'estudi, ja que aconseguirem obtenir un DataFrame que reculli tota la informació necessària per poder, finalment, escollir la millor combinació d'assignatures.

El següent pas, és analitzar què és el que han fet els veïns en el següent quadrimestre. Per tant, s'hauran de creuar el DataFrame que conté tota la informació dels estudiants que han cursat la fase

```
In [3]: l=[1,2,4]
        veins(l)
```

```
Out[3]:
```

	CODEX	S1	S2	S3
0	229390	1	2	4
1	229262	1	2	4
2	227213	1	2	4
3	232772	1	2	4
4	245509	1	2	4
...	...	...	...	...

Figura 20. Veïns de l'usuari que ha suspès Càlcul I, Mecànica Fonamental i Fonaments d'Informàtica. Font: elaboració pròpia 2021

inicial (del qual només ens quedarem amb les dades que representin assignatures cursades en el segon quadrimestre), amb el que conté tots els veïns de l'estudiant.

Cal tenir molt present que només ens interessa el quadrimestre cursat immediatament després d'haver finalitzat per primer cop el primer quadrimestre. Per això s'hauran d'eliminar totes les entrades que facin referència a altres anys, és a dir, si un alumne cursa per primer cop el Q1 en el curs 2014 (en les dades el curs escolar 2014-2015 es reflecteix com a 2014), les entrades que eliminarem seran totes aquelles que representin assignatures cursades en el quadrimestre de primavera de qualsevol altre curs.

Tenint en compte que, per eliminar aquestes entrades, s'haurà d'iterar el DataFrame creuat (*dfq2*), amb l'objectiu de millorar el temps d'execució del programa, s'aprofitarà la iteració per crear una llista de llistes que contingui els valors de les columnes 'CODEX', 'CURS', 'CODASS', 'APR' i 'NF'. Amb aquesta llista es crearan cinc diccionaris on tots tindran el 'CODEX' com a clau i es diferenciaran segons el valor de cada diccionari. Per tant, aconseguirem tenir, per a cada alumne:

- Una llista de les assignatures matriculades.
- Una llista de les assignatures aprovades.
- Una llista de les assignatures suspeses.
- Una llista de les notes de les assignatures aprovades.
- Una llista de les notes de les assignatures suspeses.

Totes aquestes dades recollides i organitzades, són la base del DataFrame que crearem (recompte) per dur a terme l'anàlisi final. Ja que, gràcies a com ha estat ordenat el DataFrame que s'ha iterat (*dfq2*) (primer pel codi de l'estudiant, després pel curs i per últim, per si s'ha aprovat o no), tots els diccionaris estan organitzats de la mateixa manera i, els codis dels estudiants, apareixen en el mateix ordre. Això facilita molt la feina ja que, simplement, cal afegir al DataFrame recompte els valors dels diccionaris com a columnes. Per un usuari que hagi suspès Càlcul I (1), Mecànica Fonamental (2) i Fonaments d'Informàtica (4), s'obté:

	CODEX	NºMatriculades	Assignatures	NºAprovades	Aprovades	NF_Apr	Suspeses	NF_Susp
0	227213	6	[5, 1, 8, 2, 4, 9]	5	[1, 8, 2, 4, 9]	[5.9, 6.8, 5.4, 6.5, 6.3]	[5]	[3.7]
1	229262	6	[8, 2, 4, 1, 6, 5]	3	[1, 6, 5]	[5.0, 5.1, 5.1]	[8, 2, 4]	[2.8, 3.7, 3.0]
2	232772	6	[7, 1, 9, 4, 8, 5]	5	[1, 9, 4, 8, 5]	[5.2, 7.8, 6.4, 6.3, 6.1]	[7]	[0.4]
3	245509	6	[8, 4, 1, 6, 2, 9]	3	[6, 2, 9]	[5.9, 6.4, 7.0]	[8, 4, 1]	[4.1, 4.1, 4.5]
4	258117	5	[9, 6, 2, 4, 7]	1	[7]	[5.0]	[9, 6, 2, 4]	[3.1, 2.9, 4.4, 2.7]
...	...	...	...	...	...	...	...	...

Figura 21. DataFrame "recomana" per un estudiant que ha suspès Càlcul I, Mecànica Fonamental i Fonaments d'Informàtica.

Font: elaboració pròpia 2021

Tot i ja tenir molta informació útil, encara falta afegir dues columnes més, el tant per cent d'assignatures aprovades respecte les matriculades, i la mitjana ponderada.

Pel que fa al tant per cent esmentat, simplement cal afegir una nova columna on es faci la divisió del número d'assignatures aprovades entre el número d'assignatures matriculades.

Per trobar la mitjana ponderada, es recorrerà el diccionari creat que tenia com a valors les assignatures matriculades i, a cada iteració, calcularem dos valors: el sumatori de la nota obtinguda a cada assignatura multiplicat pels respectius crèdit de l'assignatura i la suma total de crèdits matriculats. Totes aquestes dades les obtindrem del DataFrame *dfq2*, ja que és on podem trobar alhora els crèdits de las assignatures i la nota obtinguda per l'alumne. Fent servir la comanda *recomana.at[index, 'nom de la columna']* podem assignar la mitjana ponderada per a cada alumne.

Finalment, aquesta funció retornarà un DataFrame com el següent:

	CODEX	NºMatriculades	Assignatures	NºAprovades	Aprovades	NF_Apr	Suspeses	NF_Susp	Perc	Mitjana
0	227213	6	[5, 1, 8, 2, 4, 9]	5	[1, 8, 2, 4, 9]	[5.9, 6.8, 5.4, 6.5, 6.3]	[5]	[3.7]	83.33	5.745833
1	229262	6	[8, 2, 4, 1, 6, 5]	3	[1, 6, 5]	[5.0, 5.1, 5.1]	[8, 2, 4]	[2.8, 3.7, 3.0]	50.00	4.173913
2	232772	6	[7, 1, 9, 4, 8, 5]	5	[1, 9, 4, 8, 5]	[5.2, 7.8, 6.4, 6.3, 6.1]	[7]	[0.4]	83.33	5.429167
3	245509	6	[8, 4, 1, 6, 2, 9]	3	[6, 2, 9]	[5.9, 6.4, 7.0]	[8, 4, 1]	[4.1, 4.1, 4.5]	50.00	5.454167
4	258117	5	[9, 6, 2, 4, 7]	1	[7]	[5.0]	[9, 6, 2, 4]	[3.1, 2.9, 4.4, 2.7]	20.00	3.595238
...	...	...	...	...	...	...	...	...	...	...

Figura 22. DataFrame "recomana" per un estudiant que ha suspès Càlcul I, Mecànica Fonamental i Fonaments d'Informàtica.  
Font: elaboració pròpia 2021

### 10.1.2.3. Funció *exit* ( )

Prèviament, en l'apartat 7.4, s'ha introduït el concepte de les bonificacions i penalitzacions que es vol aplicar. Per poder realitzar-ho, és necessari conèixer el tant per cent d'èxit de cadascuna de les assignatures que es cursen a la fase inicial. Aquesta funció ens permetrà trobar-los.

Per aconseguir-ho, s'han creat dos DataFrames a partir del document *qfini2019filtrat.csv*: en el primer, s'imposarà que apareguin només totes les assignatures que s'hagin aprovat; en el segon, totes les que s'hagin suspès. A partir de la comanda *df.columna.value\_counts()*<sup>8</sup> s'obté el número de vegades que apareixen els elements que es troben en la columna indicada, per tant, ens servirà

<sup>8</sup> Permet comptar el número de cops que apareix un valor en una columna del DataFrame.

per identificar el número de cops que s'ha aprovat una assignatura i els cops que s'ha suspès.

En un nou DataFrame (*num2*), afegirem tres columnes:

- En la primera, posarem el número de cops que s'ha aprovat cada assignatura.
- En la segona, posarem el número de cops que s'han suspès cada assignatura
- En la tercera, posarem el resultat de la divisió entre la primera columna i la suma de les dues

	total_apr	total_susp	exit
3	3830	1179	0.7646
4	3827	1823	0.6773
0	3726	1574	0.7030
1	3687	1757	0.6773
2	3635	1933	0.6528
8	3610	1107	0.7653
9	3602	1110	0.7644
5	3577	1387	0.7206
6	3513	1363	0.7205
7	3440	1683	0.6715

Figura 23. DataFrame amb el número d'aprovat i suspesos per assignatura i, el tant per cent d'èxit de l'assignatura. Font: elaboració pròpia 2021

Com es pot observar a la Figura 8, els diferents indexes del DataFrame són els codis de les assignatures.

Per facilitar el tractament posterior de les dades, la funció retornarà una llista amb els indexes del DataFrame (d'on s'identifiquen les assignatures) i una llista de la columna *exit* (on es guarden els tant per cents d'èxit).

#### 10.1.2.4. Funció *addicio (l)*

En aquesta funció, donada una llista amb les assignatures suspesos de l'usuari, s'implementaran les esmentades bonificacions i penalitzacions als veïns. Internament, aquesta funció crida a les funcions *analitza\_estudiants (l)* (per aconseguir el DataFrame amb tota la informació dels veïns) i *exit ()* (per conèixer el tant per cent d'èxit de les assignatures). Degut a que ens interessa premiar les notes en els assignatures més complicades (aquelles que tinguin un percentatge de suspesos més elevat), es sumarà a la columna *PERC* la nota obtinguda a l'assignatura per el tant per cent de suspesos d'aquesta (és a dir 1 menys el tant per cent d'èxit obtingut).

A més, per fer-ho més realista, es multiplicarà la puntuació final obtinguda pel tant per cent de mostres que existeixen d'un bloc d'assignatures respecte les totals.

Es busca, doncs, augmentar o disminuir el valor de les entrades de la columna *PERC* per tenir en compte quina nota s'ha obtingut en les diferents assignatures.

#### 10.1.2.5. *Funció resposta (1)*

Finalment, fent servir aquesta funció (i donada una llista amb les assignatures que s'han suspès) s'obindrà la recomanació de la matrícula pel quadrimestre vinent.

Tenint en compte els diferents números d'assignatures a les que els estudiants es matriculen, s'ha escollit oferir a l'usuari la possibilitat de visualitzar diferents opcions de matrícules en funció del número d'assignatures que les componen. Es donaran, doncs, sis possibles opcions (només es podran visualitzar les recomanacions que continguin un conjunt d'assignatures): des d'una matrícula de tres assignatures, fins a una de vuit.

Aquesta funció, doncs, el que fa és buscar, per cada una de les sis possibles recomanacions, el bloc d'assignatures que ha aconseguit la màxima puntuació possible. Es poden donar tres situacions diferents:

- Que no hi hagi dades: és a dir, no s'ha pogut trobar cap veí; per tant, no es pot donar cap recomanació.
- Que existeixi una combinació la puntuació obtinguda de la qual sigui massa baixa com per a recomanar-la: tot i que, dins de tots els possibles escenaris, aquesta sigui la que tingui una millor puntuació, no es recomanarà.
- Que hi hagi dades suficients per donar una recomanació i, que aquesta, hagi obtingut una puntuació satisfactòria.

Es crea un diccionari on es guardaran cadascuna de les recomanacions. En el cas que no s'hagi pogut dur a terme una recomanació per a la clau X, com a valor tindrà una llista de dos elements: el primer, serà l'string "Cap opció viable"; i el segon serà un 0. Pels casos on el programa hagi estat capaç d'obtenir una recomanació, per a la clau X existirà com a valor una llista de també dos elements: en el primer es trobarà una llista amb tots els codis d'assignatura pertinents, i en el segon la puntuació aconseguida d'aquell bloc d'assignatures.

Tot i ja tenir el resultat esperat, se li han afegit dues tasques més:

- Transformar els codis de las assignatures que s'han fet servir durant tot l'estudi pels originals.
- Separar el diccionari de manera que la funció retorni les recomanacions per separat i una llista de totes les puntuacions obtingudes.

### 10.1.3. Fitxer *filtres\_q1\_q2\_q3.py*

Abans de poder iniciar l'estudi amb els alumnes que ja han cursat almenys dos quadrimestres del grau, s'han de tenir preparades les dades. En aquest fitxer, doncs, es filtraran les dades que es necessiten per poder fer el seguiment dels veïns dels usuaris.

#### 10.1.3.1. Funció *mercat()*

Dins d'aquesta funció es cridaran dos dels documents proporcionats per l'escola: *qfini2019filtrat.csv* i *qfnoi2019.csv*. Primer de tot, és necessari aplicar certs filtres en el DataFrame que es crea a partir de llegir el segon document:

- Canvi dels noms de les columnes.
- Acceptar únicament el codi 752 (Grau d'Enginyeria en Tecnologies Industrials) com a valor a la columna *CODPRO*.
- Descartar tots els cursos anteriors al nou pla d'estudis (2010).
- Eliminar columnes ja no útils (*CODPRO*, *NP*, *NC*, *GR*).
- Quedar-se únicament les assignatures cursades en el tercer quadrimestre del grau.
- Rebutjar entrades que tinguin com a valor en la columna del quadrimestre un 2.

Un cop filtrat, fent ús de la comanda *pd.concat([df1,df2])*<sup>9</sup>, ajuntarem els dos DataFrames en un de sol. En aquest nou DataFrame, se li ha d'aplicar la nomenclatura estipulada per les assignatures i ordenar-lo primer pel *CODEX*, després pel *CURS* i, per últim, per la columna *APR*.

En aquest estudi ens interessa analitzar què és el que fa l'estudiant en el quadrimestre 1 del seu segon any de carrera. Per tant, com que en el DataFrame ja només apareixen les assignatures cursades en el quadrimestre de tardor, s'hauran d'eliminar totes aquelles entrades que en la

---

<sup>9</sup> Amalgama els DataFrames indicats dins dels claudators per ordre.

columna *CURS* tinguin un valor diferent al del segon any de carrera.

Per fer-ho, es vol recrear la situació de l'apartat 10.1.2.2, on s'havien d'eliminar totes les entrades posteriors al primer any que apareix per un alumne en la columna *CURS* i aplicar la solució trobada. Per tant, s'ha d'aconseguir que l'any que ens interessa (que és el segon any que apareix per un alumne) es converteixi en el primer: és a dir, s'han d'eliminar totes aquelles entrades que continguin el primer any.

Fent servir la part del codi desenvolupada per solucionar el problema de l'apartat 10.1.2.2, aconseguirem obtenir un nou DataFrame amb totes les entrades que continguin el primer any. El que ens importa d'aquest nou DataFrame són els indexes de cada fila ja que, si no s'han modificat, equivalen als indexes de les entrades que cal eliminar del DataFrame original per replicar la situació esmentada prèviament.

Un cop arribats a aquest punt, simplement caldrà tornar aplicar el codi que s'ha fet servir anteriorment per tal que quedar-nos amb les entrades que, com a valor de la columna *CURS*, tenen el primer any que apareix per un estudiant. Ara ja es pot crear un nou fitxer *.csv* per a que es faci servir en el següent estudi.

S'ha de tenir en compte que el procediment fet servir no seria útil o recomanable si s'hagués d'executar cada vegada que un usuari volgués obtenir una recomanació ja que, les iteracions utilitzades, empitjoren el rendiment i fan que el temps d'execució augmenti considerablement.

#### 10.1.4. Fitxer *programaQ\_altres.py*

En aquest fitxer es durà a terme tot l'estudi d'aquells estudiants que ja hagin realitzat, com a mínim, el primer any de la carrera, però no hagin pogut finalitzar la fase inicial.

Aquest fitxer segueix el mateix procediment que el *programaQ1* explicat anteriorment. Tot i així, degut a que es contemplin diferents situacions a l'hora d'iniciar l'estudi, s'han aplicat diverses modificacions. Així doncs, aquelles funcions que ja hagin estat explicades anteriorment, només s'explicaran les modificacions fetes.

##### 10.1.4.1. Funció *vecinos (ls)*

Aquesta funció, igual que la funció *veïns(l)* de l'apartat 10.1.2.1, rep com a entrada una llista amb les assignatures que l'usuari ha suspès i retorna el DataFrame amb les veïns de l'usuari. L'única

diferència que existeix és que, ara, es tindran en compte fins a set casos ja que l'usuari pot haver suspès més de cinc assignatures.

#### 10.1.4.2. Funció *df\_q3* ( )

L'objectiu que te aquesta funció és llegir i filtrar el document *qfnoini2019.csv* proporcionat per l'escola. Se li aplicaran els següents canvis:

- Canvi dels noms de les columnes.
- Acceptar únicament el codi 752 (Grau d'Enginyeria en Tecnologies Industrials) com a valor a la columna *CODPRO*.
- Eliminar les entrades que estiguin convalidades
- Descartar tots els cursos anteriors al nou pla d'estudis (2010).
- Eliminar columnes ja no útils (*CODPRO*, *NP*, *NC*, *GR*).
- Quedar-se únicament les assignatures cursades en el tercer quadrimestre del grau.
- Rebutjar entrades que tinguin com a valor en la columna del quadrimestre un 2.
- Aplicar el canvi de nom al codi d'assignatures pel definit prèviament.

Per tant, la funció retornarà un DataFrame filtrat, que ja es pot fer servir per l'estudi.

#### 10.1.4.3. Funció *analiza\_vecinos* (*la*, *ls*)

Degut a que ara la major part dels usuaris ja hauran aprovat com a mínim una assignatura, a part de la llista d'assignatures suspeses, també hauran de proporcionar una llista amb les assignatures que ja aprovades. Això es deu a que, si no es disposés d'aquesta informació, se li podria recomanar que es matriculés a assignatures ja aprovades i, per tant, seria una recomanació poc útil i incorrecte. Això fa que les dades amb les que es treballen es redueixin considerablement i, per tant, és possible que, en diferents casos, no es tingui suficient informació com per donar cap recomanació.

El procediment que he seguit en aquesta funció per aconseguir el DataFrame amb tota la informació necessària, és igual que el de l'apartat 10.1.2.2, amb les úniques diferències que, ara, el document que es llegeix per conèixer el que han fet els veïns és: *alumnos\_q1\_q2\_q3\_filtrat.csv* (document obtingut a la funció *mercat* ( ) del fitxer *filtrat\_q1\_q2\_q3.py*). El problema que hi havia respecte a l'aparició d'entrades amb anys no desitjats, ja ha estat tractat dins del mateix document.

Per tant, altra vegada, aquesta funció retorna el mateix DataFrame que a la Figura 22, però amb

les dades corresponents a l'estudi actual.

#### 10.1.4.4. Funció *addicion (la, ls)*

Com en la funció descrita en l'apartat 10.1.2.4, aquí és on s'aplica, seguint el mateix procediment, el sistema de bonificacions i penalitzacions a la puntuació de cadascun dels blocs d'assignatures. Ara bé, previ a aplicar-ho, la funció *addicion/la, ls* elimina totes les entrades del DataFrame (obtingut a partir de la funció *analiza\_vecinos (la, ls)*) que, a la columna d'assignatures matriculades, contenen alguna de les assignatures que apareixen a la llista d'aprovades (proporcionada per l'usuari). D'aquesta manera ens assegurem que en la recomanació no aparegui cap assignatura que l'usuari ja hagi aprovat.

#### 10.1.4.5. Funció *respuesta (la, ls)*

A partir de llegir el DataFrame que retorna la funció *addicion (la, ls)* i, seguint el mateix procediment que la funció explicada a l'apartat 10.1.2.5, la funció *respuesta (la, ls)* retorna les sis possibles recomanacions que es proposaran a l'usuari i una llista amb totes les puntuacions obtingudes.

## 10.2. Lloc Web

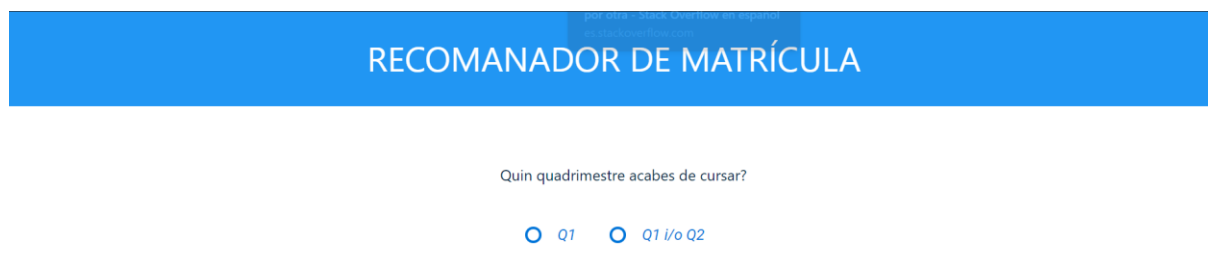
Per poder oferir aquest servei a l'estudiant és primordial que existeixi una plataforma amb la que ens puguem comunicar, tant per rebre la informació que es necessària per executar les funcions, com per poder-li oferir una recomanació que li sigui útil. Per això dissenyarem un lloc web.

Aquest lloc web s'ha plantejat perquè sigui molt intuïtiu, és a dir, que no requereixi de llegir un manual o de navegar durant molta estona per entendre'n el funcionament. Ara bé, tot i buscar senzillesa, he cregut important que la pàgina web ha de ser dinàmica i reaccioni a l'activitat de l'usuari.

Seguint aquest plantejament he ideat dos escenaris diferents: la pàgina principal i la pàgina amb la resposta.

### 10.2.1 Pàgina principal

És el primer que veurà l'usuari tant bon punt entri a la pàgina web, i on se li demanarà que introdueixi la seva informació acadèmica.



Quin quadrimestre acabes de cursar?

Q1  Q1 i/o Q2

Figura 24. Pàgina d'inici del lloc web. Font: elaboració pròpia 2021

Com es pot observar a la Figura 14, es deixa escollir a l'estudiant entre les dues opcions que s'han contemplat en aquest projecte. Aquells usuaris que acabin de cursar per primera vegada el quadrimestre 1, hauran de seleccionar el *checkbox* Q1. Per altra banda, aquells que ja es trobin en un punt més avançat de la fase inicial (i que, per tant, han cursat assignatures suspeses del Q1, assignatures del Q2 o una barreja de les dues) hauran de prémer el *checkbox* Q1 i/o Q2.

Per afegir-hi el dinamisme desitjat, i fent us de JavaScript, he fet que el formulari que s'ha d'omplir (el de Q1 o el de Q1 i/o Q2) aparegui i desaparegui en funció del *checkbox* marcat.

#### 10.2.1.1. Formulari per l'opció Q1

En aquest cas, degut a que la funció només requereix d'una llista amb les assignatures que s'han suspès al primer quadrimestre, simplement caldrà mostrar un formulari on l'alumne pugui introduir aquesta informació i enviar-nos-la.

## RECOMANADOR DE MATRÍCULA

Quin quadrimestre acabes de cursar?

Q1  Q1 i/o Q2

---

Selecciona les assignatures que hagi suspès

<input type="checkbox"/>	Àlgebra Lineal
<input type="checkbox"/>	Fonaments d'Informàtica
<input type="checkbox"/>	Química I
<input type="checkbox"/>	Càlcul I
<input type="checkbox"/>	Mecànica Fonamental

[Envia](#)

Figura 25. Formulari de Q1 que ha d'omplir l'alumne. Font: Elaboració pròpia, 2021

Quan l'alumne hagi seleccionat les assignatures que ha suspès i premi el botó *Envia*, se'l redirigirà immediatament a la pàgina on rebrà la resposta.

## RECOMANADOR DE MATRÍCULA

Quin quadrimestre acabes de cursar?

Q1  Q1 i/o Q2

---

Selecciona les assignatures que hagi suspès

	Àlgebra Lineal
	Fonaments d'Informàtica
<input type="checkbox"/>	Química I
<input type="checkbox"/>	Càlcul I
	Mecànica Fonamental

[Envia](#)

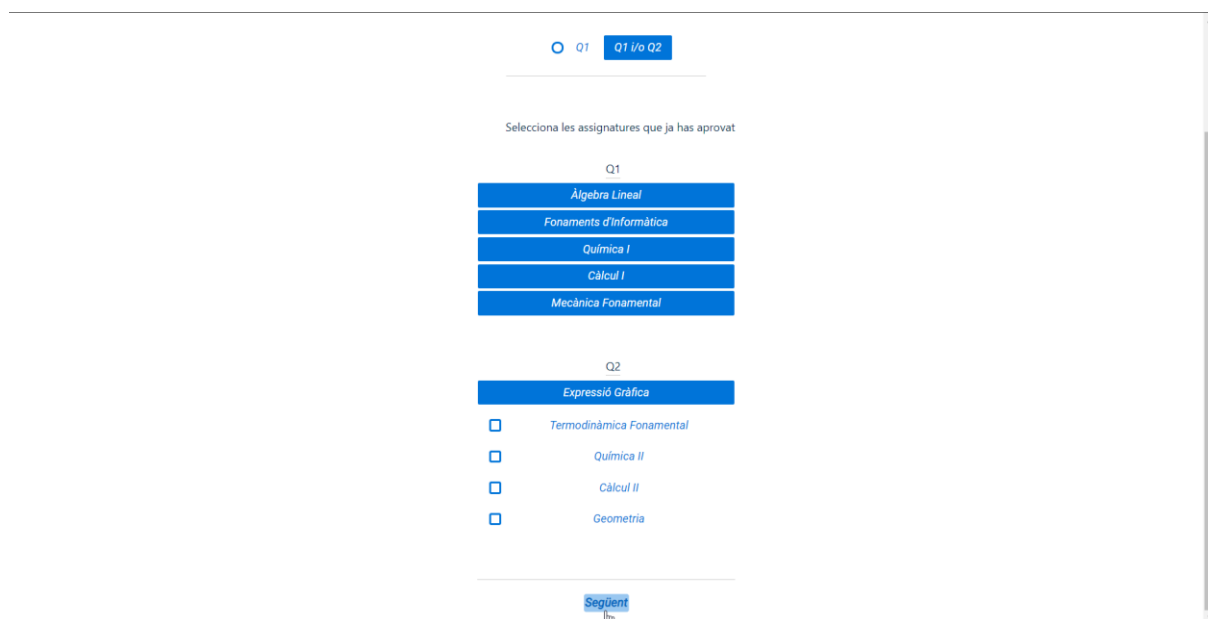
Figura 26. Formulari de Q1 omplert per l'alumne. Font: elaboració pròpia 2021

### 10.2.1.2. Formulari de Q1 i/o Q2

Per als estudiants que, com a mínim, acaben de cursar el seu segon quadrimestre del grau, serà necessari que ens diguin quines són les assignatures que ja han aprovat i les que acaben de suspendre. Per tant, tot i estar tot configurat en el mateix formulari, s'ha organitzat de manera que

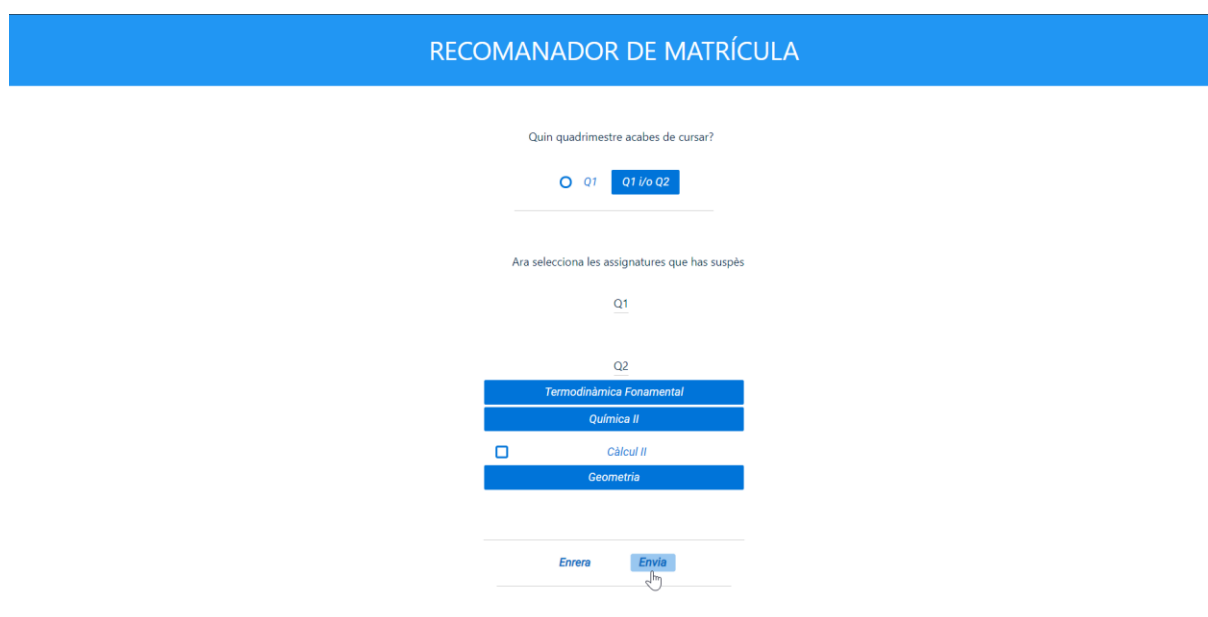
hi hagi dues etapes: en la primera es seleccionaran les assignatures ja aprovades; i en la segona les ja suspeses.

Per tal que se n'entengui el funcionament, es replicarà el que hauria de fer un alumne que ja té aprovades Àlgebra Lineal, Fonaments d'Informàtica, Química I, Càlcul I, Mecànica Fonamental i Química II (Figura 17) però que, en l'últim quadrimestre, ha cursat i suspès Expressió gràfica, Termodinàmica Fonamental i Geometria (Figura 18).



The screenshot shows a web interface for selecting approved subjects. At the top, there are radio buttons for 'Q1' (selected) and 'Q1 i/o Q2'. Below this, the instruction 'Selecciona les assignatures que ja has aprovat' is displayed. The subjects are grouped by quarter: Q1 includes 'Àlgebra Lineal', 'Fonaments d'Informàtica', 'Química I', 'Càlcul I', and 'Mecànica Fonamental', all shown as solid blue buttons. Q2 includes 'Expressió Gràfica' (solid blue button), 'Termodinàmica Fonamental', 'Química II', 'Càlcul II', and 'Geometria' (all with checkboxes). At the bottom, there is a 'Següent' button.

Figura 27. Exemplificació de la selecció d'assignatures ja aprovades. Font: elaboració pròpia 2021



The screenshot shows a web interface for selecting suspended subjects. At the top, there are radio buttons for 'Q1' and 'Q1 i/o Q2' (selected). Below this, the instruction 'Ara selecciona les assignatures que has suspès' is displayed. The subjects are grouped by quarter: Q1 has no subjects listed. Q2 includes 'Termodinàmica Fonamental', 'Química II', 'Càlcul II', and 'Geometria', all shown as solid blue buttons with checkboxes. At the bottom, there are 'Enrera' and 'Envia' buttons.

Figura 28. Exemplificació de la selecció d'assignatures suspeses. Font: elaboració pròpia 2021

Si comparem les Figures 17 i 18 podem identificar que, al prémer el botó *Següent*, les assignatures

marcades (les que ja han estat aprovades) desapareixen i canvia el text que descriu l'acció que ha de fer l'usuari, donant lloc a la selecció de les assignatures suspeses.

A part del botó *Envia*, a la Figura 18 també veiem el botó *Enrere*. Si el preméssim, es tornaria enrere en el formulari de manera que les assignatures eliminades tornarien a aparèixer i ens trobaríem en la mateixa situació que en la de la Figura 17 (sense cap opció seleccionada).

Altra vegada, al prémer *Envia*, es redirigeix directament a la pàgina on es trobarà la recomanació.

### 10.2.2. Pàgina amb la recomanació

Degut que es plantegen dues situacions diferents, existiran dues pàgines que contindran la resposta. Ara bé, aquestes pàgines estan configurades de la mateixa manera, pel que es tractaran com una sola.

Quan s'hagi enviat el respectiu formulari (explicat en els dos últims apartats) es mostraran les millors opcions de matrícules en funció del número d'assignatures que continguin. Primer es mostraran, altra vegada, un seguit d'inputs (de tipus *checkbox*) per a que l'usuari pugui seleccionar, en funció del número d'assignatures, la matrícula que li sigui més atractiva. Ja seleccionat un dels *checkbox*, es mostrarà la recomanació pertinent. Cal destacar que, en els casos extrems (matrícula amb 3 assignatures i matrícula amb 8 assignatures), és habitual que no s'hagi pogut obtenir una recomanació. En aquests casos, s'ha configurat de manera que, a l'iniciar la pàgina, s'amaguin les opcions que no continguin cap recomanació, abans d'ensenyar res.

En el cas d'un estudiant que hagi suspès Àlgebra Lineal i Fonaments d'Informàtica, la primera vegada que cursava el Q1, s'obtindran les següents opcions de recomanació:

RECOMANADOR DE MATRÍCULA

Escull el número d'assignatures a les que et voldries matricular

Has de saber que el bloc de **cinc** assignatures és el més recomanat per a tu de cara a la propera matrícula

3 Assignatures
  4 Assignatures
  5 Assignatures
  6 Assignatures
  7 Assignatures

[Enrere](#)

Figura 29. Recomanació d'un estudiant que ha cursat per primer cop Q1 i ha suspès Àlgebra Lineal i Fonaments d'Informàtica. Font: elaboració pròpia 2021

Es pot observar que tot i ensenyar les diferents opcions existents (en quant a número d'assignatures a matricular), es remarca quina d'elles és la que ha obtingut els millors resultats.

Si volem, per exemple, veure la recomanació que dona el programa amb una matrícula de cinc assignatures, que és en aquest cas la que ha obtingut una puntuació més alta, hauríem de prémer 5 assignatures:

RECOMANADOR DE MATRÍCULA

Escull el número d'assignatures a les que et voldries matricular

Has de saber que el bloc de **cinc** assignatures és el més recomanat per a tu de cara a la propera matrícula

3 Assignatures
  4 Assignatures
  5 Assignatures
  6 Assignatures
  7 Assignatures

Aquestes són les 5 assignatures que et recomanem fer

Termodinàmica Fonamental
  Expressió Gràfica
  Àlgebra
  Càlcul I
  Química II

[Enrere](#)

Figura 30. Recomanació d'una matrícula de cinc assignatures per l'estudiant del cas anterior. Font: elaboració pròpia 2021

Per fer encara més evident el bloc d'assignatures que ofereix la millor puntuació, es mostraran les assignatures amb un verd. Les altres, mantindran el fons blau que s'ha fet servir fins ara.



Figura 31. Exemple de la visualització d'un conjunt d'assignatures que no ha obtingut la puntuació màxima. Font: elaboració pròpia 2021

Per últim, s'ha tingut en compte un tercer escenari, on amb les dades proporcionades per l'estudiant no es pugui arribar a proporcionar cap recomanació. És doncs quan es mostrarà la següent pàgina:



Figura 32. Pàgina que apareix quan no existeix cap recomanació. Font: elaboració pròpia 2021

### 10.2.3. Aplicació de l'HTML

Com ja s'ha explicat a l'apartat 9.1.2, l'HTML ens servirà per crear l'estructura i, introduir els elements que es visualitzin a la pàgina web mitjançant etiquetes. Principalment, s'han usat les següents:

- Formularis (`<form></form>`): Part fonamental del funcionament del lloc web, ja que és l'etiqueta que ens permet enviar la informació de l'usuari al sistema. Tot i així, també s'han fet servir per contenir els inputs de tipus *checkbox* que ens permeten ensenyar o amagar contingut. Així doncs, s'han implementat fins a cinc formularis diferents. Tres en la pàgina principal:
  - Formulari per navegar entre *Q1* i *Q1 i/o altres*.
  - Formulari per recopilar la informació de *Q1*.
  - Formulari per recopilar la informació de *Q1 i/o altres*.

I un a cadascuna de les pàgines de resposta:

- Formulari per navegar entre les diferents opcions de matricules proposades.
- Div (`<div></div>`): Aquesta etiqueta és, segurament, una de les més utilitzades, ja que permet agrupar tot tipus de contingut (altres etiquetes) en caixes que, posteriorment i amb CSS podràs ubicar i estilitzar com millor et convingui. Per tant, ens ha estat molt útil per exemple, per agrupar dins de cada formulari els elements input i label (conjuntament formen les opcions seleccionables que s'han ensenyat als apartats previs).
- Estructura *Checkboxes* i *labels*(`<input type = "checkbox"></input> + <label></label>`): És l'element que permet aportar tot el dinamisme i atractiu a la pàgina web. Ens ha estat molt útil ja que, definint un valor a cadascuna de les etiquetes input, ens ha permès tant seleccionar la informació que és vol transmetre al servidor (el valor d'un *checkbox* és el contingut que s'envia). També ens ha permès identificar els *checkboxs* marcats amb JavaScript.

#### 10.2.4. Aplicació del CSS

En les figures adjuntades anteriorment, es pot observar com clarament s'ha aplicat el mateix estil en la major part del elements que apareixen en cadascuna de las pàgines de la web. Parlo de l'estil dels *checkboxs*. Vaig voler implementar un estil diferent als que vénen per defecte i després de buscar possibles exemples, vaig acabar trobant el que més creia que encaixaria dins la pàgina web. La principal característica dels *checkboxs* és que es defineix un estil per tres dels seus estats: quan està desmarcat, quan el ratolí passa per sobre i quan es marca.

Quan es troben desmarcats, sembla que siguin com qualsevol altre input amb un nom i amb una caixa (forma de circumferència o quadrada) que identifica si es troba marcat o desmarcat (en aquest cas desmarcat).



Figura 33. Boto desmarcat.

Font:elaboració pròpia 2021



Figura 34. Checkbox amb el ratolí a sobre. Font: elaboració pròpia 2021

Si es passa el ratolí per sobre fent us de l'atribut `.classe: hover{}` s'aconsegueix crear un nou estat en el que el color de fons passa de ser blanc a ser un blau clar i, ja es pot identificar clarament el tamany de l'input i les cantonades arrodonides.

En el moment en el que es prem, es passa a un nou estat on s'ha fet desaparèixer la figura, s'ha canviat el color de la lletra (de blanc a blau) i, per remarcar l'opció que es tria, es canvia el color de fons a un blau més fosc i intens i es centra el text dins del *checkbox*.

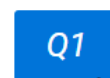


Figura 35. Checkbox seleccionat

Degut a que s'ha eliminat la figura, el tamany del *checkbox* ha disminuït però com que es troben continguts en un caixa (div) a la que si li ha aplicat un *display: flex*, quan es trobin orientats horitzontalment, els altres *checkboxs* ocuparan l'espai que ha quedat lliure.

Quin quadrimestre acabes de cursar?

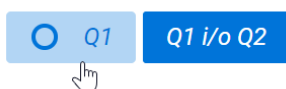


Figura 36. Q1 i/o Q2. Font: elaboració pròpia 2021

Quin quadrimestre acabes de cursar?

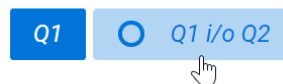


Figura 37. Q1 marcada. Font: elaboració pròpia 2021

### 10.2.5 Aplicació del JavaScript

Ens ha permès aplicar tot el moviment existent en la pàgina web, des d'amagar contingut fins a fer variar la classe CSS d'un element. Aquestes són algunes de les funcions que s'ha aconseguit aplicar gracies a JavaScript:

- Amagar i ensenyar contingut: És, possiblement, l'acció més recurrent que es troba dins el lloc web. A partir de determinar quan es fa clic sobre un input (`$("#id").click(function(){})`), i, generalment, aplicant una estructura de condicionals, es pot determinar fàcilment què és el que es vol ensenyar amb la comanda `$("#id").show()` i, què es vol amagar amb

`$("#id").hide()`. A més, dins del parèntesis es pot introduir un número que indicarà la rapidesa o la lentitud amb la que es durà a terme l'acció.

- Canviar el codi CSS: Ha estat útil poder modificar, a través de JavaScript, els estils aplicats a certs elements, en funció de l'activitat de l'usuari. Aquesta tasca es duu a terme principalment gràcies a tres comandes: `$("#Id" o ".Class").css("property", "value")` i `("Id" o ".Class").style.property = "value"` permeten modificar el valor d'una propietat, en concret de l'estil d'un element; i `$(".Class").toggleClass(".Nova_Clase")` canvia directament la classe d'un element de manera que s'apliquin uns estils definits prèviament en l'apartat del codi CSS.
- Obtenir elements de l'HTML: És habitual requerir d'un element en concret. Per exemple, hem requerit, a l'hora d'implementar l'estructura de condicionals, conèixer els valors dels *checkboxs*. Hi ha diverses maneres d'accedir a un element HTML, en aquest treball s'han usat principalment dos: `document.getElementById("Id")` i `document.getElementsByClassName("Clase")`. Combinant aquestes comandes amb d'altres, es pot aconseguir fer gairebé qualsevol cosa que es necessiti: accedir als *ChildNodes*<sup>10</sup>, modificar el valor, substituir tota l'estructura del codi de l'element, etc.
- Fer iteracions: Ens ha estat molt útil poder iterar tant la informació que es rep des del servidor com llistes que contenen identificadors de diversos blocs. Amb JavaScript, una de las formes d'iterar (i la feta servir en aquest treball) és fent ús de la següent expressió: `$.each(variable a iterar, function(ind, elem){})`. S'han fet servir iteracions, per exemple, per poder ensenyar només els blocs d'assignatures que no estiguessin buits.
- Estructura de condicionals: En Python i en la majoria de llenguatges, aquest tipus d'estructures són molt recurrents, ja que ens permet filtrar la informació en diferents categories. En JavaScript, el format per fer-ho és molt semblant i l'expressió que engloba a tots els possibles condicionals és la següent:

---

<sup>10</sup> Elements HTML que són fills, és a dir, que es troben dins d'un altre element

```
if (condició && condició){  
    accions  
} else if (condició || condició){  
    accions  
} else{  
    accions  
}
```

Figura 38. Estructura del condicionals. Font: elaboració pròpia 2021

El terme `&&`, imposa que s'hagin de complir totes les condicions imposades per poder accedir a fer les accions que es troben dins del condicional. I el terme `||`, indica que s'ha de complir alguna de les condicions escrites per a que es realitzin les accions definides. Normalment, existeixen tres tipus de condicions: aquelles que imposen igualtat (`==` o `===`), les que imposen desigualtat (`!=` o `!==`) i les que comparen la magnitud dels valors (`<`, `>`, `<=`, `>=`).

- Executar una funció abans de mostrar el contingut de la pàgina: Per eliminar les opcions de matrícules que no continguessin informació, s'ha aplicat una iteració i una estructura de condicionals, sobre una llista creada amb dades proporcionades per Flask (es fa referència a aquestes dades posant: `'{{variable}}'`) dins de la comanda `$(document).ready(function(){})`. Aquesta comanda, executa tot el que es troba dins d'ella abans d'ensenyar qualsevol altre element del document HTML.

## 11. COSTOS, PLANIFICACIÓ I IMPACTE AMBIENTAL

### 11.1. Costos

Els costos que s'han hagut de tenir en consideració en quant a instruments o maquinària són molt reduïts ja que, per poder dur a terme aquest treball, només ha estat necessari utilitzar un ordinador i tot el programari utilitzat ha estat gratuït. Ara bé, també s'han de tenir en compte les hores invertides en el projecte, el preu que s'ha d'abonar a la universitat per poder realitzar el Treball de Fi de Grau i, per últim, uns cursos prèviament realitzats, a través de la plataforma Domestika, enfocats a aprofundir el coneixement necessari per desenvolupar una pàgina web. En la següent taula:

ANÀLISI COSTOS TFG		
<b>PC amortitzat</b> <b>56,60 €</b>	PVP ordinador Vida mitjana 5 anys Temps usat pel treball	848,99 € 60 mesos 4 mesos
<b>Preu mà d'obra</b> <b>4.008 €</b>	Hores invertides Preu/hora mig (eng.indus ESP)	300h 13,36€/h
<b>Preu matriculació TFG</b> <b>332,04 €</b>	Preu per crèdit ECTS Número de crèdits TFG	27,67€/ECTS 12 ECTS
<b>Cursos Domestika</b> <b>40 €</b>	4 cursos en total	10€/curs
<b>COST TOTAL DEL PROJECTE</b>		<b>4.436,64 €</b>

Taula 3. Càlcul dels diferents costos tinguts en compte. Font: elaboració pròpia 2021

En el cas que es plantegés l'opció de portar al mercat aquest projecte, es podria comercialitzar de dues maneres: venent el projecte en si, o bé cedint-ne l'ús a canvi d'un pagament mensual o anual.

En el cas de vendre'l, s'han plantejat dos possibles escenaris que varien segons el marge de ventes fixat: aquest marge serà del 30% o del 35% respectivament. D'aquesta manera ens podrem protegir i prevenir sobre possibles despeses imprevistes o difícils de quantificar. Realitzant els càlculs pertinents, s'arriba a dos possibles preus de venda:

Quin és el PVP ideal pel qual estaria disposat a vendre aquest projecte?		
Fórmula:	<b>Marge de ventas = (PVP - Cost)/PVP</b>	
	Marge de venta ideal =	30%-35%
	PVP =	X
	Cost projecte =	4.436,64 €
<b>RESULTAT</b>	Si marge de venta = 30%	<b>6.338,06 €</b> PVP
	Si marge de venta= 35%	<b>6.825,60 €</b> PVP

Taula 4. Preu de venda del projecte. Font: elaboració pròpia 2021

Per altra banda, si s’analitza l’altra manera de comercialitzar el projecte, s’obté el següent:

Quin és el preu/any pel qual estaria disposat a llogar aquest projecte?		
Supòsit:	<b>Permanència/Compromís a 4 anys (durada grau)</b>	
	TOTAL	CADA ANY
Cost (imputable = mà d'obra)	4.008 €	1.002 €
Actualitzacions base de dades	1.000 €	250 €
Manteniment	800 €	200 €
Ampliació (a tota el grau)*	12.000 €	3.000 €
Atenció al client (per errors, etc.)*	1.200 €	300 €
<b>RESULTAT</b>	Preu anual	<b>1.452 €</b> Aprox. 2000€/semestre
	Preu anual	<b>4.752 €</b> Suposem 25€/mes

\* Serveis opcionals, es poden contractar per separat

Taula 5. Preu anual per cedir-ne l'ús. Font: elaboració pròpia 2021

## 11.2. Planificació

La primera tasca que es va realitzar del treball va estar fer un llistat de tot el que creia que s’havia de realitzar per completar aquest projecte. Amb aquest llistat de tasques, es va realitzar un diagrama de Gantt que m’ajudés a organitzar i planificar el temps que li volia dedicar a cadascuna d’elles.

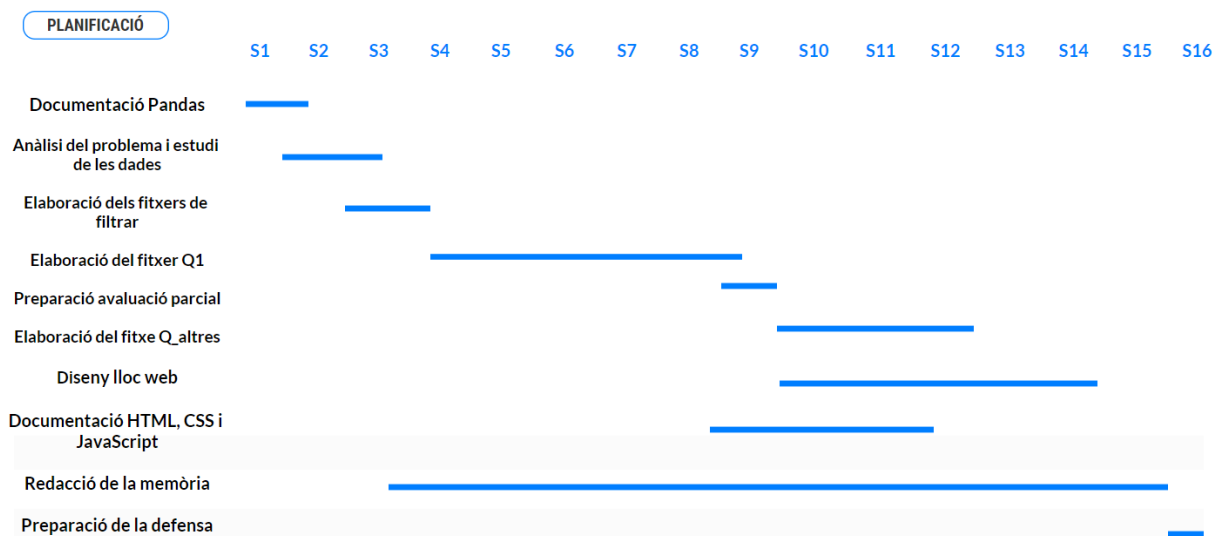


Figura 39. Diagrama de Gantt. Font: elaboració pròpia fet ús del web VENNGAGE

### 11.3. Impacte Ambiental

En tot desenvolupament, és important tenir en compte quin serà el seu impacte ambiental ja que, amb la situació global que es viu, s'ha d'intentar reduir al màxim. A més, de cara a vendre el projecte a una empresa, com menor sigui l'impacte ambiental causat, millor.

Per aquest projecte, l'impacte és realment petit ja que es pot realitzar tot només a partir d'un ordinador. Per tant, l'única manera per la que es perjudicarà l'ambient serà a partir de la consum d'electricitat.

Segons un informe de la Generalitat de Catalunya, la taxa es calcula a partir del consum en kWh anuals que queda reflectit a les factures de la companyia elèctrica. Usualment, les factures són bimensuals. Si es conserven les factures de tot l'any, serà suficient sumar-ne els consums. En cas contrari se'n pot fer una aproximació. El consum total calculat ha de ser dividit pel nombre de persones de la família. Quilowatts hora consumits per persona x 0,84 kg CO<sub>2</sub>/kWh = kg de CO<sub>2</sub><sup>11</sup>

Per tant, previ a realitzar el càlcul, fa falta fer una estimació dels kWh que s'han consumit. Com bé s'explica en l'article escrit per *chc energia*<sup>12</sup> al 22 de juny de 2019, existeixen molts factors que fan variar el nivell de consum d'un ordinador, dels quals destaquen la data de fabricació per sobre dels altres. En el seu estudi s'arriba a la conclusió que la mitja del consum d'un ordinador és entre 200 i 300 W. Per a realitzar els nostre càlcul es farà servir un consum de 250 W:

$$\frac{250 \text{ W} \cdot 300 \text{ h}}{1000} \cdot 0,84 \frac{\text{kg CO}_2}{\text{kWh}} = 63 \text{ kg de CO}_2$$

Equació 1. Càlcul dels quilograms de CO<sub>2</sub> emesos degut al projecte. Font: elaboració pròpia 2021

<sup>11</sup> Informació obtinguda del document [http://www.gencat.cat/icaen/reCORREGUTENERGIA/experiencies/Experiencia-2\\_vr.3.3.pdf](http://www.gencat.cat/icaen/reCORREGUTENERGIA/experiencies/Experiencia-2_vr.3.3.pdf)

<sup>12</sup> Informació obtinguda del document <https://chcenergia.es/blog/cuanto-consume-un-ordenador-o-pc/>

## 12. CONCLUSIONS

Fa quatre mesos, vaig plantejar una hipòtesi que a dia d'avui considero acceptada: crec que puc afirmar que els coneixements obtinguts del grau d'Enginyeria en Tecnologies Industrials, sumats a les bases de dades de què disposa l'ETSEIB i l'ús de diferents llenguatges de programació i les seves llibreries, són suficients per crear un recomanador de matrícula en format web. No obstant això, és cert que l'abast del meu treball s'ha vist limitat (sobretot per falta de temps) a la fase inicial del grau (Q1 i Q2).

L'eina creada m'ha fet combinar coneixements d'estadística, probabilitat, anàlisi de dades i programació. Mitjançant l'anàlisi del comportament dels estudiants passats he aconseguit crear una web que doni suport als estudiants futurs. Aquest anàlisi de comportament, sumat a la creació d'un criteri de puntuació, fa que consideri realistes i fiables els resultats que se n'obtenen.

A més, crec convenient mencionar el potencial de futur que té aquest projecte. A continuació n'exposaré les tres ampliacions que crec que poden ser més útils i interessants:

1. Adaptar el projecte a tot el grau d'Enginyeria en Tecnologies Industrials (i no limitar-ho només a la fase inicial). Crec que es disposa de suficients dades i coneixements per tirar-ho endavant amb facilitat, tot i que caldria tenir en compte la complexitat que comporten els quadrimestres en els que es poden escollir assignatures optatives.
2. Adaptar el projecte a altres graus universitaris (siguin de la UPC o no) que també tinguin un sistema de matriculació de caire quadrimestral.
3. Crear-ne una aplicació mòbil (APP) o, almenys, fer que la web sigui *responsive*, és a dir, que s'adapti a tot tipus de pantalles tant verticals com horitzontals.

Dit tot això, crec que hi ha una limitació que cal mencionar, i que crec que no seria possible de fer-ne una ampliació: els programes ERSAMUS i altres convenis d'intercanvi d'estudiants, ja que la recollida de dades hauria de ser tant gran i de tantes fonts diferents (diferents idiomes, universitats, graus i països) que seria impossible poder-hi donar a l'abast.

Finalment, és un projecte que he gaudit molt ja que m'ha fet dedicar temps a ampliar els meus coneixements en un àmbit que el grau cursat no s'ha centrat especialment: el de la programació i la informàtica.

### 13. BIBLIOGRAFIA

- [1] AGUILAR, J. Mostrar y ocultar el contenido de un div con un checkbox - Jose Aguilar Blog. A: [en línea]. Disponible a: <https://www.jose-aguilar.com/blog/mostrar-y-ocultar-el-contenido-de-un-div-con-un-checkbox/>.
- [2] ALEKZ, M., 2017. jquery - Cargar archivo JSON y almacenar contenido en variable - Stack Overflow en español. A: [en línea]. Disponible a: <https://es.stackoverflow.com/questions/57899/cargar-archivo-json-y-almacenar-contenido-en-variable>.
- [3] ANDROS FONOLLOSA., 2018. CSS checkbox y radios personalizados | Programador Web Valencia. A: [en línea]. Disponible a: <https://programadorwebvalencia.com/css-checkbox-y-radio-personalizados/>.
- [4] ARTURO, C., 2015. Radio Buttons y Checkboxes personalizados con HTML y CSS. A: [en línea]. Disponible a: <http://www.falconmasters.com/web-design/radio-buttons-checkbox-personalizados/>.
- [5] CARRODEGUAS, N. Como ocultar y mostrar elementos en páginas web con JavaScript. A: [en línea]. Disponible a: <https://norfipc.com/inf/javascript-como-ocultar-mostrar-elementos-paginas-web.html>.
- [6] CARRODEGUAS, N. JQuery, ejemplos prácticos para usar en páginas web. A: [en línea]. Disponible a: <https://norfipc.com/codigos/jquery-ejemplos-practicos-usar-paginas-web.php>.
- [7] CARRODEGUAS, N. JQuery, todos los selectores, eventos, métodos y funciones. A: [en línea]. Disponible a: <https://norfipc.com/codigos/jquery-todos-selectores-eventos-metodos-funciones.php>.
- [8] CASANOVA., 2018. Personalizar checkboxes y radio buttons | Front.id. A: [en línea]. Disponible a: <https://front.id/es/articles/personalizar-checkboxes-y-radio-buttons>.
- [9] CHC ENERGIA., 2019. Cuánta electricidad consume un ordenador - CHC Energía. A: [en línea]. Disponible a: <https://chcenergia.es/blog/cuanto-consume-un-ordenador-o-pc/>.

- [10] CODIGOFACILITO., 2016. 11.- Curso Flask - Formularios. A: [en línia]. Disponible a: [https://codigofacilito.com/videos/curso\\_flask\\_formularios](https://codigofacilito.com/videos/curso_flask_formularios).
- [11] CODIGOFACILITO., 2016. (11) 12.- *Curso Flask - Datos Formularios - YouTube* [en línia]. Disponible a: [https://www.youtube.com/watch?v=9S\\_f11vZ5IQ&ab\\_channel=codigofacilito](https://www.youtube.com/watch?v=9S_f11vZ5IQ&ab_channel=codigofacilito).
- [12] COSAS DE DEVS. Todo lo que necesitas para aprender PYTHON ya ? A: [en línia]. Disponible a: <https://cosasdedevs.com/python/>.
- [13] CRUZ, K., 2019. javascript - Mostrar divs al hacer click en un radiobutton - Stack Overflow en español. A: [en línia]. Disponible a: <https://es.stackoverflow.com/questions/298062/mostrar-divs-al-hacer-click-en-un-radiobutton?rq=1>.
- [14] DANIEL ÁVAREZ GILL., 2020. Primeros pasos con Pandas - Adictos al trabajo Tutoriales. A: [en línia]. Disponible a: <https://www.adictosaltrabajo.com/2020/12/29/primeros-pasos-con-pandas/>.
- [15] ETSEIB. Pla d'estudis GETI — Escola Tècnica Superior d'Enginyeria Industrial de Barcelona. ETSEIB — UPC. Universitat Politècnica de Catalunya. A: [en línia]. Disponible a: <https://etseib.upc.edu/ca/estudis/graus/pla-destudis-geti>.
- [16] G., L., 2021. Tu primer Web App con Flask: Parte 3 - DEV Community. A: [en línia]. Disponible a: <https://dev.to/gareisdev/tu-primer-web-app-con-flask-parte-3-1lb>.
- [17] GARCIA, S., 2016. html - Ocultar formulario cuando le doy clic a un botón - Stack Overflow en español. A: [en línia]. Disponible a: <https://es.stackoverflow.com/questions/34081/ocultar-formulario-cuando-le-doy-clic-a-un-botón>.
- [18] IMAD., 2017. jquery - elem.is(':checked') vs elem.prop('checked') - Stack Overflow. A: [en línia]. Disponible a: <https://stackoverflow.com/questions/43464344/elem-ischecked-vs-elem-propchecked>.

- [19] INSTITUT CATLÀ D'ENERGIA.El Recorregut de l'Energia. A: [http://www.gencat.cat/icaen/recorregutenergia/experiencies/Experiencia-2\\_vr.3.3.pdf](http://www.gencat.cat/icaen/recorregutenergia/experiencies/Experiencia-2_vr.3.3.pdf).
- [20] JOBTED.¿Cuánto Cobra un Ingeniero? (Sueldo 2021) | Jobted.es. A: [en línia]. Disponible a: <https://www.jobted.es/salario/ingeniero>.
- [21] LÍNEA DE CODIGO., 2013. Utilizar jQuery each para recorrer una lista o array - Línea de Código. A: [en línia]. Disponible a: <http://lineadecodigo.com/jquery/utilizar-jquery-each-para-recorrer-una-lista-o-array/>.
- [22] LIVING-SUN.Centrar el texto en la entrada del número html - html, css, html5, input. A: [en línia]. Disponible a: <https://living-sun.com/es/html/303575-center-text-in-html-number-input-html-css-html5-input.html>.
- [23] LOPEZ, J., 2018. Blog · Josetxu.com | Aplicar estilos CSS a campos [radio] y [checkbox]. A: [en línia]. Disponible a: <https://josetxu.com/blog/aplicar-estilos-css-a-campos-radio-y-checkbox/>.
- [24] MALDONADO, E.J., 2016. jquery - Ocultar y mostrar un div al hacer click en un radiobutton con javascript - Stack Overflow en español. A: [en línia]. Disponible a: <https://es.stackoverflow.com/questions/6649/ocultar-y-mostrar-un-div-al-hacer-click-en-un-radiobutton-con-javascript/6653>.
- [25] MANEJANDO DATOS.Plantillas de HTML5 - Manejando datos. A: [en línia]. Disponible a: <https://www.manejandodatos.es/2013/07/plantillas-de-html5/>.
- [26] MATTRUMA., 2020. ¿Cómo borrar todos los elementos seleccionados en una entrada SELECT usando jQuery? - Excelente biblioteca. A: [en línia]. Disponible a: <http://es.uwenku.com/question/p-ncthvady-y.html>.
- [27] MDN WEB DOCS., 2021. document.getElementById - Referencia de la API Web | MDN. A: [en línia]. Disponible a: <https://developer.mozilla.org/es/docs/Web/API/Document/getElementById>.
- [28] MDN WEB DOCS.HTMLFormElement.reset() - Referencia de la API Web | MDN. A: [en línia]. Disponible a: <https://developer.mozilla.org/es/docs/Web/API/HTMLFormElement/reset>.

- [29] MONTERO JAVIER., 2012. Python: Agregar y eliminar elementos de un diccionario | El Club del Autodidacta. A: [en línia]. Disponible a: <http://elclubdelautodidacta.es/wp/2012/08/python-agregar-y-eliminar-elementos-de-un-diccionario/>.
- [30] MOULANA, A., 2019. JavaScript : What happens with the browser ? - DEV Community. A: [en línia]. Disponible a: [https://dev.to/azkar\\_moulana/javascript-what-happens-with-the-browser-3cdn](https://dev.to/azkar_moulana/javascript-what-happens-with-the-browser-3cdn).
- [31] MP1990., 2014. html - An invalid form control with name="" is not focusable - Stack Overflow. A: [en línia]. Disponible a: <https://stackoverflow.com/questions/22148080/an-invalid-form-control-with-name-is-not-focusable>.
- [32] OCHO, B.D., 2020. Cómo instalar Visual Studio Code en Linux - ochobitshacenunbyte. A: [en línia]. Disponible a: <https://www.ochobitshacenunbyte.com/2020/05/21/como-instalar-visual-studio-code-en-linux/>.
- [33] OPEN JS WORLD.Download jQuery | jQuery. A: [en línia]. Disponible a: <https://jquery.com/download/#downloading-jquery>.
- [34] PANDAS.DataFrame — pandas 1.2.4 documentation. A: [en línia]. Disponible a: <https://pandas.pydata.org/pandas-docs/stable/reference/frame.html>.
- [35] RIVAS, M., 2012. Efectos de la Globalización en las economías de escala. by Michelle Rivas. A: [en línia]. Disponible a: <https://prezi.com/dl9urnxk504e/efectos-de-la-globalizacion-en-las-economias-de-escala/>.
- [36] ROBERTS, C.Roboto - Google Fonts. A: [en línia]. Disponible a: <https://fonts.google.com/specimen/Roboto>.
- [37] ROLANDOCALDAS.CSS3 básico parte 1 - PHP paso a paso - rolandocaldas.com. A: [en línia]. Disponible a: <https://rolandocaldas.com/php/css3-basico-1-php-paso-a-paso>.
- [38] SALCEDO, L.Construcción de una API de predicción con Flask, Inteligencia Artificial y Python - Mi Diario Python. A: [en línia]. Disponible a: <https://pythondiario.com/2018/08/construccion-de-una-api-de-prediccion.html>.

- [39] SOFT CATALÀ CORRECTOR. Corrector ortogràfic i gramatical català | Corrector valencià. A: [en línia]. Disponible a: <https://www.softcatala.org/corrector/>.
- [40] TEOREMA. JavaScript, HTML and CSS Web Development (workshop) - Teorema. A: [en línia]. Disponible a: <https://teorema-rd.com/producto/workshop-javascript-html-and-css-web-development>.
- [41] TUTORIALFOR., 2020. python - pandas cannot be imported. A: [en línia]. Disponible a: <https://www.tutorialfor.com/questions-311017.htm>.
- [42] UDDIN, N., 2020. Social Network for Programmers and Developers. A: [en línia]. Disponible a: <https://morioh.com/p/3b2d2eeb23f3>.
- [43] UNIPYTHON., 2018. Solicitudes HTTP en Python con Requests - ▷ Cursos de Programación de 0 a Experto © Garantizados. A: [en línia]. Disponible a: <https://unipython.com/solicitudes-http-en-python-con-requests/>.
- [44] URIBE GOMEZ, G., 2018. html - Cambiar value de un input con jQuery o Javascript - Stack Overflow en español. A: [en línia]. Disponible a: <https://es.stackoverflow.com/questions/153476/cambiar-value-de-un-input-con-jquery-o-javascript>.
- [45] W3SCHOOLS. W3Schools Online Web Tutorials. A: [en línia]. Disponible a: <https://www.w3schools.com/default.asp>.
- [46] WIKIPEDIA. Proyecto Jupyter - Wikipedia, la enciclopedia libre. A: [en línia]. Disponible a: [https://es.wikipedia.org/wiki/Proyecto\\_Jupyter](https://es.wikipedia.org/wiki/Proyecto_Jupyter).
- [47] WIKIPEDIA. Globalització - Viquipèdia, l'enciclopèdia lliure. A: [en línia]. Disponible a: <https://ca.wikipedia.org/wiki/Globalització>.