

Treball de Fi de Grau
Grau en Enginyeria en Tecnologies Industrials (GETI)

**Aplicació per l'anàlisi de dades
acadèmiques de l'ETSEIB**

MEMÒRIA

1 de juliol de 2022

Autor: Anna Sagués i Mollà
Directors: Lluís Talavera Mendez
Convocatòria: 07/2022



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resum

En aquest treball es respon a la problemàtica: De quina manera es pot presentar l'anàlisi d'un conjunt de dades? L'estudi agafa les dades dels estudiants de l'escola com a exemple i en fa una anàlisi. A més, es busca la millor interfície per a presentar-lo i se'n crea un exemple.

Al llarg del treball, s'arriba a la conclusió que la millor eina per a realitzar-ho és una pàgina web tipus dashboard. En aquest, es presenta una anàlisi de gènere de les dades dels estudiants del grau.

S'ha creat una pàgina web tipus dashboard on es presenta una anàlisi evolutiva, anual, i departamental de les dades obtingudes. Aquesta anàlisi de gènere busca ser una eina que faci arribar la informació als usuaris i faciliti la presa de decisions. Gràcies a aquest exemple s'ha mostrat la utilitat i versatilitat dels dashboards.

Resumen

En este trabajo se responde a la problemática: De qué manera se puede presentar el análisis de un conjunto de datos? El estudio toma los datos de los estudiantes de la escuela como ejemplo y realiza un análisis. Además, se busca la mejor interfície para presentarlo y se crea un ejemplo.

A lo largo del trabajo, se llega a la conclusión que la mejor herramienta para efectuarlo es una página web tipo dashboard. En este, se presenta un análisis de los datos de los estudiantes del grado.

Se ha generado, pues, una página web tipo dashboard donde se presenta un análisis evolutivo, anual y departamental de los datos obtenidos. Este análisis de género busca ser una herramienta que haga llegar la información a los usuarios y facilite la toma de decisiones. Gracias a este ejemplo se ha podido demostrar la utilidad y versatilidad de los dashboards.

Summary

This project answers the question: How can you present the analysis of a data set? The study uses the data of the school's alumni as an example and analyses it. Moreover, it studies the best interface to present it and builds an example.

Throughout this project, the conclusion reached exposes that the best tool to generate this is a Dashboard web app. In the dashboard, the data presented is analysed from a gender perspective.

The Dashboard website has been created as to show an evolution analysis, as well as a departmental and annual analysis of the data. This gender analysis tries to be a tool that gets the information to all the users and helping in the decision-making process. Thanks to this example, the usefulness and versatility of Dashboards has been proved.

Índex

1	Glosari	7
2	Prefaci	9
2.1	Motivació	9
2.2	Requeriments previs	9
3	Introducció	11
3.1	Ciència de Dades	11
3.2	Context	12
3.3	Objectius del projecte	13
3.3.1	Objectius generals	13
3.3.2	Objectius Personals	14
3.4	Abast	14
4	Marc Teòric	15
4.1	EDA: Exploratory Data Analysis	15
4.2	Arquitectura d'una aplicació web	19
4.2.1	Components	20
4.2.2	Models i tipus de Web App	21
4.2.3	Llenguatges de programació	21
4.3	Eines de visualització de dades: <i>Dashboards</i>	23
4.3.1	Definició	23
4.3.2	Característiques	24
4.3.3	Components	25
4.3.4	Criteris	25
5	Disseny i anàlisi de dades	27
5.1	Estructura plantejada	27
5.1.1	Pantalla Evolució	27
5.1.2	Pantalla Anual	27
5.1.3	Pantalla Departaments	28
5.2	Pre-Procés	28
5.2.1	Fitxer d'informació	28
5.2.2	Dades preinscripció	29
5.2.3	Dades Grau	30
5.3	Unitats d'informació a representar	32
5.3.1	Evolució	32
5.3.2	Anual	32
5.3.3	Departament	34
6	Desenvolupament	37
6.1	Back end	37
6.1.1	Programari	38
6.1.2	Explicació del codi	38

6.2	Front end	44
6.2.1	Programari	44
6.2.2	Explicació del codi	44
7	Impacte en l'entorn	57
7.1	Impacte social i d'igualtat de gènere	57
7.2	Estudi ambiental	57
8	Pressupost	59
8.1	Cost de personal	59
8.2	Cost de llicències	59
8.3	Cost de material	59
8.4	Cost energètic	59
8.5	Cost total associat	59
9	Planificació	61
	Conclusions	63
10	Propostes de futur	65
	Agraïments	67
A	Annex	71
A.1	Codi Python	71
A.1.1	Fitxer views.py	71
A.1.2	Fitxer __init.py__:	71
A.1.3	Fitxer run.py	71
A.1.4	Fitxer genere.py	71
A.1.5	Gràfiques estàtiques	73
A.1.6	Fitxer genere_anyoutput.py	73
A.2	Codi HTML	75
A.2.1	Fitxer 2016.html	75
A.2.2	Fitxer departament.html	77

Índex de figures

1	Esquema de les parts del Data Science	12
2	L'EDA dins el Data Science	16
3	Exemple d'un diagrama de barres agrupat	18
4	Exemple d'un Box Plot	18
5	Exemple d'un pie chart tipus <i>donut</i>	19
6	Exemple d'un Timeplot on l'eix x representa els anys	19
7	Diagrama del funcionament de l'arquitectura d'una aplicació web	20
8	Exemple d'un Dashboard	24
9	BoxPlot de les notes definitives dels estudiants	31

10	Croquis de la pàgina evolució del dashboard	33
11	Croquis de la pàgina anual del dashboard	35
12	Croquis de la pàgina departament del dashboard	36
13	Evolució del nombre de matriculats en funció del temps	39
14	Evolució de les mitjanes al llarg del temps	39
15	PieChart de la taxa d'èxit	40
16	Histograma agrupat per gènere de les diferents assignatures del departament	41
17	Figura tipus indicador de velocitat il·lustrant les mitjanes de les dones	43
18	Element menú tipus cascada de la pàgina anual	46
19	Element mostrant la taxa de matriculats anual	46
20	Element mostrant la taxa d'èxit anual	47
21	Element mostrant la informació per franges de cada any	47
22	Barra lliscant i gràfic de barres de la pàgina anual	48
23	Element representant les mitjanes anuals dels estudiants	49
24	Taula d'informació per departament	50
25	Dashboard Anual	51
26	Dashboard Departament	53
27	Dashboard Departament	55
28	Diagrama de Gant del projecte.	62

Índex de taules

1	Taula de la informació dels departaments a la pàgina anual	41
2	Cost total associat al projecte.	60

1 Glosari

- **Drop-down:** De l'anglès, desplegable. Fa referència a un menú o llista d'opcions desplegable que apareix en una pantalla d'on es pot seleccionar un element. [15]
- **File:** De l'anglès, fitxer. En CS fa referència a informació guardada en un ordinador en una sola unitat amb un nom assignat. [16]
- **Framework:** De l'anglès, marc de treball. Fa referència a les idees, informació, principis o estructures d'un pla organitzatiu. [17]
- **Input:** De l'anglès, entrada. En les ciències de la computació (CS, en endavant) fa referència a tot el que es dona a l'ordinador, és a dir, quan un ordinador o similar rep una senyal de qualsevol tipus, aquest s'anomena *input*. [18]
- **Loop:** De l'anglès, bucle. En les CS és una serie d'instruccions que es repeteix de manera continuada. [19]
- **Output:** De l'anglès, sortida. En les CS fa referència a tot el que dona l'ordinador. Quan un ordinador general algun senyal o comanda, s'anomena *output*. [20]
- **Scatterplot:** De l'anglès, gràfic de disperssió. Es una gràfica que mostra la relació entre dos valors numèrics, on cada punt representa una observació. [9]
- **Script:** De l'anglès. En les CS, fa referència a un programa o serie d'instruccions dut a terme per un altre programa, en comptes de ser compilat pel propi ordinador. [21]
- **Slider:** De l'anglès, control lliscant. Fa referència a una barra que es mou horitzontalment per controlar una variable, per exemple el volum. [22]
- **Static:** De l'anglès, estàtic. En les CS es refereix a elements son constants i no varien sense intervenció del propi programador. [23]
- **Web Server:** De l'anglès, servidor web. Consisteix en un sistema o programa que envia informació o pàgines web als usuaris d'Internet. [24]
- **Widget:** De l'anglès, giny. En CS fa referència a una part del software d'una web que interactua amb l'usuari en una petita part de la pantalla general. [25]

2 Prefaci

2.1 Motivació

Al llarg dels quatre anys de carrera he pogut descobrir diverses àrees de l'enginyeria. Totes m'han ensenyat conceptes que m'han interessat de valent. Però l'àrea que ha despertat un interès especial en mi ha estat el sector de la programació. A primer, vaig poder endinsar-me en el món de la programació amb l'assignatura de Fonaments d'Informàtica descobrint el llenguatge de Python. En els cursos següents, vaig poder profunditzar els meus coneixements de Python així com descobrir nous llenguatges de programació i el poder de cadascun.

Al segon curs, em vaig apuntar al projecte d'escola *ARM2u*, on se'm va assignar la tasca de programació web. Allà vaig tenir la primera oportunitat d'aprendre a programar de manera autònoma. El quadrimestre següent, a l'assignatura de Projecte 1 vaig poder treballar en el món d'anàlisi de dades i disseny d'aplicacions. Vaig poder complementar el meu coneixement en programació amb el de l'anàlisi de dades, i veure l'aplicabilitat d'aquest sector en l'enginyeria, així com totes les portes que obre.

Quan vaig tenir la oportunitat de triar treball de final de grau, sabia que el volia fer en aquest departament. En Lluís Talavera em va proposar un treball que combinava programació web amb anàlisi de dades, i que a més era una gran oportunitat per a complementar la meva formació.

2.2 Requeriments previs

El principal requeriment previ per a la realització d'aquest treball, és el coneixement previ adquirit durant el grau. Una bona base de coneixements de programació és clau per a poder desenvolupar el projecte. Ser capaç d'aprendre i resoldre dubtes de manera autònoma és bàsic per a qualsevol tasca de programació.

3 Introducció

3.1 Ciència de Dades

El *Data Science* o ciència de dades és un camp multidisciplinari que engloba el tractament de grans quantitats de dades [7]. Mitjançant mètodes científics, processos, algorismes i sistemes, extreu informació de grups de dades. El *Data Science* unifica l'estudi estadístic, informàtic i l'anàlisi de dades per a poder entendre i analitzar esdeveniments a partir de dades. Tot i que aquest concepte és molt ampli, i en podem trobar moltes definicions, es pot resumir com a un camp que engloba la preparació de les dades per a la posterior anàlisi i processament, l'anàlisi avançat de les dades i la presentació de resultats que permetin extreure conclusions. Termes com *Data Mining*, *Big Data* i *Machine Learning* estan estretament lligats al *Data Science*.

El *Data Mining* o KDD (de l'anglès *Knowledge Discovery in Data*) és també un camp interdisciplinari contingut dins els camps d'estadística i ciències de computació [6]. Aquest mètode consisteix en un procés d'extracció que permet descobrir patrons en grans conjunts de dades. El *Data Mining* permet doncs trobar relacions entre les dades prèviament desconegudes, com per exemple dependències o grups de dades. Aquest procés té per objectiu dos camps bàsics. Pot descriure les dades proporcionades o bé pot generar prediccions mitjançant el *machine learning*. Un exemple on s'utilitza el data mining, és en detecció de frau.

El *Machine Learning* consisteix en l'estudi d'algorismes per a computació que aprenen automàticament i a base d'experiència gràcies a l'ús de dades [8]. És també un camp multidisciplinari, derivat de la intel·ligència artificial i les ciències de la computació. Aquest mètode fa servir "dades d'entrenament" per a construir un model que permet fer prediccions o classificacions de manera autònoma. Alguns exemples d'ús del *Machine Learning* són el reconeixement de veu, l'anàlisi d'imatges digitals i altres camps on generar un algorisme convencional seria d'alta dificultat, o fins i tot impossible.

El *Big Data* és un camp que analitza, processa sistemàticament o tracta grans conjunts de dades [5]. S'empra el *Big Data* quan aquests conjunts tenen una mida massa gran o són massa complexes per a ser tractats amb mètodes convencionals.

Aquest mètode és conegut per tres conceptes claus: Velocitat, Volum i Varietat. El volum fa referència a la quantitat de dades a processar, que pot arribar a grans quantitats, sovint de mida desconeguda com per exemple en l'anàlisi de dades d'una xarxa social. La velocitat mesura el valor al qual es rep i tracten les dades. La varietat correspon a quants diferents tipus de dades hi ha disponibles. Les dades poden estar estructurades, és a dir ordenades, per l'empresa per exemple. En aquest cas és més senzill conèixer la variabilitat de la mostra. Però les dades també poden estar desestructurades, és a dir, informació desorganitzada i que no pertany a cap categoria en concret, com per exemple el contingut obtingut d'una xarxa social.

Recentment, s'han afegit dos nous conceptes que són: Veracitat i Valor. Les dades tenen valor intrínsec que s'ha d'explorar. A més a més, cal comprovar la veracitat d'aquestes

i si s'hi pot confiar.

El procés de *Data Science* [7] és complex, però es pot dividir en 5 grans parts, com s'observa a la figura 1

- **Obtenció:** Procés en el qual s'assoleixen dades estructurades o desestructurades. Es poden recaptar manualment o bé aconseguir-les de sistemes reals.
- **Manteniment:** Neteja, preparació, processament i arquitectura de les dades per ser tractades.
- **Processament:** Classificació, models, resums i hipòtesis sobre les relacions entre variables. És també on es decideix quin mètode d'anàlisi és més adient.
- **Anàlisi:** Exploració de les dades, confirmació de les hipòtesis, i anàlisi qualitativa o predictiu. Es treballa també amb *machine learning* i regressions.
- **Comunicació:** Visualització de les dades, presentació i presa de decisions. Els programadors fan servir eines de visualització com gràfiques, variables representatives o informes.

El procés és iteratiu i un cop comunicades les dades, es poden tornar a obtenir de noves per a millorar el procés.

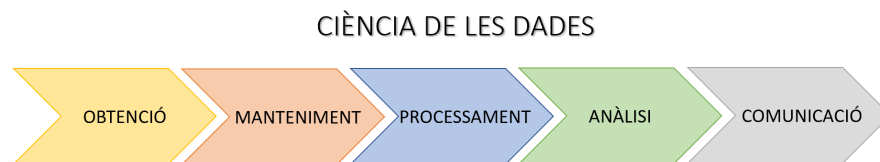


Figura 1: Esquema de les parts del Data Science

Un exemple que ajuda a entendre els conceptes és el següent. Una empresa de tecnologia digital de mèdia ha creat una plataforma d'anàlisi de l'audiència que permet als clients veure la reacció dels espectadors als diferents continguts que se'ls ofereixen. La solució fa servir mètodes com el *Machine Learning* per a recollir informació en temps real de l'audiència.

3.2 Context

Per tal de posar en pràctica la teoria mencionada anteriorment, s'han adquirit les dades dels estudiants del Grau en Enginyeria de Tecnologies Industrials (GETI, en endavant) de l'Escola Tècnica Superior d'Enginyeria Industrial de Barcelona (ETSEIB, en endavant).

Aquest projecte estudiarà doncs les dades de l'ETSEIB aplicant els conceptes del *Data*

Science. Per tal de fer-ho es vol programar una aplicació web que permeti veure aquesta anàlisi. Es vol, que la web sigui fàcil d'entre, ràpida, intuïtiva i actualitzable. S'opta doncs per a la realització d'un Dashboard que plasmi tots els punts que es considerin necessaris. Un Dashboard és una eina que permet visualitzar dades. El principal objectiu d'un Dashboard és plasmar la informació a primera vista. Més endavant s'entra en detall tant de les pàgines web com dels Dashboards.

Per tal de decidir quina perspectiva d'anàlisi agafar, s'ha fet un petit estudi de documentació.

L'estudi estadístic del *Society of Women Engineers* el novembre del 2019 que el 26% dels treballadors en ciència de la computació són dones. En l'àmbit global, pública que tan sols 13% dels enginyers són dones [29]. Es cert que el mateix estudi descriu una evolució positiva d'aquests nombres, on el nombre de titulacions en els camps d'enginyeria i ciència de la computació per a dones ha augmentat en 58% entre el 2012 i el 2017. No obstant, aquests nombres encara són alarmants, i indiquen que la igualtat d'oportunitats a les carreres tecnològiques és un objectiu encara per assolir.

Sense anar més lluny, les dades estadístiques de l'ETSEIB publiquen que el percentatge de noves matriculacions d'alumnes dones al GETI, entre el 2017 i el 2022 s'ha mantingut estable en un marge d'entre el 22 i 24% [2].

Com es pot veure, doncs, una anàlisi de gènere de les dades de l'Escola és necessari i pertinent. La diferència entre homes i dones en el món STEM segueix vigent i es pot percebre fins i tot en el nombre de matriculats del grau. Tenint accés a les dades dels alumnes, es podria realitzar un estudi de varis paràmetres per tal de visualitzar les diferències acadèmiques entre sexes a l'escola. Presentar aquest anàlisi a l'equip directiu pot ajudar en la presa de decisions. Visualitzar les dades analitzades pot mostrar si existeix o no un problema, i ser d'ajuda en una possible solució.

S'ha decidit doncs analitzar-les amb perspectiva de gènere per a poder observar aquestes diferències i facilitar a l'equip directiu la presa de decisions sobre un pla d'acció.

3.3 Objectius del projecte

Per a tal d'enfocar aquest treball cal plantejar una sèrie d'objectius per a assolir. Es buscarà que siguin clars i assequibles. Per tant, els objectius que s'aconseguiran al llarg d'aquest treball queden explicats a continuació.

3.3.1 Objectius generals

L'objectiu principal del projecte és desenvolupar una aplicació que permeti als usuaris carregar, processar i visualitzar dades acadèmiques dels estudiants de l'ETSEIB. Es cobriran les fases de filtratge i transformació de dades, tractament mitjançant eines estadístiques i disseny d'una interfície intuïtiva que inclogui diversos tipus de gràfics per facilitar l'exploració de les dades. A continuació es desglossen en petits objectius:

- **Anàlisi de dades**
 - Filtrat de dades, en funció de les necessitats i requeriments de l'usuari
 - Preprocés de les dades i càlculs preliminars
 - Desenvolupament d'eines de visualització de dades com gràfiques, taules o indicadors de rendiment
- **Estudi del cas concret d'anàlisi de les dades de l'ETSEIB amb perspectiva de gènere**
- **Disseny d'una eina que permeti visualitzar de manera eficaç i eficient les dades analitzades**
 - Estudi de les millors eines per a representar dades
 - Estudi de les eines per plasmar elements de visualització de dades
 - Programació d'una interfície per a l'anàlisi i presa de decisions
 - Facilitar la presa de decisions als usuaris

3.3.2 Objectius Personals

- **Ampliar els coneixements en programació:**
 - Complementar la formació rebuda al grau amb noves llibreries de programació
 - Estudiar nous llenguatges de programació per tal d'assolir els objectius marcats

3.4 Abast

Tal com s'ha mencionat anteriorment, aquest projecte busca desenvolupar una eina de visualització de dades que permeti una anàlisi ràpid de la diferència de gènere a l'ETSEIB. Per tant, es busca analitzar les dades existents per tal de facilitar la presa de decisions i avanç en l'assoliment de la igualtat d'oportunitats en el món de l'enginyeria.

Queda doncs fora de l'abast la predicció de comportaments. De la mateixa manera, queda fora de l'abast la presa de decisions en si, doncs l'aplicació busca ser l'eina que faciliti aquestes, i no una eina decisiva.

4 Marc Teòric

Com s'ha mencionat en introducció, aquest treball es centra en l'anàlisi de dades i creació de programari. En aquest apartat, s'entra en detall de diversos conceptes d'aquests temes.

4.1 EDA: Exploratory Data Analysis

L'anàlisi exploratòria de dades, o *Exploratory Data Analysis (EDA)* engloba l'estudi analític d'un conjunt de dades [1]. Per a realitzar-lo es disposa d'eines gràfiques i models de visualització de dades. Aquesta metodologia consisteix a analitzar les dades independentment de possibles hipòtesis que es tinguin. És gràcies a aquesta anàlisi inicial que es trobaran correlacions i hipòtesis sobre les dades. [28]

El matemàtic John Tukey va escriure el llibre *Exploratory Data Analysis* [14]. En ell, defèn que cal una exploració de les dades prèvies a la formulació d'hipòtesis. Tukey defineix els objectius de l'EDA com:

- Creació d'hipòtesis sobre les causes dels fenòmens observats.
- Avaluar les hipòtesis en les quals es basaran les conclusions.
- Triar les eines adequades per visualitzar les dades.
- Crear una base per a obtenir més dades .

El procés de l'EDA es pot resumir en un diagrama de flux (Figura 2, on es veu la posició que ocupa aquest procés en el conjunt del Data Science.

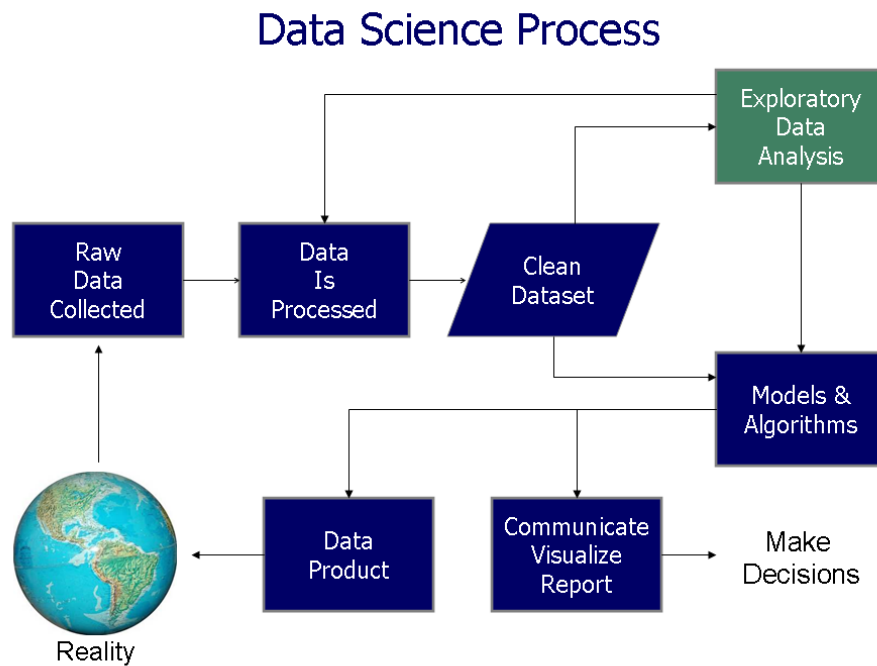


Figura 2: L'EDA dins el Data Science

A l'hora d'analitzar les dades, cal seguir uns passos per assegurar-se que el conjunt que es tracta és avaluable. Cal comprovar que no falten dades, i en cas que faltessin s'haurien de completar per tal de fer el conjunt llegible. Un cop comprovat l'estat de les dades, es pot passar a analitzar el conjunt mitjançant eines gràfiques o estadística descriptiva. Les eines d'anàlisi descriptiva es poden separar en anàlisi invariant i anàlisi bivariant.

L'anàlisi invariant descriu la distribució d'una única variable. Per a fer-ho, existeixen diversos paràmetres que són d'utilitat.

En primer lloc, tenim la mitjana aritmètica, a no confondre amb la mediana o la mitjana geomètrica. Aquesta mitjana es calcula com:

$$\bar{x} = \frac{1}{n} \cdot \sum_{i=1}^n x_i \quad (1)$$

D'altra banda, la mediana és el valor que separa la meitat superior i la meitat inferior d'un conjunt numèric. Aquest valor es calcula de maneres diferents segons si el conjunt té un número parell o senar d'elements:

$$n \text{ senar} : \text{mediana}(x) = x_{n+1/2} \quad (2)$$

$$n \text{ parell} : \text{mediana}(x) = \frac{x_{n/2} + x_{(n/2)+1}}{2} \quad (3)$$

El rang és la diferència entre el valor superior i inferior d'un conjunt de dades. És útil per a mostrar la dispersió del conjunt.

Els quartils són un tipus de quantil que divideixen el conjunt de dades en quatre parts igual. Així com un quantil pot dividir en un número qualsevol, dividir en quartils assegura que un 25% de les dades es troben en cada divisió.

La desviació estàndard és una mesura de la dispersió d'un set de dades. Un valor baix indica que les dades es troben a prop de la mitjana, mentre que un valor elevat indica el contrari. Es calcula com:

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}. \quad (4)$$

Finalment, la variància és el quadrat de la desviació estàndard. De manera anàloga, aquest paràmetre mesura la dispersió d'un conjunt de dades. Es calcula com:

$$\sigma^2 = \frac{\sum_{i=1}^n (x_i - \mu)^2}{n} \quad (5)$$

L'anàlisi bivariant es realitza quan la mostra conté més d'una variable. Aquesta anàlisi sovint s'utilitza per a fer comparacions entre variables. Es poden fer servir mesures de dependència, *scatterplots* o taules de múltiples entrades d'anàlisis multivariants.

Les eines que es fan servir a l'hora de visualitzar les dades, consisteixen en diverses gràfiques. Alguns exemples d'aquests són:

- **Diagrama de barres:** Els diagrames de barres representen dades numèriques en format de barres. Aquestes, tenen una alçada proporcional a la dada que representen. Poden ser tant verticals com horitzontals. A més a més, es poden categoritzar per grups per tal de facilitar-ne la comparació. S'anomenen llavors *stacked* o bé *grouped* segons si s'apilen o es posen costat a costat, com es pot veure a la Figura 3

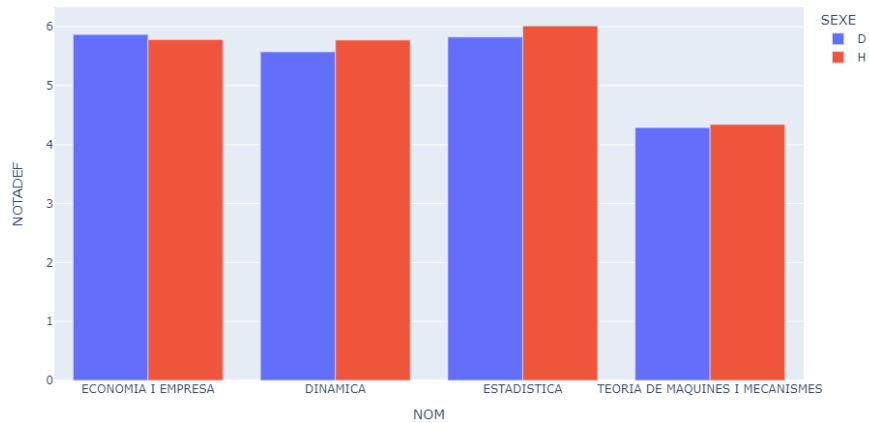


Figura 3: Exemple d'un diagrama de barres agrupat

- **Box Plot:** Diagrama que indica la mitjana, els quartils i el màxim i mínim d'un paquet de dades. (Figura 4).

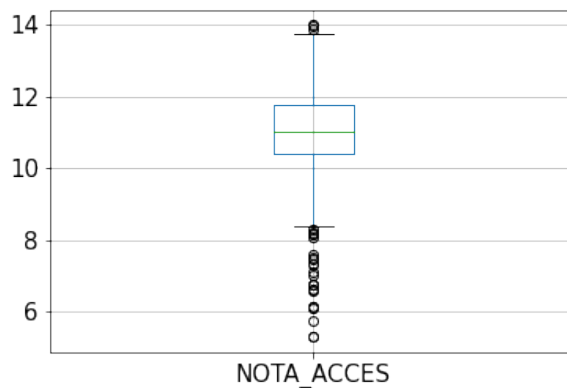


Figura 4: Exemple d'un Box Plot

- **Pie Chart:** Permet representar les dades en percentatge. És útil per a comparar ràpidament dos valors. (Figura 5).

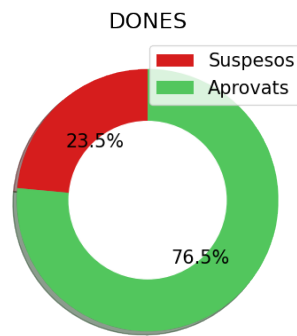


Figura 5: Exemple d'un pie chart tipus *donut*

- **Time Plot:** És una gràfica de línia on l'eix x representa el temps. S'empra per a visualització evolucions temporals o variacions en el temps. (Figura 6).

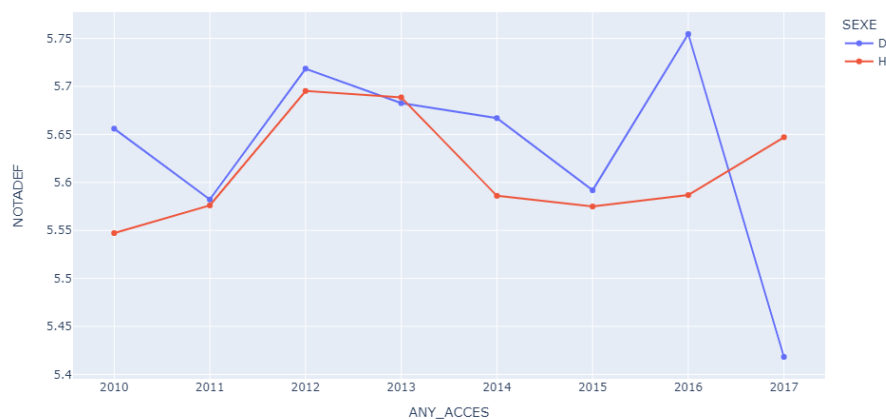


Figura 6: Exemple d'un Timeplot on l'eix x representa els anys

4.2 Arquitectura d'una aplicació web

L'arquitectura d'una aplicació és l'esquelet de totes les connexions entre aplicacions, bases de dades i sistemes auxiliars que fa que funcionin correctament. De manera pràctica, descriu les interaccions des del moment que l'usuari prem el *link* de l'aplicació web. Això genera que el servidor web enviï les dades al navegador per a obtenir accés a la web demanada.

Per exemple, si escrivim *etseib.upc.edu* al buscador d'un navegador, això reconeix l'adreça i sol·licita accés als servidors de l'escola. El servidor reconeix la sol·licitud i l'envia a la base de dades, que de la mateixa manera sol·licitarà permís per a exposar la informació al navegador. Finalment, l'usuari pot visualitzar la informació demanada.

Típicament, l'arquitectura es divideix en dues parts. El *Client-side Code* o codi enfocat al client, també anomenat *Frontend* i el *Server-side Code* o codi enfocat al servidor, també

conegut com a *Backend*.

El *Frontend* conté tota la interfície que l'usuari veu i amb la que pot interactuar. És la part que respon als *inputs* del usuari.

El *Backend* conté el servidor amb els fitxers del sistema i la base de dades.

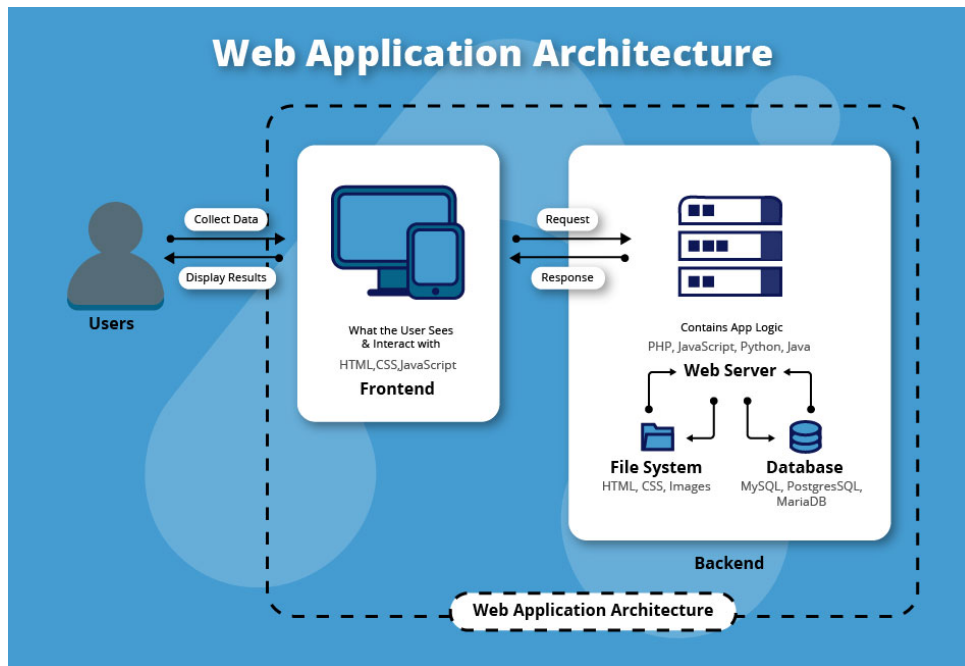


Figura 7: Diagrama del funcionament de l'arquitectura d'una aplicació web

4.2.1 Components

Els components que formen l'arquitectura d'una aplicació web es poden dividir en dues grans parts:

- **Components de la interfície de l'usuari:** conté la part visual de la web. Representa la interfície amb la qual interaccionarà l'usuari. Aquesta part conté notificacions, *dashboards*, *widgets*, i altres components interactius. Aquests components es poden programar amb llenguatges com *Hyper Text Markup Language*, (*HTML en endavant*), *Cascading Style Sheet* (*CSS, en endavant*) o *Java Script* (*JS, en endavant*).
- **Components estructurals de la web:**
 - El navegador web del client, que és el que permet a l'usuari interactuar amb la web.
 - El servidor de l'aplicació web, que conté la lògica de l'aplicació i les diverses capes de l'aplicació.
 - El servidor de la base de dades, que reté, extreu i ofereix dades que manipula el servidor i que són necessàries per al funcionament de l'aplicació.

4.2.2 Models i tipus de Web App

Existeixen diversos models de web app, que poden variar segons el nombre de servidors i bases de dades. A continuació, s'explica en què varia cadascun.

El model d'un servidor i una base de dades és poc fiable, ja que si el servidor falla, tota la web fallarà. Aquest model només s'empra per a proves o per a entendre el funcionament de les webs, però no per a implementar-ne una.

El model de múltiples servidors i una única base de dades és més fiable, perquè si un servidor falla, un servidor de recanvi pot agafar el relleu. Tanmateix, si falla la base de dades, falla tota l'aplicació.

El model de múltiples servidors i múltiples bases de dades és el més fiable i eficient. En cas de fallada tant de la base de dades com del servidor no fallarà l'aplicació. Es poden tenir dues bases de dades idèntiques o bé repartir les dades en les dues bases.

El tipus d'aplicació web depèn de com està distribuïda la lògica entre la part del client i la del servidor. Es divideixen en 3 tipus:

- **Single Page Applications (SPAs):** Aquest tipus d'arquitectura soluciona el problema d'aplicacions que carreguen lentament. Aquests tipus d'aplicació recarrega la web amb els canvis fets per l'usuari de manera ràpida i dinàmica, ja que sempre presenta la mateixa distribució. Es caracteritza per ser intuïtiva i interactiva.
- **Microservices:** Aquesta arquitectura és poc pesada i realitza petites funcions. Permet crear aplicacions senzilles i ràpid.
- **Serverless Architecture:** Aquest tipus d'arquitectura permet delegar algunes funcions a un tercer, per tal d'evitar que l'aplicació falli en hores on és molt concorreguda.

4.2.3 Llenguatges de programació

Python Per al tractament, anàlisi i creació d'eines de visualització de dades, un llenguatge de programació que és de gran utilitat és Python [27]. La gran flexibilitat i versatilitat d'aquest llenguatge el converteix en una eina idònia per al tractament de dades. Algunes biblioteques de Python de gran utilitat en el context d'anàlisi de dades són:

- **Pandas:** Pandas és una llibreria que permet tractar, visualitzar i modificar dades en forma de *DataFrames* [13]. Els *dataframes* són objectes de tipus taules, amb files, columnes i índex. Permeten treballar de manera còmoda amb grans sets de dades. Les diferents funcions integrades de la llibreria ens permeten fer operacions necessàries per al tractament de dades. Notablement, per la part de preprocés, aquesta eina permet detectar valors absents (tipus *NaN*), permet eliminar files o columnes no desitjades o filtrar en funció d'una o més condicions.

- **NumPy:** Numpy [11] és una llibreria que permet tractar, i fer operacions sobre formacions de dades (en anglès, *array*). És una eina útil per a manipular sets de dades, i treballa molt bé en concordança amb la llibreria *Pandas*.
- **Matplotlib:** Aquesta llibreria [12] permet crear gràfiques de visualització de dades com per exemple: histogrames, diagrames de barres o *scatterplots* entre d'altres. La llibreria és d'accés obert. Permet elaborar directament amb *Pandas DataFrames* facilitant les operacions de creació de gràfiques.

HTML Pel que fa a la programació d'una interfície web, un dels llenguatges de programació més populars és el HTML. HTML és un dels llenguatges principals en el *World Wide Web*, (WWW en endavant). Aquest llenguatge permet visualitzar contingut en forma d'aplicació web. Es lliga sovint amb el llenguatge CSS i JS.

Els servidors web reben els documents HTML i projecten com a pàgina web. Aquest llenguatge permet treballar amb diversos tipus d'elements. Inclou la inserció de figures, texts, elements interactius i molt més.

El llenguatge CSS tracta la part de disseny o estil de la web. Permet definir les posicions, colors, mides i altres característiques atribuïdes a la part estètica.

El llenguatge de JS permet generar *scripts* o codis dins el mateix HTML. Aquest llenguatge complementat amb HTML permet generar pàgines web més complexes i que poden adaptar-se millor a l'usuari.

Una llibreria que sovint es fa servir a l'hora de programa *HTML*, és la de *W3-school*. Aquesta conté molts elements ja estilitzats que faciliten la feina als programadors i permeten un codi més net i eficient.

A més a més, aquest llenguatge permet connectar-se amb el *back-end* del servidor web, notablement amb el llenguatge de *Python*.

Flask *Flask* és un marc web escrit en llenguatge *Python* que no requereix eines particulars. Permet afegir extensions i llibreries. [10]. El codi Flask permet generar una pàgina web gràcies a *Python* i *HTML*. L'estructura base que segueix una aplicació Flask és:

```

web
├── run.py
├── app
│   ├── __init__.py
│   └── views.py

```

Tots els fitxers estan continguts dins una carpeta, aquí anomenada *web*. El fitxer `__init__.py` crea la drecera *Flask* i inicialitza el servidor. El fitxer `run.py` arrenca l'aplicació, i crida tots els subfitxers necessaris. El fitxer `views.py` conté totes les dreceres de l'aplicatiu i és el fitxer que connecta l'usuari amb el *back-end* de l'aplicació.

A partir d'aquí, es poden afegir fitxers i carpetes extra si són necessaris.

4.3 Eines de visualització de dades: *Dashboards*

La visualització de dades és essencial per tal d'entendre tendències i patrons en grans bases de dades. Aquestes eines permeten als analistes entendre i visualitzar conceptes de manera fàcil i ràpida.

Avui en dia, és habitual que les empreses tractin amb bases de dades de molt grans dimensions i una eina que permeti entendre-les s'ha tornat essencial. La manera més comú de presentar-les, és gràcies a un panell de dades, o *Dashboard*, en anglès.

4.3.1 Definició

Un *Dashboard* és una eina que permet visualitzar dades. El principal objectiu d'un *Dashboard* és plasmar la informació a primera vista. El format més habitual és d'una sola pantalla que permet a l'usuari rebre tota la informació de cop. La interfície presenta sovint *Key Performance Indicators (KPI, en endavant)*, és a dir indicadors de rendiment claus. Aquests permeten als usuaris prendre decisions ràpid, així com mesurar eficiències i ineficiències entre d'altres. És per això, que el més habitual es fer-ne us en context professional.

Aquestes interfícies s'accedeixen normalment mitjançant llocs webs i tenen la possibilitat d'afegir noves dades per tal d'actualitzar-ne el contingut.

Els *Dashboards* es poden classificar en tres categories: estratègic, analític i operacional.

Els *Dashboards* operacionals es fan servir per a exposar informació actual, que està passant ara. Fan servir dades recents i avaluen els rendiments i KPIs.

D'altra banda, els *Dashboards* analítics es fan servir per a grans quantitats de dades on cal trobar rellevàncies, tendències i fer prediccions. Treballen amb dades fixes, que no requereixen tantes actualitzacions, sinó comparacions amb dades històriques.

Finalment, els *Dashboards* estratègics s'utilitzen per a anàlisis de llarg termini. Permeten analitzar dades per a la presa de decisions. Requereixen actualitzacions però menys freqüents, ja que solen basar-se en terminis mensuals, quadrimestrals o anuals.

Avui en dia, els *Dashboards* son cada cop més populars i permeten substituir els informes. Presenten dinamisme, son personalitzables i interactius. Un exemple de *Dashboard* molt famós és el *Dashboard* creat per la Universitat Johns Hopkins per a visualitzar l'evolució de les dades de la COVID-19 durant la pandèmia mundial del 2020 [26]. Aquest *Dashboard* va fer arribar la informació sobre la pandèmia a tot tipus d'usuaris. Va aconseguir filtrar una base de dades de molt grans dimensions com és les estadístiques de la pandèmia i presentar-la de manera gràfica, intuïtiva i senzilla. (Figura 8)

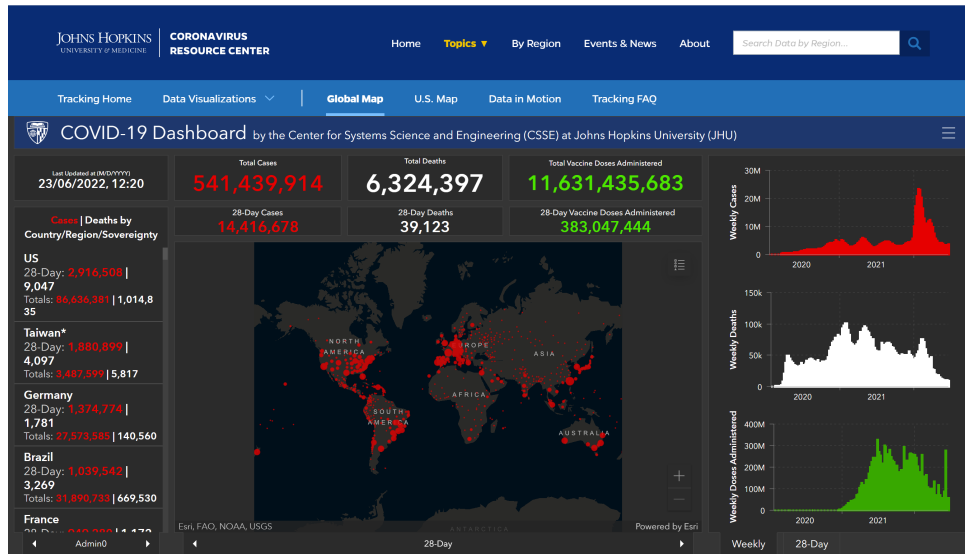


Figura 8: Exemple d'un Dashboard

4.3.2 Característiques

Un bon dashboard ha de ser clar, intuïtiu i personalitzable. Per ser efectiu, cal que mostri informació útil, i pertinent. Tota la informació essencial es pot accedir immediatament. Aquesta està plasmada en una sola pantalla de disseny coherent amb una jerarquia establerta. Un element que juga una part clau en un dashboard és el disseny i en especial les característiques de disseny. Es busca adaptar-se al màxim a l'usuari. Una manera d'aconseguir-ho és plantejar-se una serie d'objectius seguint la política dels objectius SMART (Específics, Mesurables, Assequibles, Realistes i Basats en el Temps). Els objectius d'un dashboard pretenen respondre la problemàtica: Quin problema ens permetrà resoldre aquest disseny? Les característiques es poden resumir com:

- Comuniquen informació ràpid.
- Objectius clars.
- Consistència al llarg de la pàgina.
- Mostren informació de manera clara i eficient.
- Proximitat i coherència entre elements.
- Mostren tendències i canvis en les dades al llarg del temps.
- Son fàcils de personalitzar.
- Posen l'usuari al capdavant.
- Son presentats de manera efectiva.
- Mostren totes les dades en un espai limitat.

- Espais en blanc per no atapeir la pàgina.
- Paleta de colors per donar continuïtat.
- Tipus de lletra fàcil de llegir i constant al llarg de la pàgina.
- Limitar l'ús de text per fer el dashboard més fàcil de llegir.
- Integrar taules i marcs de dades quan hi ha molta informació a mostrar.
- Tria informada i pertinent de les eines de visualització de dades
- Basar el disseny final en la visió del usuari

4.3.3 Components

Els *Dashboards* estan compostats per una serie de components que fan que siguin el més eficients possibles. Aquests son:

- KPI's: Els *Dashboards* tenen indicadors de rendiment clau dels quals en faran un seguiment
- Visualització de dades: Com ja s'ha presentat anteriorment, un dels punts principals d'un *Dashboard* és la pertinent visualització de dades.
- Filtres: Per tal d'assegurar que les dades visualitzades son pertinents, aquestes han de passar per uns filtres que les adequaran a l'usuari.
- Navegació: La pantalla ha de presentar una navegació intuïtiva mitjançant una bona jerarquia.

4.3.4 Criteris

A l'hora de dissenyar el *Dashboard*, es recomanable basar-se en una serie de criteris que permetran un bon disseny. Aquests duren el programador a una llista de presa de decisions sobre el disseny que asseguraran un *Dashboard* que compleixi la funció desitjada. A més a més, apart de seguir els criteris, el disseny ha de vetllar per complir amb les característiques mencionades anteriorment. Els criteris son:

- Entendre l'usuari i marcar un objectiu
- Assegurar simplicitat
- Triar eines de visualització de dades i definir el repartiment dels elements
- Establir jerarquia d'informacions
- Permetre l'evolució del *Dashboard*

És recomanable també seguir l'ordre mencionat, per tal de crear un *Dashboard* fet a mida

per l'usuari.

5 Disseny i anàlisi de dades

Per a poder realitzar aquest anàlisi, s'han obtingut les dades dels estudiants de l'ETSEIB gràcies a la secretaria de l'escola. S'ha proporcionat l'autoria amb les dades corresponents des de l'inic del grau fins al 2019. A continuació s'explica en detall el disseny de la interfície web així com el contingut dels fitxers de programació

5.1 Estructura plantejada

Per tal de assolir els objectius establerts en apartats anteriors d'aquest treball, s'ha decidit que l'aplicació web que millor s'adapta als requisits és un *Dashboard*. Per tal de tractar la base de dades de la millor manera, s'ha trobat pertinent analitzar les dades des de un punt de vista temporal, separant la part d'evolució i la part d'anàlisi anual. A més a més, es vol fer també un anàlisi per departament, que tindrà en compte les dades històriques. L'usuari d'aquesta aplicació és principalment el departament encarregat de prendre decisions envers la política de gènere. En aquest cas, va dirigida al equip directiu de l'ETSEIB. No obstant, pot ser de gran interès que aquestes dades es facin públiques i puguin arribar no només al equip directiu i altres membres de l'escola, sinó a agents externs a la UPC.

Com s'ha vist en apartats anteriors, és important que un *Dahsboard* presenti tota la informació en una sola pantalla tot mantenint claredat i sense crear pàgines atapeïdes. És per això que s'han realitzat tres pantalles, una dedicada a l'evolució, una a l'anàlisi anual, i una als departaments.

Per tal de mantenir l'ordre i el caire intuïtiu del *Dashboard*, es busca una línia estètica entre les tres pantalles que generi una navegació senzilla i mantenir la uniformitat de l'aplicació web.

Per poder procedir a un disseny més acurat, cal definir en quina categoria cauen cada una de les pantalles del *Dashboard*.

5.1.1 Pantalla Evolució

L'objectiu d'aquesta pantalla és realitzar una anàlisi temporal de les dades proporcionades. Es busca analitzar com les diferències de gènere al grau han evolucionat. Per aconseguir-ho caldrà analitzar varis paràmetres temporals com indicadors de rendiment clau. S'espera que aquesta pantalla permeti veure si hi ha una clara diferència respecte el gènere i en cas afirmatiu, l'evolució d'aquesta. A més a més, pot permetre veure com han afectat les polítiques de gènere implementades per l'escola com per exemple el Pla d'Igualtat de la UPC (2016-2022) [3].

5.1.2 Pantalla Anual

Aquesta pantalla realitzarà un anàlisi estàtic de els dades any per any. Té per objectiu buscar relacions i rellevàncies de les dades, per tal d'encaminar els usuaris a la presa de decisions. Aquesta pantalla no requereix actualitzacions més enllà d'introduir les

dades d'un nou any si se'n vol fer l'anàlisi. Per a aquest *dashboard* és important triar bones eines que siguin pertinents a la presa de decisions. Es volen trobar relacions, tendències i rellevàncies que permetin fer aquesta presa de decisions el més informada possible.

5.1.3 Pantalla Departaments

El *dashboard* per departaments té per objectiu analitzar les dades en base al departament, deixant de banda el caire temporal. Aquí no es buscarà veure una evolució, sinó que es treballa amb l'històric de dades. Caldrà actualitzar-lo anualment amb les noves dades, o bé quadrimestralment, si s'escau. Aquesta pantalla vol estar enfocada a que cada departament pugui entendre les seves dades i prendre les decisions en base a aquesta informació. A més a més, pel que fa a l'equip encarregat de prendre decisions globals, pot permetre veure si existeix un problema concret o es tracta d'un tema global.

Aquests tres *Dashboards* es poden categoritzar com a *Dashboards* estratègics ja que analitzen dades a llarg termini per a la presa de decisions. No requereix una actualització constant, i estan dissenyats per ser consultats poc habitualment, només quan requereixi una presa de decisions.

5.2 Pre-Procés

A l'hora de programar un *dashboard* on alguna pàgina web, interessa dividir els processos de *back-end* entre preprocés i en viu. Tot el que es preprocessi, agilitza després la càrrega de la pàgina web i permet una navegació més ràpida. En aquest exemple, s'han decidit preprocessar diverses funcions, que s'expliquen a continuació.

El banc de dades amb el que es comença a treballar consta de fulls de càlcul proporcionats per l'escola. L'etapa d'obtenció ha estat doncs el procés d'aconseguir les dades gràcies a l'equip directiu. La informació disponible consta de tres fitxers diferents. En primer lloc, tenim el de les dades dels estudiants de la pre-matricula des de el 2010 fins al 2019. D'altre banda tenim dos fitxers més similars, on hi ha la informació dels resultats obtinguts pels estudiants a cada convocatòria de cada assignatura. Un fitxer correspon a les dades de la fase inicial, i l'altre al de la resta del grau. El tractament de les dades s'ha fet mitjançant el llenguatge de programació *Python*, principalment amb el servidor *Jupyter Notebook*. La llibreria emprada per el tractament ha estat principalment *Pandas*.

El primer pas, comú a tots els processos, és importar el fitxer de tipus *.csv* com a un *pandas DataFrame*. Això permet treballar molt còmodament amb fitxers de dades.

A continuació s'explica l'etapa de manteniment de les dades.

5.2.1 Fitxer d'informació

Per tal de facilitar l'anàlisi de les dades s'ha creat un fitxer auxiliar de dades informatives que poden ser de gran ajuda. Aquest fitxer conté informació sobre les assignatures

obligatòries del GETI. Aquest dataframe s'ha anomenat *assignatures*.

No s'han tingut en compte les assignatures optatives (corresponents al quadrimestre 4 i 8) ni les assignatures de projecte 1 i projecte 2. Les columnes del dataframe corresponent son:

- **NOM:** És una columna de text, que conté el nom de cada assignatura
- **CODIUPC:** Aquesta columna numèrica indica el codi que la UPC assigna a cada assignatura
- **CREDITS:** Aquesta columna numèrica informa sobre els crèdits que atorga cada assignatura. Actualment, poden prendre els valors: 3; 4,5; 6
- **DEPARTAMENT:** En aquesta columna de valors de text, s'indica el departament al qual correspon cada assignatura. El nom del departament apareix abreviat amb les sigles assignades per l'escola.
- **QUADRIMESTRE:** Aquesta columna és de text i indica el quadrimestre al qual correspon l'assignatura. Pren valors entre Q1 i Q7.
- **CURS:** És una columna numèrica que pot prendre valors entre el 1 i el 4 i que indica a quin curs del grau correspon l'assignatura.
- **ASS:** Aquesta columna conté el nom de les assignatures abreviats. Per exemple, l'assignatura *Teoria de Màquines i Mecanismes* apareix com a *TMM*. D'aquesta manera, s'agilitzen els processos a l'hora de programar.

5.2.2 Dades preinscripció

Com s'ha mencionat, aquest fitxer de dades conté tota la informació necessària per a la realització de la pre-matricula dels estudiants. Aquest dataframe s'ha anomenat *general*. Les columnes d'interès d'aquest fitxer son les següents:

- **CODI_EXPEDIENT:** Aquesta columna és numèrica i dona un codi únic a cada estudiant que permet identificar a qui pertanyen les dades.
- **SEXE:** Aquesta columna és binària i pot prendre els següents valors: D o bé H segons si l'estudiant és Dona o Home.
- **ANY_ACCES:** Aquesta columna és numèrica i assigna a cada estudiant l'any on va realitzar la pre-matricula. Pot prendre valors en el rang entre 2010 i 2019. Si es decidís actualitzar el banc de dades, l'interval augmentaria fins a l'any que s'afegís.
- **NOTA_ACCES:** Aquesta columna numèrica mostra la nota de selectivitat amb la qual va accedir l'alumne.

Les columnes que s'han descartat son *CP_FAMILIAR*, *TIPUS_ACCES*, *CP_CENTRE_ACCES*.

S'ha considerat que aquestes columnes no eren pertinents per a l'anàlisi que es vol realitzar.

Un cop enteses les dades, es pot procedir a la filtració i neteja.

En un primer lloc s'ha constatat que les dades d'aquest fitxer corresponen a estudiants que només han realitzat la pre-inscripció i per tant existeix la possibilitat que no hagin acabat cursant el grau. Caldrà filtrar els expedients en base a un altre fitxer per assegurar-ho.

5.2.3 Dades Grau

Els fitxers d'informació del grau contenen les dades de cada convocatòria de cada expedient. Aquest dataframe s'ha anomenat *dades2*. Les columnes d'interès d'aquest fitxer son:

- **CODI_EXPEDIENT**: Aquesta columna és numèrica i indica l'expedient associat a les dades de cada fila
- **CODIUPC**: Aquesta columna numèrica indica el codi que l'Escola associa a cada assignatura del grau.
- **ANY**: Es tracta d'una columna numèrica que indica l'any on es va realitzar la convocatòria corresponent. Pot prendre valors entre el 2010 i el 2018, amb la possible actualització.
- **QUAD**: És una columna binària que pot prendre els valors 1 o 2 segons si la convocatòria va ser el quadrimestre de tardor (1) o el de primavera (2).
- **SUPERA**: Columna binària de valors 'S' o 'N', on els valors corresponen a si l'estudiant ha superat ('S') o no superat ('N') la convocatòria corresponent.
- **NOTADEF**: És una columna numèrica que pot prendre valors entre 0 i 10 i que indica la nota definitiva que ha obtingut l'estudiant en aquesta convocatòria.
- **GRUP**: És una columna numèrica que indica a quin grup es va realitzar la matrícula de la corresponent convocatòria.

Les columnes descartades han estat *CODPROGR*, *NOTA* i *AVAL*. Ja que s'han considerat no pertinents per a l'anàlisi realitzat en aquest treball.

A continuació s'explica el procés de neteja de les dades d'aquest dataframe.

En primer lloc s'han eliminat totes les files que no corresponen a convocatòries reals.

A continuació, s'ha aplicat la funció *isin* que retorna valor booleans segons si un objecte pertany o no al dataframe indicat. En aquest cas, s'ha mirat que els codis d'expedient dels dataframe *dades2* i *general* corresponguin. En cas negatiu, s'ha prescindit de la fila en qüestió.

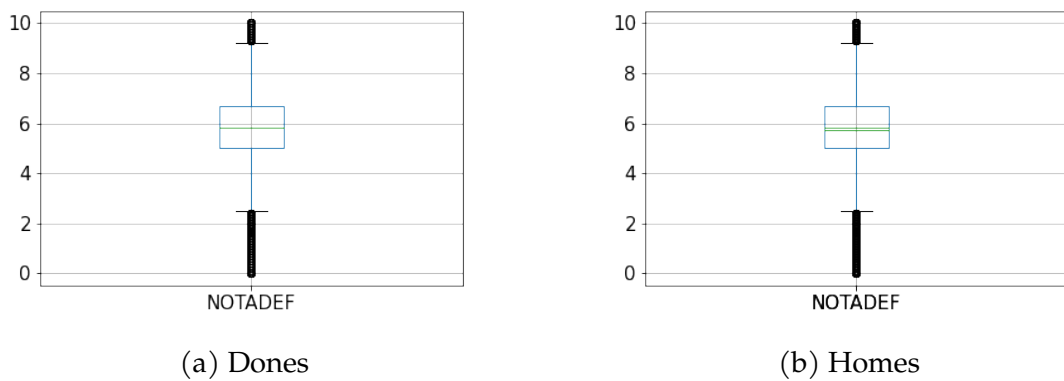


Figura 9: BoxPlot de les notes definitives dels estudiants

Aquesta mateixa funció s'ha aplicat per comprovar que totes les assignatures referenciades al dataframe *dades2* mitjançant el codi UPC apareixen al dataframe *assignatures*. Un cop més, s'ha prescindit de les files que no compleixen la condició. D'aquesta manera, treballarem només amb les assignatures obligatòries del grau.

Finalment, s'ha aplicat la funció *merge* de la llibreria *pandas*. Aquesta funció permet seleccionar dos dataframes i fusionar-los. Els dataframes amb els quals es treballa són *dades2* i el dataframe *assignatures*. La fusió es pot fer com els dos seguits, o es pot fer en base a una columna ja existent. En aquest cas, s'ha seleccionat la columna *CODIUPC* i s'ha realitzat la fusió en base a això.

El resultat final ens donarà doncs tota la informació mencionada en aquest apartat més la informació proporcionada per *assignatures*.

Cal clarificar també algunes decisions que s'han pres a l'hora de filtrar de les dades. Per exemple, s'ha mencionat que s'han descartat les assignatures no obligatòries. Això descarta les optatives tant de segon com de quart i les assignatures Projecte I i Projecte II, a segon i tercer ja que formen part dels blocs *optatives* i *projecte* del grau. S'ha pres aquesta decisió perquè s'ha decidit analitzar només el bloc obligatori del grau i aquestes assignatures no en fan part. Evidentment, com que un dashboard es personalitzable, es podrien incloure o fins i tot fer-ne un anàlisi exclusiu.

Finalment, per poder visualitzar les dades obtingues, s'ha realitzat un box plot de la mitjana de la nota definitiva dels alumnes en funció del gènere.

Com es pot observar a les figures 9a i 9b, hi ha bastants punts que es poden considerar *outliers*. S'ha decidit no prescindir-ne. En aquest set de dades, cada punt representa una convocatòria d'un estudiant. Si es decidís prescindir dels punts extrems, s'estaria anul·lant un resultat acadèmic vàlid. S'ha considerat doncs que com que les notes que apareixen en el set original de dades van ser vàlides per la comissió curricular en el seu moment, no es prescindirà de cap valor.

5.3 Unitats d'informació a representar

Un cop les dades han estat filtrades estan preparades per a treballar amb elles. Per tant, el següent pas d'aquest anàlisi és decidir de manera clara quines unitats d'informació es volen analitzar i sobretot com es volen representar.

Primer de tot, s'ha decidit que per a canviar de pantalla en pantalla s'afegiran tres botons a la part superior que permetin a l'usuari canviar entre les pàgines mencionades. A continuació s'explica quin procés s'ha seguit per a cada pantalla.

5.3.1 Evolució

L'anàlisi d'evolució consta principalment d'anàlisis temporals. Per això caldrà visualitzar diversos paràmetres en funció del temps. S'ha decidit mostrar:

- **Evolució Matriculacions:** Un dels paràmetres més importants a estudiar és l'evolució dels matriculats per any. S'analitzarà el nombre de d'alumnes de nou accés per any. Per això, es vol una gràfica lineal del número d'alumnes de nou accés en funció del temps. La gràfica presentarà dues línies, una per als homes i una per les dones.
- **Evolució Mitjana Ponderada:** De manera similar al paràmetre estudiat anteriorment, es vol analitzar l'evolució de la mitjana ponderada dels estudiants. Es vol una gràfica similar, és a dir una línia temporal per cada gènere que varii segons l'evolució dels resultats de l'estudiantat al llarg dels anys.
- **Taxa d'èxit global:** Per a tenir una idea general del rendiment dels estudiants segons el seu gènere, es vol visualitzar el percentatge d'aprovat i suspesos de cada gènere. Per a fer-ho, es fan servir diagrames de barra agrupat, on cada grup té una barra per al percentatge d'aprovat i una per el percentatge de suspesos. Es realitzen dues gràfiques, una per a cada gènere.

Tenint en compte tots els punts mencionats anteriorment, s'elabora un disseny de croquis preliminar que ens permet visualitzar com queda la pantalla d'evolució. Es pot observar a la figura [10](#)

5.3.2 Anual

Per a l'anàlisi anual, cal tenir en compte que es treballa amb dades estàtiques, i que les eines de visualització poden variar. Es vol doncs una pantalla estàtica que mostri tots els elements que s'han considerat pertinents per cada any. Cal una eina que permeti a l'usuari seleccionar l'any que vol visualitzar en cada moment. S'ha decidit fer servir un menú desplegable tipus *drop-down* que contingui tots els anys disponibles a la base de dades. Actualment, consta de les dades des de el 2011 fins al 2018. Un cop seleccionat l'any, les dades que s'han decidit visualitzar son:

- **Matriculats:** Es vol mostrar el percentatge de matriculats d'homes i dones a l'any corresponent. Per poder contextualitzar-ho, és interessant també afegir la variació respecte l'any anterior. Per a fer-ho s'ha decidit mostrar en números grans i

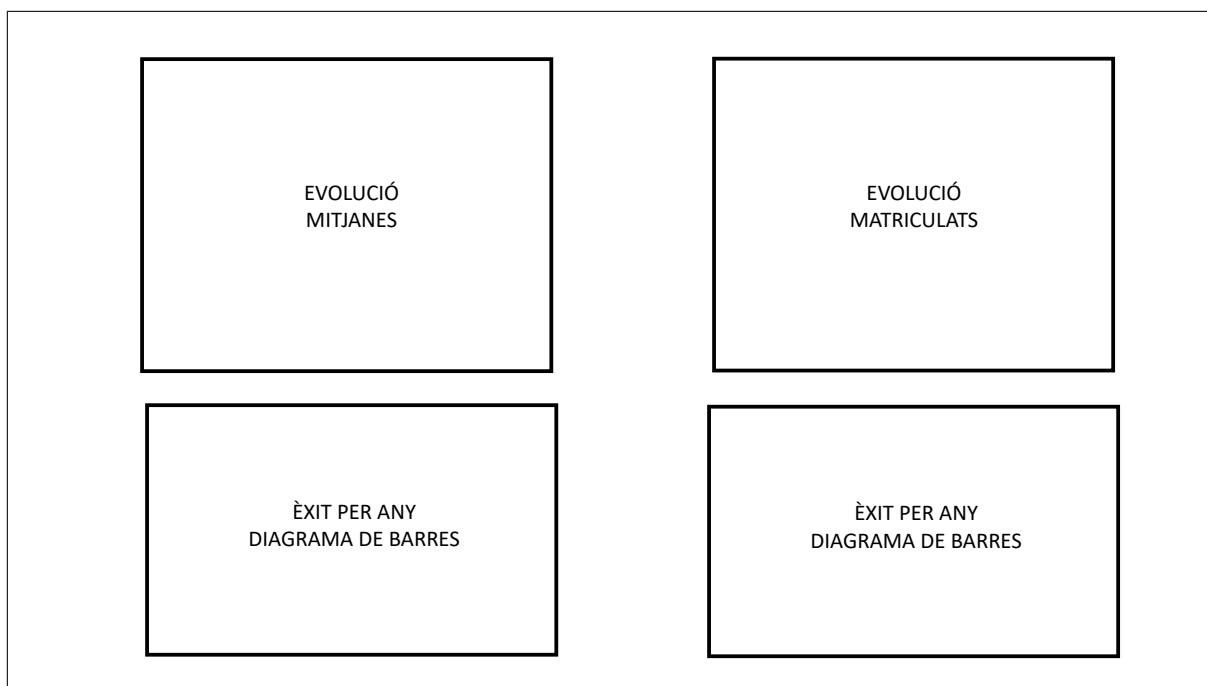


Figura 10: Croquis de la pàgina evolució del dashboard

destacables el percentatge, i afegir la variació en més petit tot acompanyat d'un indicador que mostri si es tracta d'un increment o decrement. Aquest element es considera el de màxima importància i es per això que s'ha decidit situar-lo a la part superior esquerra de la pantalla. A més a més, al tractar-se d'un element estàtic, és més fàcil de llegir i assegura la comprensió ràpida de part de l'usuari.

- **Resultats:** Cal també un anàlisi dels resultats acadèmics en funció del gènere. Tenint en compte que hi ha moltes assignatures al grau, es busca trobar la manera de separar-les per tal de no sobrecarregar la pantalla.
 - **Resultats per quadrimestre:** Una manera útil i intuïtiva de separar les dades és per quadrimestre. Permet avaluar el rendiment tot seguint una separació lògica. Per a fer-ho, es farà servir un diagrama de barres agrupat. Cada grup tindrà dues barres corresponents a homes i dones. Les gràfiques es fan quadrimestrals i per tant cada una mostra les assignatures del quadrimestre corresponent. La manera de seleccionar el quadrimestre serà amb una barra tipus *slide*, que permet fer-ho el més visual possible i evitar submenús que dificultin la comprensió de la pàgina. Aquest element és important, però el caire interactiu fa que requereixi més dedicació per entendre i sobretot per a visualitzar tota la informació, ja que hi ha 7 gràfiques per veure. Per tant, s'ha decidit situar a la part esquerra però inferior.
 - **Resultats per departament:** D'altra banda, és interessant també la informació per departament. Com que no cal entrar en detall de departament en aquesta pàgina, s'ha decidit mostrar el millor i pitjor departament de cada

gènere. El criteri amb el que es decideix és la mitjana ponderada obtinguda pels estudiants a cada departament. Es vol mostrar aquest paràmetre, així com el percentatge d'aprovat i suspesos de cada gènere dels departaments pertinents. Per fer-ho, les dades es col·loquen en una taula que mostrarà els paràmetres mencionats. Aquesta taula conté molta informació i a més a més és interactiva. No s'ha considerat l'element més important de la pantalla, i per això s'ha decidit encabir a la part inferior dreta. D'aquesta manera l'usuari hi interactuarà un cop hagi visualitzat altres elements, i tingui ja una idea global de la informació.

- **Taxa d'èxit:** Finalment, el rendiment es vol mostrar també amb la taxa d'èxit. Es mostren els percentatges globals d'aprovat i suspesos aquell any. Per fer-ho, es genera un diagrama tipus Pie Chart, on apareixen els percentatges mencionats. Aquest element és també estàtic, molt visual i d'una importància destacable. Per això s'ha decidit col·locar a la part superior esquerra, just després de l'element d'informació de matriculats.
- **Anàlisi per franjes:** Per a tractar el gran percentatge de dades, es vol fer un anàlisi per franjes. El paràmetre que permet fer aquesta divisió és la mitjana ponderada. S'ha decidit dividir el rang de nota possible, és a dir del 0 al 10, entre 4. S'assigna a cada expedient un dels rangs. La informació que es vol mostrar de cada rang és el percentatge total de l'alumnat que pertany a la franja, així com el percentatge de dones i homes que hi ha. Per a fer-ho, s'ha decidit representar una barra interactiva que al seleccionar un quartil representi la informació mencionada. Aquest element es de gran importància i a més a més és fàcil i ràpid de llegir. No obstant, és interactiu i per tant no volem que sigui el primer que veu l'usuari. Per això, s'ha decidit situar-lo a la part superior dreta, perquè sigui el primer element interactiu amb el que es topa l'usuari, però no el primer de tota la pàgina.
- **Mitjana i variància:** Aquests paràmetres son molt importants a l'hora d'analitzar una serie de dades. Per a fer-ho visual i immediat, s'ha decidit representar la mitjana amb una gràfica tipus *gauge* que indiqui ràpidament si la mitjana es aprovada o no. Aquest element és estàtic, i no es difícil d'entendre. S'ha decidit doncs que es col·locarà on hi hagi un espai buit, doncs la seva posició no és determinant.

Així doncs, tenint en compte tots els elements que es volen representar, s'ha decidit que la repartició en pantalla es pot veure a la figura 11.

5.3.3 Departament

La pantalla d'anàlisi de departament mostra un anàlisi del rendiment dels estudiants segons el gènere. Permet veure de manera més detallada les dades de cada departament en aquest anàlisi de gènere.

L'usuari ha de poder triar quin departament vol visualitzar. S'ha decidit altre cop fer servir un menú desplegable del tipus *drop-down*. D'aquesta manera, es permet a l'usuari seleccionar fàcilment el departament.

Per a aquesta pantalla es vol visualitzar:

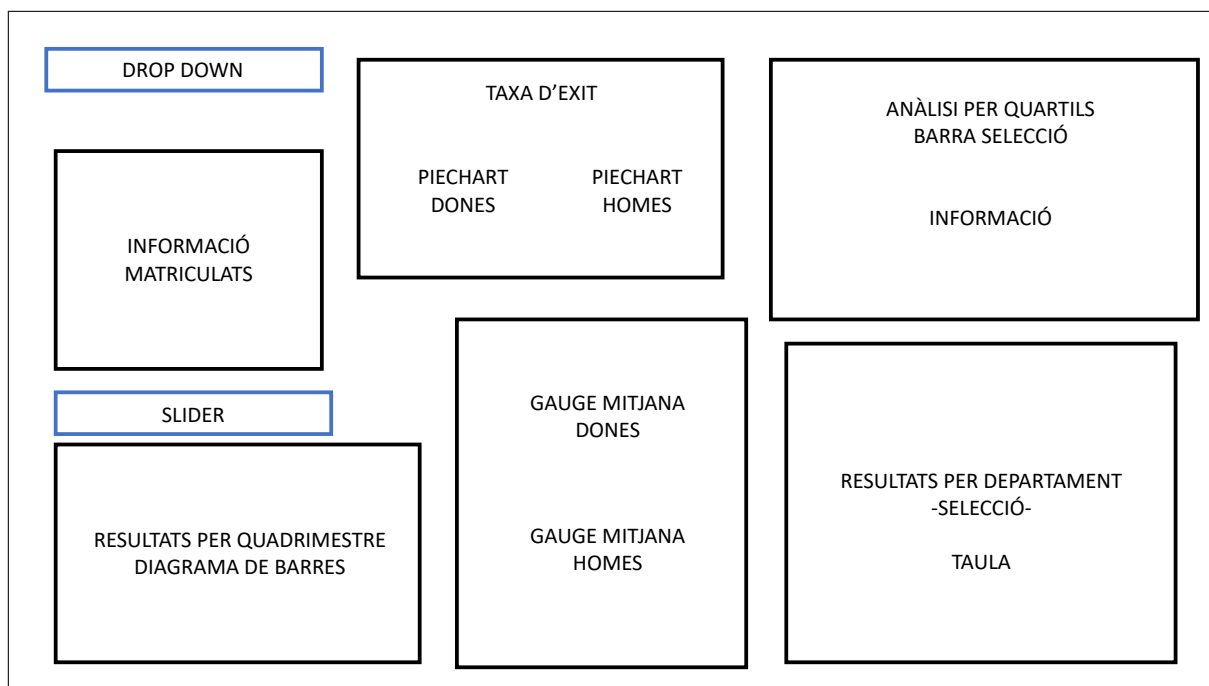


Figura 11: Croquis de la pàgina anual del dashboard

- **Alumnat:** Es vol mostrar el percentatge d'alumnes dones i homes que hi ha al departament.
- **Taxa d'èxit:** Un paràmetre important a mostrar és la taxa d'èxit del departament per gènere. Aquí, es mostren els percentatges d'aprovat i suspesos en el departament seleccionat. Per a fer-ho, es realitza una gràfica tipus Pie Chart *donut* tant per a homes com per a dones on apareixen els percentatges de cada categoria.
- **Mitjanes:** S'ha decidit també mostrar la nota mitjana de l'alumnat segons el departament seleccionat. Per a fer-ho, es vol ensenyar la mitjana de cada assignatura del departament. Per tant, per realitzar-ho, es vol un diagrama de barres agrupat, on cada grup correspon a una assignatura del departament. Dins de cada grup, hi ha una columna per a homes i una per a dones per tal de poder comparar-ne les mitjanes.
- **Evolució:** Finalment, és interessant remarcar com ha evolucionat el departament al llarg dels anys del grau. Per fer-ho, es vol un timeplot on l'eix horitzontal correspon als anys. Els paràmetres que s'analitzaran són:
 - **Matriculats:** S'estudia l'evolució del nombre de matriculats homes i dones al departament
 - **Mitjanes:** S'estudia com han evolucionat les mitjanes tant d'homes com de dones al llarg dels anys.

Així doncs, tenint en compte tots els elements que es volen representar, s'ha decidit que

la repartició en pantalla es pot veure a la figura 12.

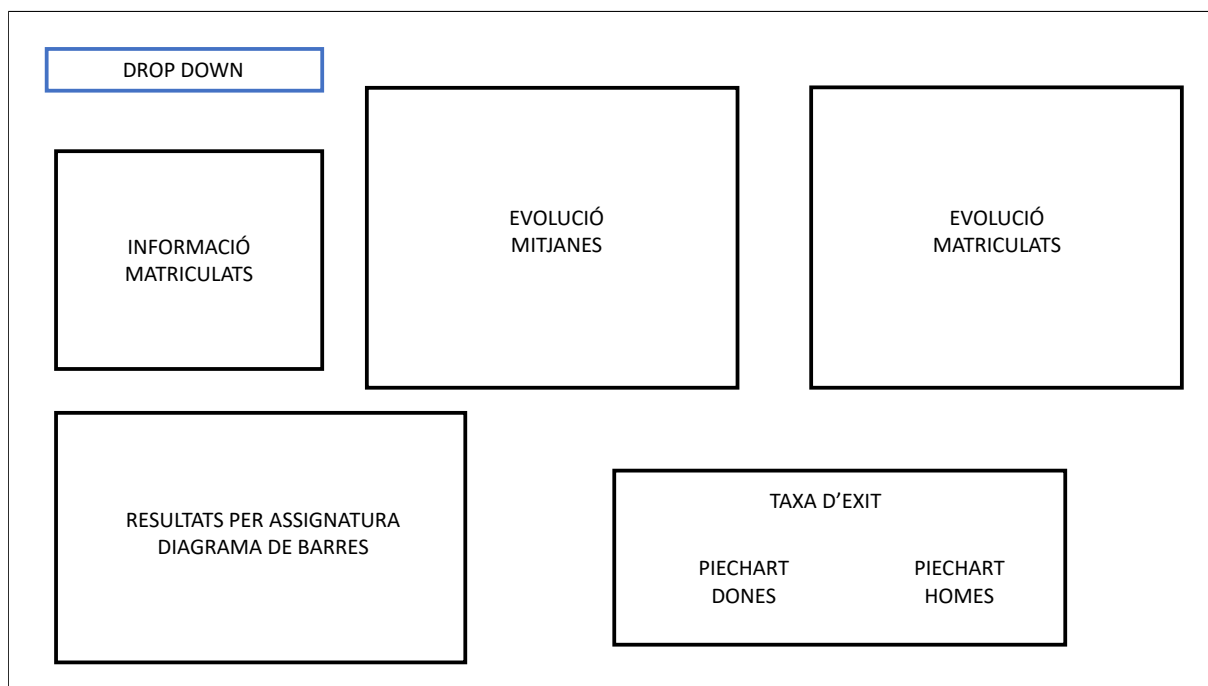


Figura 12: Croquis de la pàgina departament del dashboard

6 Desenvolupament

Un cop s'ha elaborat el disseny preliminar de com ha de ser l'aplicatiu web, quines dades ha de mostrar i quines eines de visualització de dades es volen fer servir, es pot passar a la part de programació web. Aquesta part és l'últim pas per a assolir el pas de comunicació en el marc del *Data Science*.

Com ja s'ha vist en apartats anteriors, tota aplicació web es basa d'una part de front-end i una de back-end que tenen funcions diferents i, per tant, una programació diferent. A continuació, s'explica la programació que cada part requereix.

El desenvolupament del codi per a l'aplicació web s'ha fet mitjançant el procediment Flask. La maquetació o *layout* de la carpeta on es treballa queda de la següent manera:

```
dashboard
├── run.py
├── app
│   ├── __init__.py
│   ├── views.py
│   ├── dades
│   │   ├── grau .5 qfaseini19.csv
│   │   │   └── qfasenoini19.csv
│   │   ├── assignatures_geti.csv
│   │   └── general .5 dpersnomespreins19esc.csv
│   ├── scripts
│   │   ├── genere
│   │   └── genere_anyoutput
│   ├── static
│   │   ├── any
│   │   │   └── fitxers.png
│   │   ├── evolucio
│   │   │   └── fitxers.png
│   │   └── departaments_etseib
│   │       └── fitxers.png
│   └── templates
│       ├── 2016.html
│       ├── departament.html
│       └── evolucio.html
```

A continuació, s'explica la funcionalitat de cada fitxer i com treballa el conjunt

6.1 Back end

El back-end conté tota la part de programació que no veu directament l'usuari. El codi d'aquest bloc treballa per darrere de l'aplicació i assegura que totes les dades, figures i eines necessàries per a l'aplicació estiguin disponibles.

6.1.1 Programari

El llenguatge de programació emprat per al codi del back-end ha estat Python Idle 3. Aquest llenguatge és molt potent per a anàlisi de dades i complementat amb les bones llibreries permet construir una gran varietat d'eines de visualització de dades.

Les llibreries que s'han fet servir són:

- **Pandas:** Per a poder tractar amb DataFrames i facilitar la manipulació dels conjunts de dades
- **Matplotlib:** Per a la realització de gràfiques
- **NumPy:** Complementat amb *Pandas*, per a agilitzar operacions
- **Plotly Express:** Complementat amb Matplotlib, per a la realització de gràfiques
- **Kaleido:** Per a poder descarregar i desar les imatges elaborades al llarg del codi.

6.1.2 Explicació del codi

L'elaboració del codi ha consistit en diverses funcions que s'han dividit segons la pantalla on envien els valors sortint, i segons si són operacions a realitzar prèviament (pre-procés) o bé càlculs que es faran en directe segons la tria de l'usuari.

Pre-Procés El codi de pre-procés s'ha elaborat al fitxer *genere.py*.

Es treballa amb les dades filtrades que s'han mencionat en apartats anteriors. En aquest codi, es crea també una llista per als diversos departaments que apareixen a les dades i una llista anàloga amb els anys disponibles. L'objectiu d'aquesta llista és que l'usuari visualitzi les opcions disponibles tant de departaments com d'anys. Aquest apartat de neteja, filtració i preparació de les dades es pot trobar a l'annex [A.1.4](#).

Evolució D'altra banda el fitxer conté també l'elaboració de gràfiques estàtiques. Aquestes són les gràfiques que no canvien en funció de la tria de l'usuari. En el cas d'aquesta web, corresponen a les gràfiques de la pàgina Evolució. Aquestes gràfiques fan servir l'històric de dades i no canvien segons l'*input* de l'usuari. Consisteixen en dos gràfics de línia tipus *timeplot*, un que mostra el nombre de matriculats en funció del temps (Figura 13) i un altre que mostra les mitjanes ponderades en funció del temps (Figura 14). Les dues gràfiques tenen dues línies, una per a cada gènere. L'altra gràfica estàtica consisteix en dos *Pie Charts* de tipus *Donut* que mostren el percentatge d'aprovat i suspesos per gènere. (Figura 15).

Aquestes gràfiques es guarden a una carpeta anomenada *statics* que conté totes les imatges que la web requereix per a funcionar. Les figures *Python* es guarden com a format imatge (*.png* o *.svg* segons requeriments) mitjançant la llibreria *Kaleido*.

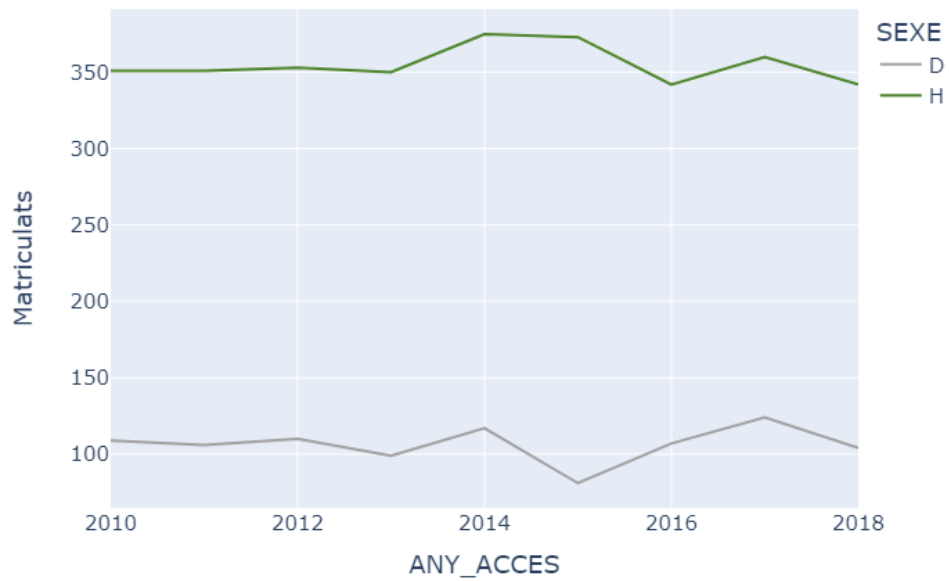


Figura 13: Evolució del nombre de matriculats en funció del temps

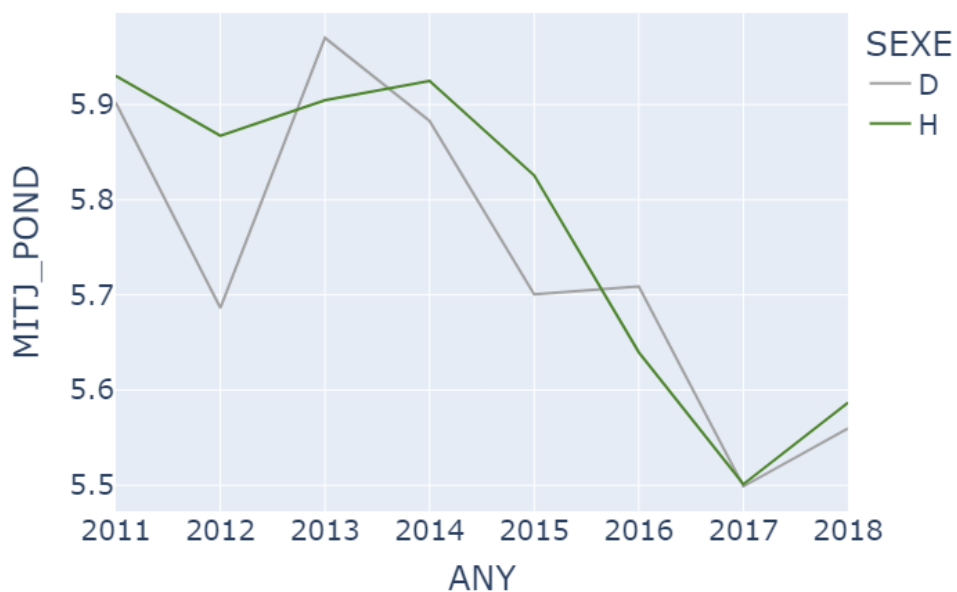


Figura 14: Evolució de les mitjanes al llarg del temps

Departament Per als càlculs de la pàgina departament, s'ha elaborat un bucle dins el document. Aquest *loop* genera totes les figures necessàries per la pàgina en funció de cada departament. Aquesta part treballa principalment amb dos *dataframes*. En un primer lloc un *dataframe* amb tota la informació anomenat *junt_dept* que manté les ma-

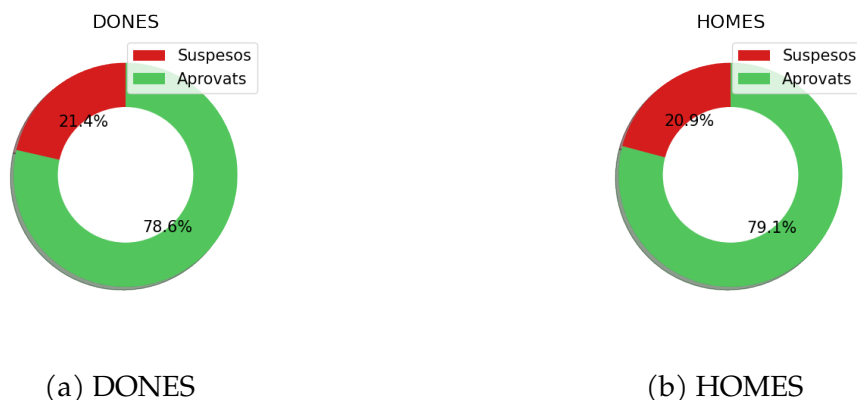


Figura 15: PieChart de la taxa d'èxit

teixes columnes que el *dataframe* *dades2*, però eliminant totes les files que corresponguin a departaments diferents del seleccionat. Després, s'ha creat un *dataframe* anomenat *dept* que ha agrupat les files en funció dels expedients i el nombre de crèdits. Això ens permetrà calcular la mitjana ponderada de cada estudiant. Amb aquests dos *dataframes* i una sèrie de comandes que es poden trobar en annex, s'ha pogut calcular la mitjana ponderada de cada alumne. Finalment, s'ha creat un últim set de dades anomenat *exit_dept* que agrupa les dades en funció de gènere i del paràmetre 'SUPERAR' que indica si l'estudiant ha aprovat o no aquella convocatòria.

Amb aquestes dades, ja es pot passar a l'elaboració de les gràfiques.

Primer de tot s'han elaborat els *PieCharts* de la taxa d'èxit del departament. El codi emprat és el mateix que el que s'ha fet servir a l'apartat anterior, però amb les dades pertinents per a aquest apartat. També s'ha realitzat un histograma o diagrama de barres vertical de les notes mitjanes de cada assignatura del departament. Cada assignatura té dues columnes que mostren les notes segons el gènere. (Figura 16).

Finalment, s'han generat dos timeplots similars als de la pantalla d'evolució. S'ha buscat representar l'evolució del nombre de matriculats i de les mitjanes ponderades de cada departament al llarg dels anys. Un cop més, el diagrama mostra dues línies d'evolució, una per a cada gènere.

	Dones	Homes
Departament		
Mitjana		
Aprovats		
Suspesos		

Taula 1: Taula de la informació dels departaments a la pàgina anual

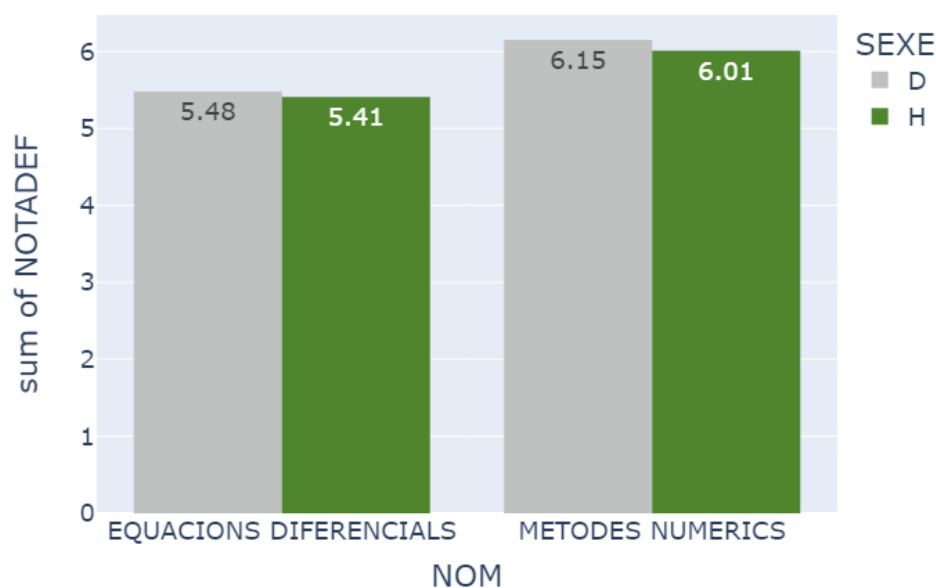


Figura 16: Histograma agrupat per gènere de les diferents assignatures del departament

D'altra banda, es vol elaborar una taula que contingui la informació del millor i pitjor departament en funció del gènere. La taula ha d'informar sobre la informació esmentada en apartats anteriors, i, per tant, té un aspecte com es pot observar a la Taula 1.

Un cop més, totes les imatges es guarden automàticament a la subcarpeta *static* de la carpeta *app*.

Càlculs en viu D'altra banda, s'ha generat un fitxer per als càlculs en directe. Aquests són els càlculs que s'elaboren cada vegada que l'usuari fa una tria o selecció. En el cas d'aquesta aplicació, les pàgines que requereixen càlculs en directe són la pàgina de departament i l'anual. Aquests es troben en el fitxer *genere_anyoutput.py*. La primera comanda del fitxer consisteix a importar el *script* del fitxer mencionat anteriorment *genere.py*. D'aquesta manera, es pot treballar amb les dades ja tractades.

Anual Per al codi d'aquesta pàgina, s'ha elaborat una funció dins el mateix fitxer *genere_anyoutput.py*. La funció anomenada *dades_genere(a)* té com a *input* l'any seleccionat

per l'usuari.

Els primers càlculs que es realitzen són els del nombre de noves matrícules, i els punts que ha variat respecte a l'any anterior. Per a fer-ho, es treballa amb un *dataframe* anomenat *acces* que ha agrupat el *dataframe* original *general* en funció de l'any d'accés i el gènere i ha comptat els expedients en cada un. Tot seguit, amb una sèrie de comandes es poden calcular els números desitjats. D'aquestes operacions, surten les variables anomenades:

- **homes_matric:** Percentatge de noves matrícules homes
- **dones_matric:** Percentatge de noves matrícules dones
- **variacio_dones:** Punts que ha variat la taxa de matriculades dones
- **variacio_homes:** Punts que ha variat la taxa de matriculats homes
- **imh i imd:** Nom de la imatge que caldrà mostrar en pantalla, segons si l'increment es positiu o negatiu.
- **Informació de la taula:** Es retorna també una sèrie de valors que permeten
- **Dmillor_dept, Hmillor_dept, Dpitjor_dept, Hpitjor_dept:** Nom del millor i pitjor departament per dones i per homes.
- **Dmillor_nota, Hmillor_nota, Dpitjor_nota, Hpitjor_nota:** Mitjana ponderada del departament seleccionat tant per homes com per dones.
- **Dmillor_suspesos, Hmillor_suspesos, Dmillor_aprovats, Hmillor_aprovats, Dpitjor_suspesos, Hpitjor_suspesos, Dpitjor_aprovats, Hpitjor_aprovats:** Percentatge de suspesos i aprovats per al departament seleccionat tant per homes com per dones.
- **percentD, percentH i percentTOT:** Percentatges de Dones, Homes i Totals de noves matrícules l'any seleccionat.

Els elements que surten com a *output* de la funció són cridats quan es requereixen, i permeten que amb una sola funció, es tingui tot el codi disponible.

D'altra banda, aquesta funció també genera gràfiques no estàtiques, és a dir que canvien segons l'any que seleccioni l'usuari.

En primer lloc, s'han realitzat dos *PieCharts* de tipus "dònut" per a il·lustrar la taxa d'èxit per a homes i per a dones. Aquesta gràfica és del mateix tipus que les mencionades anteriorment.

A continuació s'han elaborat les gràfiques per a veure les notes mitjanes de cada sexe de cada assignatura, separades per quadrimestre. Aquestes gràfiques s'han dut a terme amb la comanda de diagrama de barres explicada anteriorment. S'ha creat un *loop* dins

la funció que elabora una gràfica per a cada quadrimestre.

Finalment, s'han generat les gràfiques tipus *gauge* on s'il·lustra la mitjana dels estudiants homes i dones. Aquestes gràfiques indicador de velocitat s'han generat amb una subfunció de la llibreria *plotly*. Se'n pot veure un exemple a la figura 17.

Anàlogament, als altres codis explicats, les figures es guarden a la carpeta *static*.

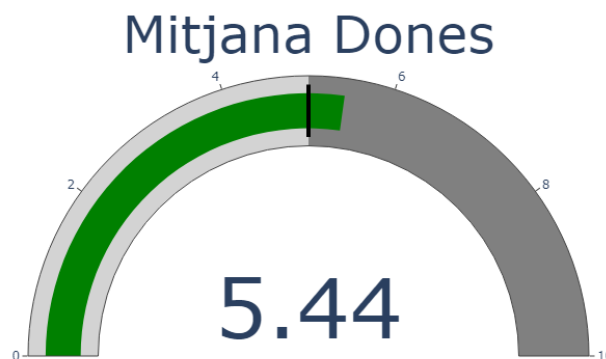


Figura 17: Figura tipus indicador de velocitat il·lustrant les mitjanes de les dones

Codi web Fins ara s'ha comentat el codi de *Python* que du a terme el tractament de dades i elabora les eines de visualització d'aquestes. Però el *back-end* d'una pàgina web també conté fitxers que assegurin el bon funcionament de la web. A continuació, es detalla la funció de cada un.

run.py El fitxer anomenat *run.py* es troba fora de la carpeta *app*. Aquest fitxer importa la carpeta *app* i l'executa. El codi es pot veure en annex [A.1.3](#)

__init__.py Aquest fitxer serveix per a executar l'aplicació mitjançant l'eina Flask. Aquest fitxer genera la instància Flask. En podem torbar el codi en annex [A.1.2](#).

views.py El fitxer *views.py* conté totes les rutes que seguirà l'aplicació. Cada *template* del programa *HTML* és cridat per una ruta marcada en aquest fitxer. A més a més, també s'encarrega d'executar les funcions que retornen *outputs* variables, és a dir que canvien segons la selecció de l'usuari. Aquest fitxer treballa en viu i és l'intermediari entre el codi de *Python* i el d'*HTML*. Cada ruta té diversos *outputs*, segons si requereix informació en viu que ha de ser enviada a posteriori a una selecció de l'usuari. L'output que totes tenen en comú és el *template* que han d'enviar. Això es fa mitjançant la comanda *render_template* importada de la llibreria *flask*. En el cas d'aquesta aplicació, hi ha tres rutes creades:

- `/`: correspon a la ruta inicial, la que s'obre per defecte. En el nostre cas, està posat que la pàgina inicial per defecte sigui la d'evolució.
- `/evolucio`: Aquesta ruta és similar a l'anterior, ja que crida la pàgina *evolució*.
- `/any`: Aquesta ruta retorna la pàgina web 'anual'. Aquesta conté una funció que executa els codis del preprocés necessaris. En concret, executa la funció *dades_any(a)* on l'*any input* és aquell seleccionat per l'usuari. Recull les dades de la funció i les retorna com a *output* per a ser utilitzades a la pàgina web.
- `/departament`: Aquesta crida el fitxer *departament* que retorna la pàgina amb el mateix nom. A més a més, executa la funció *departaments* explicada anteriorment. Recull els valors que retorna aquesta, i els envia com a *output*.

El codi d'aquest fitxer es pot trobar en annex [A.1.1](#).

6.2 Front end

El front-end conté la part de programació que està en contacte amb l'usuari. Això inclou tant la part de disseny com pot ser colors, formes, posicions, com la part interactiva. En aquest apartat s'explica el programari emprat i el codi final.

6.2.1 Programari

El programari que s'ha fet servir per a aquesta part de l'aplicació és *HTML* acompanyat dels mòduls *CSS* i *JS*.

HTML permet crear pàgines web des de zero, adaptant-se el millor possible a les necessitats de l'usuari. Principalment, s'han fet servir mòduls de *w3-school*. Mitjançant aquestes eines s'ha pogut crear la part de disseny. El codi *CSS* permet crear estils uniformes al llarg de la pàgina i vetlla perquè el programador pugui personalitzar la pàgina al seu gust.

Finalment, el llenguatge *JS* ha permès crear la part més interactiva de la web. Com s'ha explicat, aquest llenguatge permet generar *scripts* dins el mateix codi *HTML*. Això permet crear elements interactius com menús, botons, eines lliscadores tipus *slider* que poden ser útils.

6.2.2 Explicació del codi

Per a crear les diverses pàgines, s'han generat tres fitxers *HTML*, que anomenem *templates*. Cada fitxer està generat d'una manera concreta, però tots tenen alguns elements en comú. Aquests són els més emprats:

- `<div>`: Aquest element genera un contenidor que engloba tots els elements que s'hi escriu dins. Pot contenir text, imatges o qualsevol element que es desitja. La gran utilitat que té és que permet establir colors de fons, posicions i altres elements estètics de tot un conjunt d'elements, que interressi que vagin junts.

- ****: Aquesta etiqueta permet afegir imatges. Cal cridarles seguint el directori necessari. Permeten també especificar llargada, alçada i posició de la imatge importada.
- **<h1>**, **<h2>**,... Aquestes etiquetes permeten generar quadres de text de diferents mides de lletra. Permeten especificar detalls de disseny com color de fons o posició.
- **<table>**: Etiqueta que genera una taula, totalment personalitzable.
- **class**: Aquest element es col·loca dins una etiqueta, i defineix la classe de l'element. La classe pot determinar el comportament de l'element o bé l'estil.
- **<script>**: Etiqueta que permet generar un codi dins el propi HTML. Es d'utilitat perquè permet posar el front-end en contacte amb el back-end, crear eines interactives com botons o sliders. Permet elaborar funcions.
- **style**: Aquest element es pot situar al principi del document com a etiqueta per a tot el llarg del document o bé dins de cada etiqueta específica. La seva funció es determinar paràmetres d'estil.
 - Al principi del document: Permet definir paràmetres d'estil de les classes establertes
 - Dins de cada etiqueta: Defineix els paràmetres d'estil de l'element en concret.
- **<form>**: Aquest element permet crear un formulari, on l'usuari introdueix les dades escollides, ja sigui mitjançant un click d'un botó, selecció d'un menú o entrada escrita. i aquesta informació és enviada al back-end.

Per tal de mantenir la línia estètica al llarg dels diversos *dashboards*, s'ha creat un *header*, o títol, que es manté en les tres pàgines. Aquest porta com a text el títol del treball i a la part esquerra conté els tres botons que permeten navegar al llarg de les pàgines.

Pel que fa a la paleta de colors que s'emptra en tot l'aplicatiu, s'ha triat el color gris clar pel fons general, el blanc pel fons dels elements, i el blau per elements destacables (com el títol, botons, i altres elements interactius). Aquest codi es pot veure en Annex.

2016.html Aquest fitxer correspon al *template* per a la pàgina anual. És el més complex dels tres i conté la més gran quantitat d'elements. La part més rellevant del codi d'aquest fitxer es pot trobar en annex [A.2.1](#)

La part d'estil col·loca on s'ha considerat idoni tots els elements mencionats anteriorment. A continuació s'explica el codi darrere els elements més interessants.

Menú anys Aquest element és el selector d'anys, que s'ha decidit presentar-lo en forma de menú en cascada. Això permet mostrar els anys possibles que pot triar l'usuari. Aquest menú està contingut dins d'un element `<form>` i s'anomena `<select>`. Gràcies

a un bucle tipus *for*, el programa del *front end* pot presentar tots els valors de la llista anomenada *anys* que envia el *back end*. Acompanyat d'aquest, a la seva dreta, hi ha un botó amb el text *Ves-hi* que un cop seleccionat, envia l'usuari a la pantalla anual de l'any seleccionat. Aquest element es pot observar per més claredat a la figura 18.

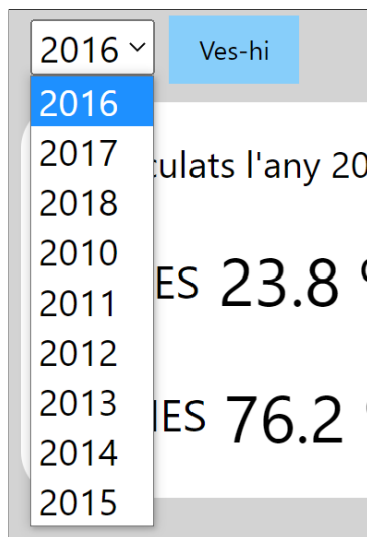


Figura 18: Element menú tipus cascada de la pàgina anual

Informació matriculats El següent element que s'afegeix és la informació del nombre de matriculats. Com s'ha explicat anteriorment, aquestes dades apareixen de forma numèrica. S'ha decidit optar per números de mida gran dins un element *<div>*, acompanyats d'una icona que indica si incrementa o disminueix respecte l'any anterior. Tota la barra superior està continguda dins un element barra que assegura que estan ben alineats.

Aquest element és estàtic dins el codi *HTML*, ja que recull directament els *outputs* de la funció cridada. L'element es pot veure la figura 19.

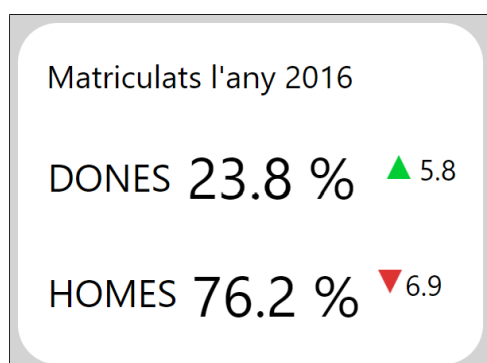


Figura 19: Element mostrant la taxa de matriculats anual

Taxa d'èxit: Les gràfiques tipus *PieChart* mostrant la taxa d'èxit s'han col·locat en la mateixa línia que la informació de matriculats. Anàlogament a l'element anterior, les gràfiques es criden de la carpeta *static*, on han estat guardades per la crida inicial de la

funció. A la figura 20 es pot observar com queda aquesta peça.

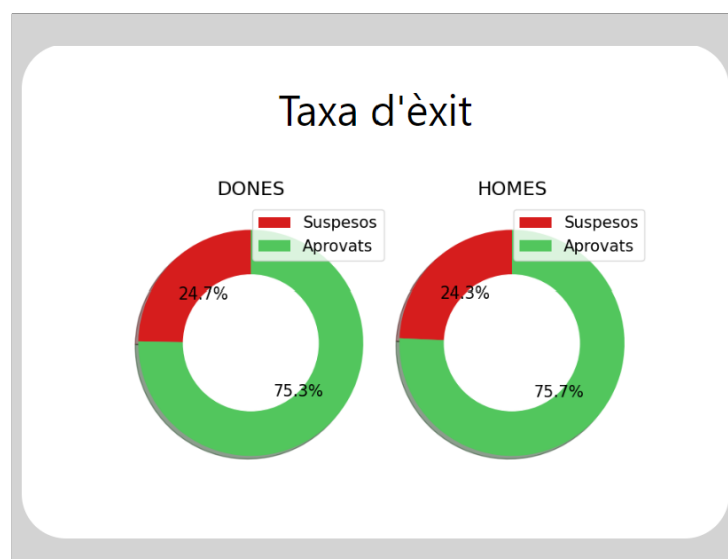


Figura 20: Element mostrant la taxa d'èxit anual

Anàlisi per franges: L'últim element en la línia superior és l'anàlisi per franges. Per a fer-ho encara més visual, s'ha optat per una barra horitzontal al llarg de l'element `<div>`. Aquesta barra està formada de quatre botons d'element tipus `<w3-button>`. Aquest element pertany a la llibreria de *W3-school*. Immediatament a sota d'aquesta barra, hi ha quatre quadres de text que informen sobre el significat de cada valor que apareix. En seleccionar un dels quatre botons de la barra, aquests valors canvien per adaptar-se al botó seleccionat. El codi darrere aquest element consta d'una funció, anomenada *Quartil*, que té com a input el quartil que selecciona l'usuari. Segons el valor d'aquest, retorna un o altre valor per a cada peça d'informació. A la figura 21 s'observa aquest element.

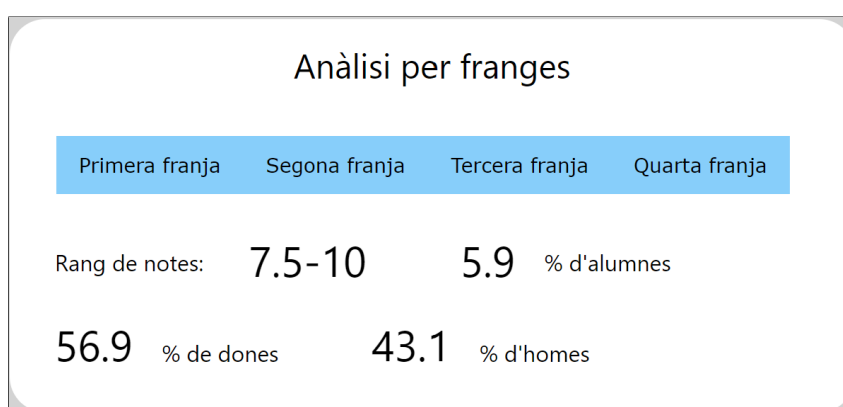


Figura 21: Element mostrant la informació per franges de cada any

Element Informació per quadrimestre: Aquest element conté tres parts. Primer de tot, el títol, que indica quin quadrimestre estem observant. Immediatament a sota, apareix la barra que permet lliscar entre quadrimestres. A l'interactuar amb la barra, el títol

canvia indicant en tot moment quin quadrimestre s'ha seleccionat. Un cop s'ha triat, a sota la barra apareix la gràfica corresponent a la tria.

El codi darrera aquest element consta de dues parts. Primer, un `<div>` de classe `slide-container`, és a dir la barra per a lliscar, acompanyat d'un element `` que crida una imatge en funció del valor `output` de la funció de l'element. Aquesta està continguda dins un element `script` que genera una variable en funció del quadrimestre seleccionat. A partir d'aquí, retorna el valor del quadrimestre pel títol, i crida la imatge adient a la carpeta `static` per a ser mostrada a la web. Anàlogament a la barra superior, els elements d'aquesta part estan continguts dins d'un element barra que manté l'alineació. El resultat final de l'element es pot observar a la figura 22

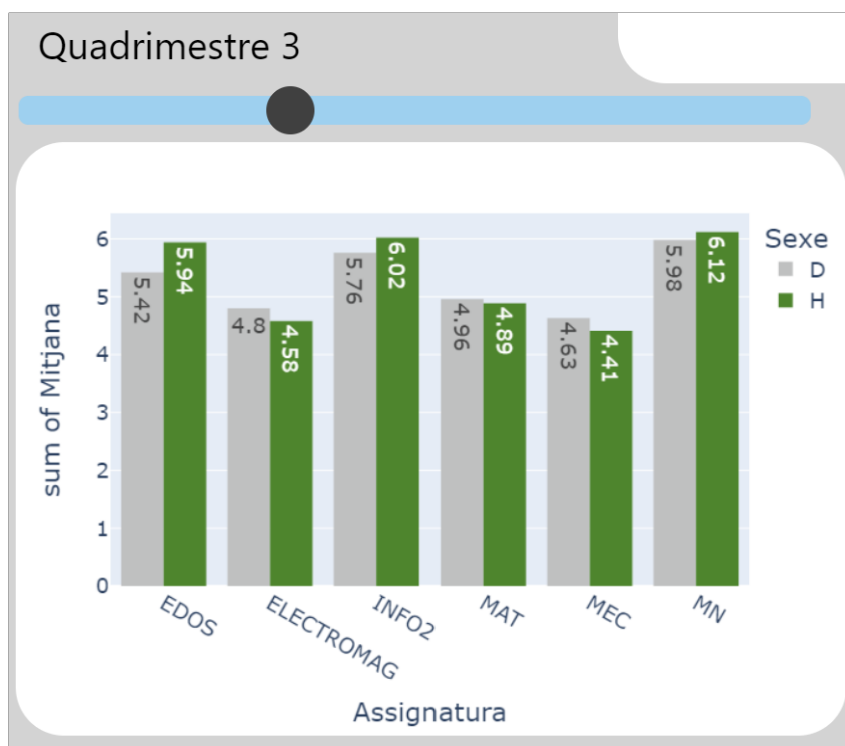


Figura 22: Barra lliscant i gràfic de barres de la pàgina anual

Element Mitjanes: Aquest element s'ha decidit col·locar a la part inferior, de manera vertical. Les dues gràfiques apareixen una sobre l'altre, dins un mateix element `<div>`. A la figura 23 es pot observar com queda aquest element.

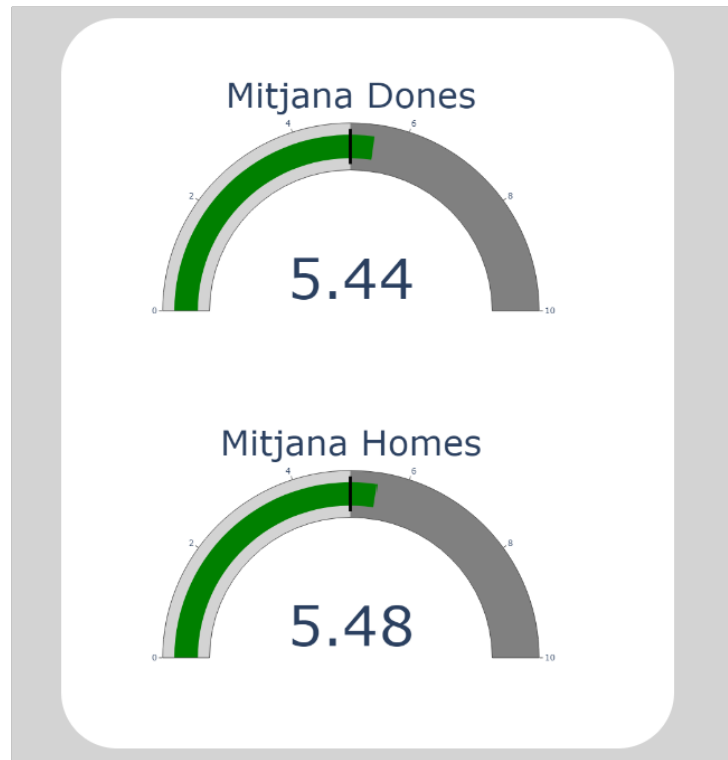


Figura 23: Element representant les mitjanes anuals dels estudiants

Rendiment per departament: Finalment, trobem l'element que realitza l'estudi de cada departament. L'element general és tipus `<div>` i conté el títol (Rendiment per departament), els botons anomenats "Millor" i "Pitjor" i la taula interactiva. Al pitjar un dels botons, apareixen a la taula els valors corresponents. Aquesta és de tipus `<table>` i cada casella crida un element en funció del seu identificador. Amb això, s'han generat dues funcions, una pel botó de millor i l'altre pel de pitjor. Les funcions treballen de manera anàloga. Les funcions criden els valors *outputs* de la funció general del codi *Python*. A cada variable se li assigna un codi identificatiu anomenat "id" que serveix perquè l'element `<table>` els pugui reconèixer. El resultat final es pot veure la figura [24](#).

Rendiment per departament

	Millors	Pitjors
	Dones	Homes
Departament	EPC	EPC
Mitjana	6.8	6.9
Aprovats	100.0%	97.5%
Suspesos	0.0%	2.5%

Figura 24: Taula d'informació per departament

Un cop col·locats tots els elements, ja es pot observar el resultat final de la pàgina (Figura 25)



Figura 25: Dashboard Anual

departament.html Aquest fitxer és el *template* per la pàgina d'anàlisi per departament. Conté molts elements semblants al codi explicat anteriorment, per tal de mantenir la línia estètica. A diferència del codi anterior, els elements no formen part de diversos elements barra. Aquesta pàgina no conté tants elements, s'ha vetllat doncs d'obtenir que totes les imatges siguin grans i fàcils de llegir. Això s'ha aconseguit mitjançant el codi CSS dins dels elements, que permet fixar la posició i mides. Notablement, dins de cada element, es pot afegir la classe *style* i es pot fixar la posició, en percentatges de l'espai total, amb les comandes *margin-left*, *margin-right* entre d'altres. El resultat final es pot observar a la figura [26](#)

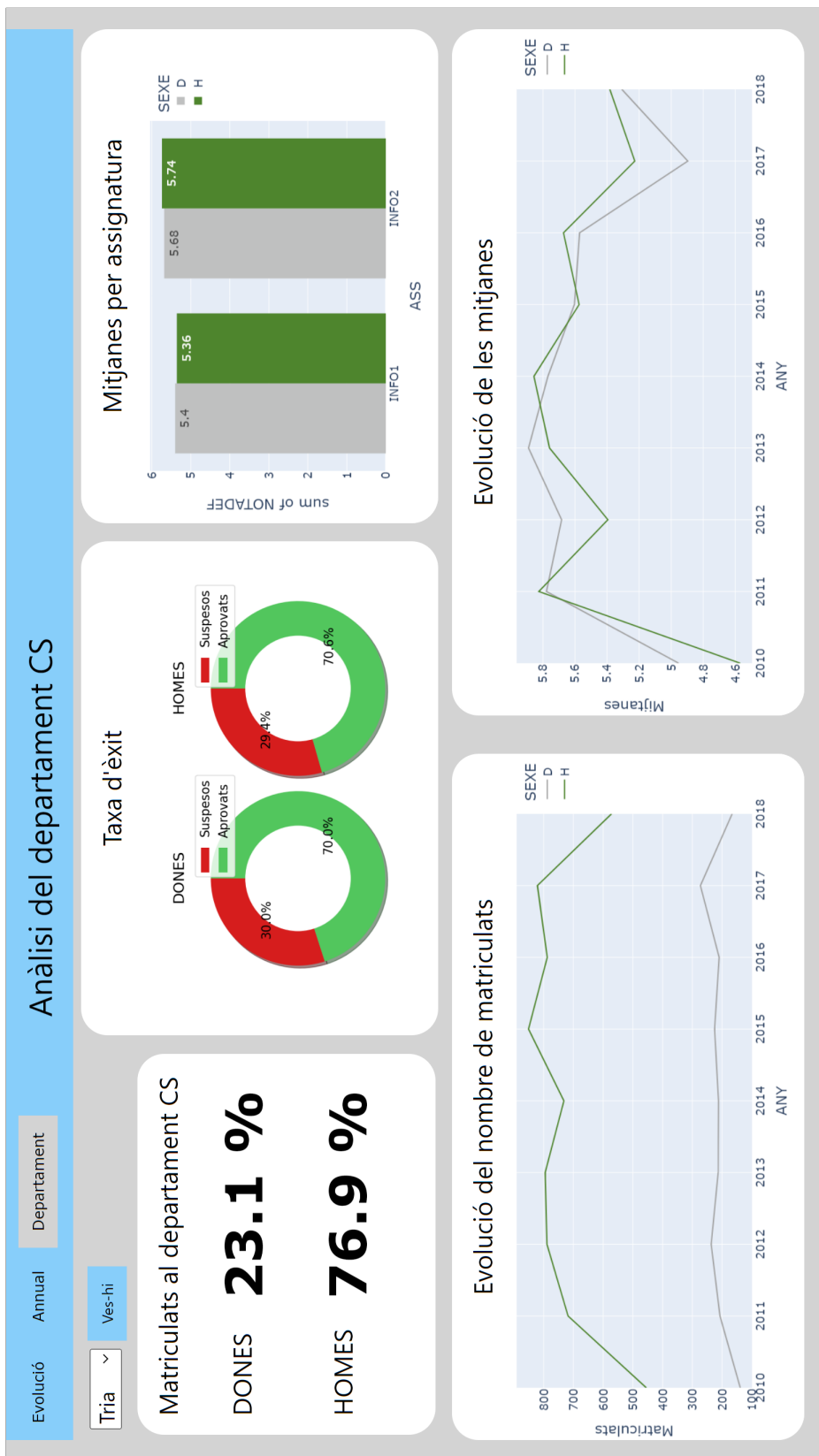


Figura 26: Dashboard Departament

evolucio.html Aquest fitxer és el *template* que retorna la pàgina d'evolució. Aquesta pàgina és estàtica, en el sentit que no té elements interactius i no canvia en funció de la tria del usuari. El codi d'aquesta pàgina doncs es limita a posicionar els elements com desitja l'usuari. Es pot observar el resultat final a la figura [27](#)

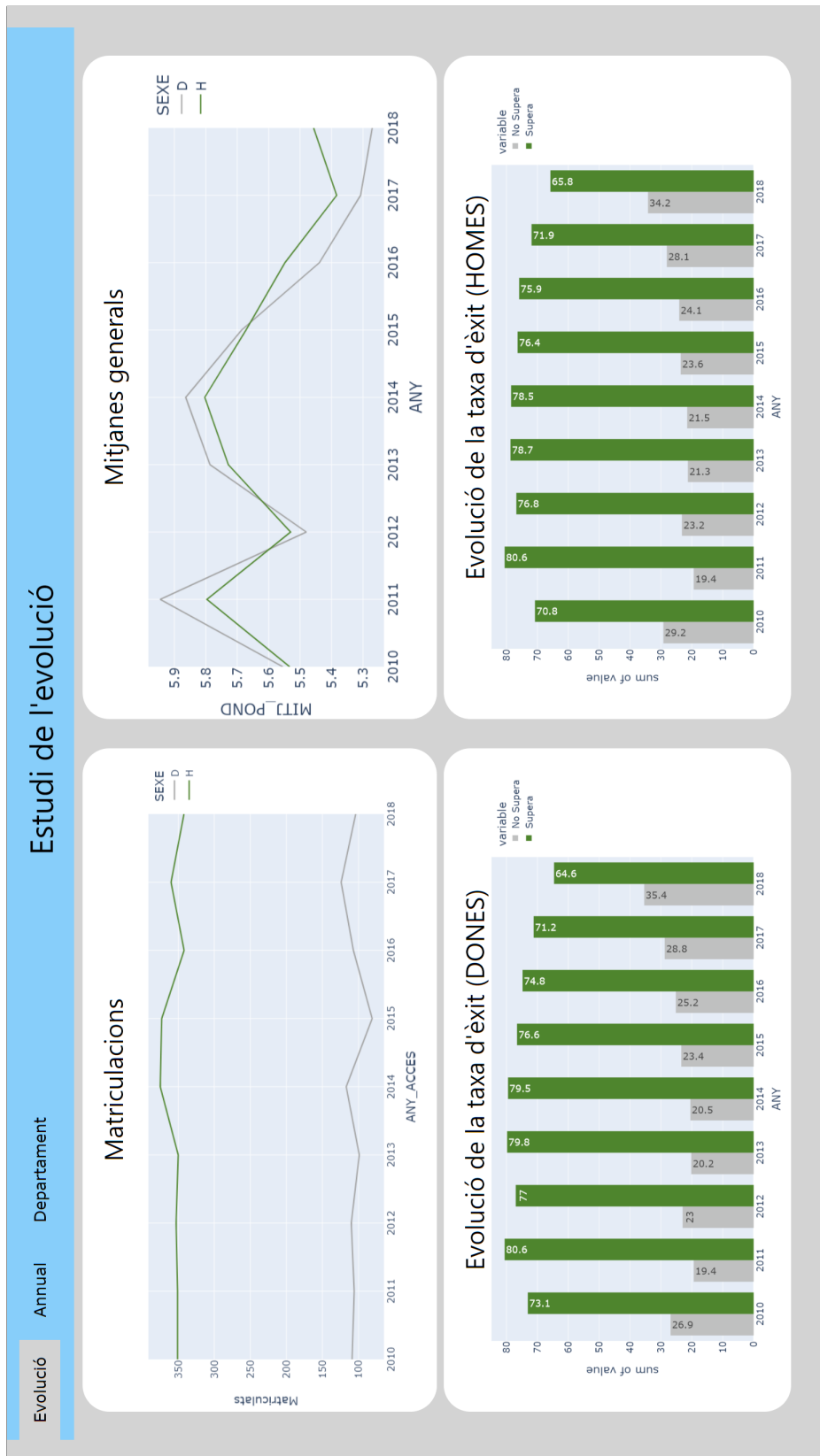


Figura 27: Dashboard Departament

7 Impacte en l'entorn

Aquest projecte pretén generar un impacte positiu sobre la població. Per això cal estudiar l'efecte que té sobre el seu entorn. Cal vigilar, doncs, les repercussions que pot tenir sobre el medi ambient, les persones i usuaris que impliquen directe o bé indirectament.

Primerament, es realitzarà un estudi de l'impacte social i d'igualtat de gènere, ja que és el camp més pertinent en aquest treball. A continuació, es durà a terme un estudi de l'impacte ambiental.

7.1 Impacte social i d'igualtat de gènere

Com s'ha mencionat al llarg de tot aquest treball, hi ha una clara diferència de presència entre homes i dones al món de la ciència. Tot i que aquests números semblen que van a l'alça, encara queda molta feina per a fer. Com s'ha mencionat en altres apartats del treball, la UPC ha presentat diversos plans d'igualtat al llarg dels anys. Per exemple, des de l'existència del GETI se n'han vist dos, un de 2013-2015 i un de 2016-2020 prorrogat al 2022. Tanmateix, com s'ha pogut observar a l'anàlisi d'aquest treball, la diferència del nombre de matriculats s'ha mantingut pràcticament constant des de l'inici del grau.

A més a més, aquesta diferència no s'acaba aquí. Posem per exemple el camp de la sanitat. Aquest sector, a escala d'entrada predominen les dones, amb un 66% de presència en treballadors de primera línia. No obstant això, si mirem el percentatge de treballadores que arriben a rangs més alts, el percentatge cau, fins a arribar a un 30% de dones. Es veu, doncs, que el problema va molt més enllà que la necessitat de promoure carreres STEM en les dones.

Es pretén que aquest treball permeti mostrar aquesta evident desigualtat a l'hora de triar una carrera de futur. És gràcies a l'ús d'eines com un *Dashboard* que es pot generar consciència d'aquests afers.

Anàlogament, aquesta mateixa eina pot ajudar a altres causes. El mateix programari, prèviament adaptat, pot servir per a donar veu a altres desigualtats socials i de gènere. La manera més efectiva de conscienciar, educar i actuar sobre una causa és mitjançant la presentació de dades.

7.2 Estudi ambiental

El 2019, la UPC va crear una xarxa d'associacions i persones de la comunitat universitària que comparteixen la lluita contra el canvi climàtic. [4] Aquest programa de consciència, educa els alumnes de la universitat en la importància de l'acció climàtica a més a més de donar veu a projectes de recerca que vetllen per la conservació del medi ambient.

Com totes les accions que fem, aquest treball ha tingut un impacte ambiental. En aquest apartat, s'analitza com i en què s'ha generat una petjada en el medi ambient.

Aquest treball ha estat realitzat des d'un ordinador, minimitzant el cost de matèria pri-

mera, com podria ser fulls o llibretes. Cal doncs tenir en compte l'energia consumida pel portàtil durant els mesos de realització del treball. Més endavant es fa una anàlisi elaborada d'aquest cost.

D'altra banda, el treball no ha comportat cap ús de matèria primera que necessiti ser reciclat o reutilitzat, ja que ha consistit d'elaboració de programari. Com que el malbaratament generat no va més enllà que alguns fulls emprats per la presa de notes, aquest impacte es pot considerar pràcticament nul.

Pel que fa a l'impacte ambiental que pot tenir el servidor, cal destacar que la pàgina web elaborada no està destinada a un públic massiu, sinó que, a hores d'ara, va encarada a un ús local i poc freqüent. Per tant, es podria dir que aquest impacte és mínim.

8 Pressupost

En aquest apartat es quantifica el cost d'aquest projecte. Tot i que aquest treball no ha suposat cap cost directe, a continuació s'analitzen els costos indirectes que ha suposat la realització d'aquest projecte.

8.1 Cost de personal

El cost de personal inclou el cost de l'autora i del tutor. L'autora hi ha dedicat un total de 345h al llarg del semestre. Considerant un cost de 15€/h queda un cost total de 5.175€.

8.2 Cost de llicències

Aquest treball s'ha fet emprant servidors de programari lliure, i per tant el cost de llicències ha estat nul. La memòria s'ha redactat amb L^AT_EX. El codi s'ha programat mitjançant Python, HTML, CSS i JS, tots d'accés lliure.

8.3 Cost de material

Com s'ha mencionat anteriorment, l'únic material emprat per a aquest projecte ha estat un ordinador portàtil. L'ordinador que s'ha utilitzat té un cost de 800€. S'ha fet servir durant els 4 anys de carrera, per tant, per als 5 mesos que ha durat la realització del treball, el cost associat és de 83,35€.

8.4 Cost energètic

La bateria de l'ordinador emprat (Surface Pro) pot durar fins a 13,5 hores amb una capacitat de 45 Wh. Per tant, el consum horari és de

$$P = \frac{W}{t} = \frac{45Wh}{13,5h} \approx 3,3W \quad (6)$$

Comptant que s'ha fet servir l'ordinador durant les 345h explicades prèviament, queda un consum de 1,1385 kWh. Considerant que el preu mitjà de l'elèctric avui, dia 21 de Juny del 2022 és de 0,36419€/kWh, queda un cost energètic total de 0,4146€

8.5 Cost total associat

Resumint, el cost total d'aquest projecte es pot veure a la taula 2. S'observa la suma desglossada de tots els costos mencionats anteriorment.

Es pot observar doncs, que el cost total del projecte és de 5.258,96 €

Agrupació	Concepte	Import [€]
Personal	Autoria	5175,00
Material	Ordinador	83,55
Energia	Ordinador	0,4146
Total		5258,9646

Taula 2: Cost total associat al projecte.

9 Planificació

A la figura 28 s'observa la planificació inicial del projecte. Les setmanes que apareixen en gris són setmanes que s'han descartat, les vacances de primavera a l'abril així com la primera setmana de juny, que correspon a la setmana on l'autoria va tornar de l'estada de mobilitat.

D'altra banda, es pot veure com la càrrega de feina les últimes setmanes és més elevada, ja que és quan es perfeccionen totes les parts del projecte, i quan cal treballar en diverses àrees simultàniament. A més a més, en aquestes dates, hi ha més disponibilitat i s'hi podran dedicar més hores.

Es pot veure també que l'última setmana s'ha deixat bastant buida, per tal d'acabar de repassar detalls i tenir temps de polir la memòria.

Al llarg del semestre, es fan reunions amb el tutor, que permeten anar avaluant l'evolució del treball així com guiar-ne el curs.

Aquest treball va començar amb un plantejament inicial que va anar acompanyat d'una pluja d'idees i un plantejament inicial d'objectius. Junt amb la tutoria del professor, es va anar construint una idea i un possible títol. Paral·lelament, amb l'ajuda de documentació, l'autoria es va anar informant de diversos aspectes d'aquest món per anar concretant aspectes.

Un cop el plantejament inicial es va començar, el procés es va complementar amb cerca de bibliografia i documentació, per tal de fer un plantejament informat. Dins d'aquesta documentació, s'inclouen les dades que van ser proporcionades pel tutor. En aquesta anàlisi exploratòria, s'inicia el filtratge de dades paral·lelament amb la cerca de documentació. Un cop les dades han estat estudiades, es pot començar a cercar elements que serviran per a la representació de dades, i que ens porten a un primer disseny preliminar.

La part de programació és la més llarga i més constant al llarg del quadrimestre. Comença d'hora, quan es comença amb l'anàlisi exploratòria, i és constant fins al final del treball. Es combina la programació Python amb el codi HTML segons necessitats del treball.

Finalment, la redacció de la memòria comença al principi, on es deixaran redactat apartats preliminars, com el marc teòric o els objectius. La resta de la memòria s'escriu un cop el treball comença a finalitzar-se, i es manté fins a l'hora d'entregar-la, ja que caldrà fer revisions i relectures.

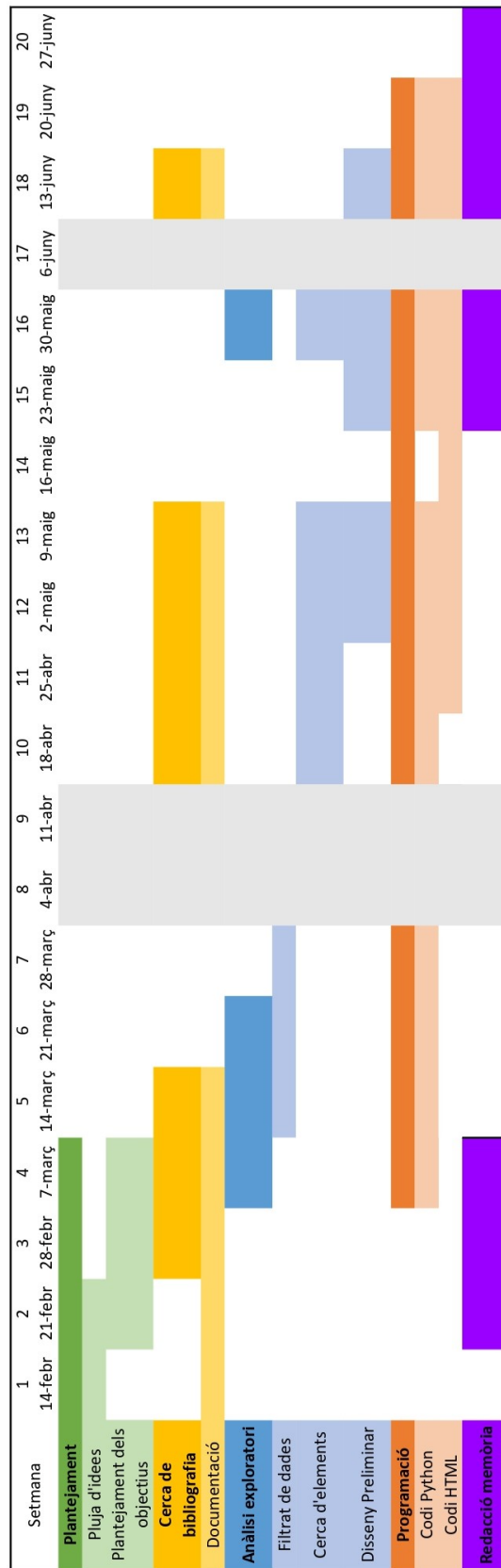


Figura 28: Diagrama de Gant del projecte.

Conclusions

Un cop finalitzat aquest projecte, es pot dir que s'han complert tots els objectius establerts. Al llarg d'aquest projecte s'ha desenvolupat una interfície web capaç de mostrar dades prèviament analitzades. Mitjançant l'ús de les dades de l'ETSEIB, s'ha realitzat un preprocés, filtració i anàlisi de dades. Aquest treball s'ha acompanyat de la programació d'una interfície web tipus *Dashboard* on s'han mostrat les dades analitzades.

La primera part de la programació del projecte ha consistit en l'anàlisi de dades. Respecte als objectius planejats inicialment, es pot dir que s'han complert. Les dades triades per aquest exemple concret han passat un procés de filtració i pre-procés. Posteriorment, s'han generat eines de visualització gràfica que han permès visualitzar aquest anàlisi. Al llarg del treball s'ha vist que aquests passos s'han de fer segons els requeriments de l'usuari.

Cal destacar que la metodologia usual de creació d'un *dashboard* amb anàlisi de dades es basa en la relació amb l'usuari o client. Els requeriments que posi aquest definiran el disseny i contingut de l'aplicació. En aquest cas, no s'ha contactat amb l'usuari sinó que s'ha fet servir el criteri de l'autoria. Malgrat tot, si aquest projecte necessites tenir en compte els requeriments de l'usuari, caldria simplement ajustar el disseny a les seves preferències i generar les eines de visualització demanades. El procés global seria el mateix i només canviarien petits detalls en la programació.

La part de programació front-end ha consistit en el disseny i creació d'un *dashboard*. Com s'ha explicat, s'ha triat aquesta eina per ser la millor manera de presentar l'anàlisi de dades. S'ha realitzat una cerca d'exemples per tal de trobar les millors eines i a continuació s'ha elaborat el codi propi, per tal de personalitzar al màxim la interfície.

Finalment, pel que fa als objectius personals, es considera que s'ha pogut profunditzar àmpliament en els coneixements previs de programació, complementant-los amb nous llenguatges. A més a més, s'ha millorat l'habilitat de resolució de problemes.

S'ha pogut mostrar a través d'un exemple pràctic que els dashboards son una eina de visualització de dades molt potent que permet fer arribar informació als usuaris. Aquests poden ajudar en la presa de decisions, a informar i educar. Analitzant l'exemple triat en aquest projecte, es podrien treure diverses conclusions. Primerament, s'observa una tendència estable en el nombre de matriculats tant a la universitat com als diversos departaments. Es pot veure també una gran semblança en els resultats dels alumnes independentment del gènere. Ara bé, s'observa una gran diferència entre el nombre de matriculats homes i dones.

10 Propostes de futur

Al llarg d'aquest projecte s'ha dissenyat i elaborat una pàgina web que permet analitzar dades i prendre decisions en base a aquestes. S'ha descrit al llarg de tot el treball que aquesta és una eina molt potent i molt versàtil. Es per això que deixa moltes possibles propostes de futur.

En un primer lloc es comenta la durabilitat d'aquest projecte. El disseny de la pàgina està fet de tal manera que es poden actualitzar les dades sempre que l'usuari ho desitgi, permetent així un ús indefinit de la interfície. Simplement cal afegir el fitxer a la carpeta corresponent, segons si son dades del grau o d'accés.

De fet, cal remarcar que amb aquestes mateixes dades, es poden realitzar altres anàlisis del rendiment, deixant de banda el gènere. Es podria emprar el mateix servidor web, i el filtrat de dades es podria realitzar també probablement. Simplement cal generar noves gràfiques en funció del punt de vista que se li vol donar a l'anàlisi.

Una altre manera de prorrogar l'ús d'aquest projecte, és adaptant-lo a altres bases de dades. Com s'ha explicat, aquí s'han fet servir les dades de l'ETSEIB com a exemple, però el treball vol presentar la versatilitat dels *dashboards* i entendre que amb aquesta mateixa interfície es poden generar milions d'anàlisi diferents.

D'altre banda, també cal destacar possibles millores del projecte. Deixant de banda els possibles canvis i millores estètiques que se li podrien fer, es poden realitzar altres tipus d'optimitzacions. Ara per ara, el codi del *back-end* es divideix en parts de pre-procés i procés en viu. Es podria realitzar un estudi de velocitats per esbrinar quina part del codi en viu triga més a processar, i estudiar la manera d'afegir-ho al codi preprocessat. Això estalvia temps i fa que la pàgina web carregui més ràpid un cop en marxa. A més a més, per aquest projecte s'ha treballat amb un sol servidor, però es podria fer un estudi de viabilitat i proposar treballar amb múltiples servidors, com s'ha comentat anteriorment en el marc teòric. Finalment, prèviament s'ha comentat la possibilitat d'afegir dades al servidor. De cara a una millora de futur, es podria elaborar una interfície que permeti penjar els fitxer sense haver de saber a quina carpeta cal que vagin.

Agraïments

Vull agrair aquest treball a tothom que m'ha donat suport i m'ha acompanyat.

Primer de tot agrair aquest treball a en Lluís Talavera, per accedir a fer-me de tutor malgrat una situació excepcional. Per adaptar-se a les meves condicions i per haver aconseguit que gaudeixi d'aquest projecte.

Moltes gràcies també a les persones que m'han acompanyat al llarg d'aquest projecte. Als meus pares i el meu germà per a donar-me suport i empenta quan han fet falta. Al Gerard i els companys de biblioteca per a guiar-me i acompanyar-me aquest últim més, i a l'Emma, la Mar, el Jordi, el Joan i el David per ser-hi, no només aquest últim semestre però tota la carrera.

Sense tots vosaltres, aquest treball hauria estat molt més dur, i no s'hauria pogut arribar tant lluny. Un cop més, agrair-vos tot el que m'heu donat.

Bibliografia

- [1] Vishesh Arora. *Exploratory Data Analysis (EDA) – A step by step guide*. URL: https://en.wikipedia.org/wiki/Exploratory_data_analysis. [en línia] (Accés el 23/1/2022).
- [2] Universitat Politècnica de Catalunya. *Dades estadístiques i de Gestió*. URL: https://gpaq.upc.edu/lldades/indicador.asp?index=1_1_5. [en línia] (Accés el 9/6/2022).
- [3] Universitat Politècnica de Catalunya. *Pla d'Igualtat de la UPC*. URL: <https://igualtat.upc.edu/ca/pla>. [en línia] (Accés el 10/6/2022).
- [4] Universitat Politècnica de Catalunya. *UPC Clima*. URL: <https://upcclima.upc.edu/qui-som>. [en línia] (Accés el 21/6/2022).
- [5] IBM Cloud Education. *Big Data analytics*. URL: <https://www.ibm.com/uk-en/analytics/hadoop/big-data-analytics>. [en línia] (Accés el 23/1/2022).
- [6] IBM Cloud Education. *Data Mining*. URL: <https://www.ibm.com/cloud/learn/data-mining>. [en línia] (Accés el 23/1/2022).
- [7] IBM Cloud Education. *Data Science Introduction*. URL: <https://www.ibm.com/cloud/learn/data-science-introduction>. [en línia] (Accés el 23/1/2022).
- [8] IBM Cloud Education. *Machine Learning*. URL: <https://www.ibm.com/cloud/learn/machine-learning>. [en línia] (Accés el 23/1/2022).
- [9] R Graph Gallery. *Scatterplot*. URL: <https://r-graph-gallery.com/scatterplot.html>. [en línia] (Accés el 23/6/2022).
- [10] Miguel Grinberg. *Flask web development: developing web applications with python*. "O'Reilly Media, Inc.", 2018.
- [11] Charles R. Harris et al. "Array programming with NumPy". A: *Nature* 585.7825 (set. de 2020), pàg. 357-362. DOI: 10.1038/s41586-020-2649-2. URL: <https://doi.org/10.1038/s41586-020-2649-2>.
- [12] J. D. Hunter. "Matplotlib: A 2D graphics environment". A: *Computing in Science & Engineering* 9.3 (2007), pàg. 90-95. DOI: 10.1109/MCSE.2007.55.
- [13] The pandas development team. *pandas-dev/pandas: Pandas*. Vers. latest. Febr. de 2020. DOI: 10.5281/zenodo.3509134. URL: <https://doi.org/10.5281/zenodo.3509134>.
- [14] John W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.
- [15] Cambridge University. *Cambridge Dictionary*. URL: <https://dictionary.cambridge.org/es/diccionario/ingles/drop-down-menu?q=drop+down>. [en línia] (Accés el 23/6/2022).
- [16] Cambridge University. *Cambridge Dictionary*. URL: <https://dictionary.cambridge.org/es/diccionario/ingles/file>. [en línia] (Accés el 23/6/2022).
- [17] Cambridge University. *Cambridge Dictionary*. URL: <https://dictionary.cambridge.org/es/diccionario/ingles/framework>. [en línia] (Accés el 23/6/2022).
- [18] Cambridge University. *Cambridge Dictionary*. URL: <https://dictionary.cambridge.org/es/diccionario/ingles/input>. [en línia] (Accés el 23/6/2022).
- [19] Cambridge University. *Cambridge Dictionary*. URL: <https://dictionary.cambridge.org/es/diccionario/ingles/loop>. [en línia] (Accés el 23/6/2022).

- [20] Cambridge University. *Cambridge Dictionary*. URL: <https://dictionary.cambridge.org/es/diccionario/ingles/output>. [en línia] (Accés el 23/6/2022).
- [21] Cambridge University. *Cambridge Dictionary*. URL: <https://dictionary.cambridge.org/es/diccionario/ingles/script>. [en línia] (Accés el 23/6/2022).
- [22] Cambridge University. *Cambridge Dictionary*. URL: <https://dictionary.cambridge.org/es/diccionario/ingles/slider>. [en línia] (Accés el 23/6/2022).
- [23] Cambridge University. *Cambridge Dictionary*. URL: <https://dictionary.cambridge.org/es/diccionario/ingles/static>. [en línia] (Accés el 23/6/2022).
- [24] Cambridge University. *Cambridge Dictionary*. URL: <https://dictionary.cambridge.org/es/diccionario/ingles/web-server>. [en línia] (Accés el 23/6/2022).
- [25] Cambridge University. *Cambridge Dictionary*. URL: <https://dictionary.cambridge.org/es/diccionario/ingles/widget>. [en línia] (Accés el 23/6/2022).
- [26] Johns Hopkins University. *COVID-19 Map- Johns Hopkins University*. URL: <https://coronavirus.jhu.edu/map.html>. [en línia] (Accés el 10/6/2022).
- [27] Guido Van Rossum i Fred L Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.
- [28] Wikipedia. *Exploratory Data Analysis*. URL: https://en.wikipedia.org/wiki/Exploratory_data_analysis. [en línia] (Accés el 23/1/2022).
- [29] Society of Women Engineer. *Research Update*. URL: <https://alltogether.swe.org/2019/11/swe-research-update-women-in-engineering-by-the-numbers-nov-2019/>. [en línia] (Accés el 9/6/2022).