

## Treball final de grau

**Estudi:** Grau en Enginyeria Informàtica

**Títol:** Visualització de l'evolució de curses

**Document:** Memòria

**Alumne:** Luis Alejandro Leon Corcuera

**Tutor:** Marta Fort Masdevall

**Departament:** Informàtica, Matemàtica Aplicada i Estadística (IMAE)

**Àrea:** Llenguatges i Sistemes Informàtics (LSI)

**Convocatòria (mes/any):** Juny 2020

# Índex

1	Introducció . . . . .	<b>9</b>
1.1	Motivacions . . . . .	9
1.2	Propòsit i objectius del projecte . . . . .	11
2	Estudi de viabilitat . . . . .	<b>12</b>
3	Metodologia . . . . .	<b>14</b>
4	Planificació . . . . .	<b>16</b>
5	Marc de treball i conceptes previs . . . . .	<b>18</b>
5.1	Marc de treball . . . . .	18
5.2	Conceptes previs . . . . .	18
5.2.1	Curses de ciclistes . . . . .	18
5.2.2	Visualització de curses . . . . .	21
5.2.3	Definició de grup . . . . .	21
5.2.4	Disseny d'una pàgina web . . . . .	23
5.2.5	Servidor local vs Servidor en el Cloud vs Servidor Remote . . . . .	23
5.2.6	W3Schools . . . . .	24
5.2.7	D3.js . . . . .	24
5.2.8	Paleta de colors . . . . .	26
6	Requisits dels sistema . . . . .	<b>28</b>
6.1	Requisits funcionals . . . . .	28
6.2	Requisits no funcionals . . . . .	29
7	Estudis i decisions . . . . .	<b>30</b>
7.1	Llenguatges . . . . .	30
7.1.1	JavaScript . . . . .	30
7.1.2	HTML . . . . .	30
7.1.3	CSS vs Bootstrap . . . . .	31
7.2	Llibreries . . . . .	31
7.2.1	jQuery . . . . .	31
7.2.2	D3 vs Qt . . . . .	32
7.3	Programari utilitzat . . . . .	32
7.3.1	Sublime Text . . . . .	33

7.3.2	GitHub . . . . .	33
7.3.3	Wavi . . . . .	34
7.3.4	GanttProject . . . . .	34
7.3.5	Draw.io . . . . .	34
7.3.6	Skype . . . . .	35
7.3.7	Overleaf . . . . .	35
7.3.8	Color Brewer . . . . .	36
8	Anàlisi i disseny del sistema . . . . .	<b>37</b>
8.1	Anàlisi i disseny de la interfície usuari . . . . .	37
8.1.1	Anàlisi dels requeriments . . . . .	39
8.2	Anàlisi i disseny dels elements de la gràfica . . . . .	48
8.2.1	Punts quilomètrics . . . . .	48
8.2.2	Trajectòries . . . . .	48
8.2.3	Representació del grups . . . . .	53
8.2.4	Informació temporal . . . . .	56
8.2.5	Perfil de l'etapa . . . . .	58
8.2.6	Informació d'una trajectòria . . . . .	59
8.2.7	Zoom de la pàgina . . . . .	60
8.3	Anàlisi i disseny del codi . . . . .	62
8.3.1	Diagrama de classes . . . . .	62
8.4	Fitxer d'entrada . . . . .	71
9	Implementació i proves . . . . .	<b>73</b>
9.1	Implementacions . . . . .	73
9.1.1	Generar els grups de ciclistes . . . . .	73
9.1.2	Generació dels polígons per representar els grups . . . . .	74
9.1.3	Actualitzar posicions dels elements al fer zoom . . . . .	74
9.1.4	Trobar els punts quilomètrics a mostrar . . . . .	77
9.1.5	Visualització en diferents navegadors . . . . .	78
9.2	Proves . . . . .	80
9.2.1	Obtenir dades del servidor . . . . .	80
9.2.2	Ordre d'execució dels fitxers del projecte . . . . .	81
9.2.3	Mostrar mallot d'un ciclista líder . . . . .	82
9.2.4	Filtrar per nom de ciclista o equip . . . . .	83
9.2.5	No hi han dades d'etapa o ciclista en l'etapa actual . . . . .	83

---

9.2.6	Mostrar punts quilomètrics . . . . .	84
10	Implantació i resultats . . . . .	<b>86</b>
10.1	Implantació d'un servidor local . . . . .	86
10.2	Lectura dels fitxers d'entrada . . . . .	88
10.3	Resultats dels objectius assolits . . . . .	88
10.3.1	Visualització de la pàgina . . . . .	88
10.3.2	Visualització dels grups . . . . .	89
10.3.3	Seleccionar un ciclista o trajectòria . . . . .	91
10.3.4	Seleccionar un equip . . . . .	92
10.3.5	Aplicar zoom a la gràfica . . . . .	93
10.3.6	Comparar dos ciclistes . . . . .	94
11	Conclusions . . . . .	<b>96</b>
12	Treball futur . . . . .	<b>97</b>
13	Bibliografia . . . . .	<b>99</b>
14	Annexos . . . . .	<b>101</b>
14.1	Generar grups de ciclistes . . . . .	101
14.2	Generació dels polígons . . . . .	102
14.3	Comportament zoom . . . . .	103
14.4	Trobar punts quilomètrics a mostrar . . . . .	105
15	Manual d'usuari i/o instal·lació . . . . .	<b>107</b>
15.1	Seleccionar una etapa de la cursa . . . . .	107
15.2	Fer zoom en la gràfica . . . . .	107
15.3	Seleccionar una trajectòria . . . . .	108
15.4	Seleccionar dues trajectòries . . . . .	108
15.5	Seleccionar un ciclista . . . . .	109
15.6	Seleccionar un equip . . . . .	109
15.7	Seleccionar dos ciclistes . . . . .	110
15.8	Desmarcar les trajectòries i elements seleccionats . . . . .	110

## Índex de figures

1	Diagrama de flux del mètode Scrum . . . . .	14
2	Diagrama de Grantt de la planificació del projecte . . . . .	17
3	Exemple del transcurs del temps respecte el primer (1) . . . . .	19
4	Exemple del transcurs del temps respecte el primer (2) . . . . .	20
5	Exemple de l'evolució del grups (extreta de [1]) . . . . .	22
6	Exemple canviar color dels paràgraf amb W3C DOM API . . . . .	25
7	Exemple canviar color dels paràgraf amb D3.js . . . . .	25
8	Exemple canviar color dels paràgraf aleatòriament amb D3.js . . . . .	25
9	Exemple canviar color del fons i modificar el radi dels cercles . . . . .	26
10	Pàgina web de Color Brewer . . . . .	27
11	Icona de JavaScript . . . . .	30
12	Icona de HTML . . . . .	30
13	Icona de CSS . . . . .	31
14	Icona de Bootstrap . . . . .	31
15	Icona de jQuery . . . . .	31
16	Icona de D3 . . . . .	32
17	Icona de Qt . . . . .	32
18	Icona de Sublime . . . . .	33
19	Icona de GitHub . . . . .	33
20	Icona de Wavi . . . . .	34
21	Icona de GanttProject . . . . .	34
22	Icona de Draw.io . . . . .	35
23	Icona de Skype . . . . .	35
24	Icona de l'Overleaf . . . . .	35
25	Icona de Color Brewer . . . . .	36
26	Esbós de la visualització de curses . . . . .	37
27	Diagrama cas d'ús de Gestionar etapes . . . . .	39
28	Fitxa de cas d'ús de Seleccionar etapa . . . . .	39
29	Diagrama cas d'ús de Buscar ciclista . . . . .	40
30	Fitxa de cas d'ús de Buscar ciclista . . . . .	40
31	Diagrama cas d'ús de Buscar equip . . . . .	41
32	Fitxa de cas d'ús de Buscar equip . . . . .	41
33	Diagrama cas d'ús de Visualitzar ciclista . . . . .	42

34	Fitxa de cas d'ús de Seleccionar trajectòria . . . . .	42
35	Fitxa de cas d'ús de Seleccionar un ciclista . . . . .	42
36	Diagrama cas d'ús de Visualitzar un equip . . . . .	43
37	Fitxa de cas d'ús de Seleccionar equip . . . . .	43
38	Diagrama cas d'ús de Mostrar elapsed time . . . . .	44
39	Fitxa de cas d'ús de Mostrar retard . . . . .	44
40	Diagrama cas d'ús de Comparar dos ciclistes . . . . .	45
41	Fitxa de cas d'ús de Comparar ciclistes . . . . .	45
42	Diagrama cas d'ús de Fer zoom . . . . .	46
43	Fitxa de cas d'ús de Fer zoom . . . . .	46
44	Diagrama cas d'ús de Fer drag . . . . .	47
45	Fitxa de cas d'ús de Fer drag . . . . .	47
46	Exemple de les trajectòries de cada ciclista en la web . . . . .	48
47	Exemple de les coordenades cartesianes de cada ciclista en la web . . . . .	49
48	Gràfica de les trajectòries de cada ciclista, v1 . . . . .	50
49	Exemple del transcurs del temps respecte el primer . . . . .	51
50	Gràfica de les trajectòries de cada ciclista, v2 . . . . .	51
51	Exemple de diagrama Sankey . . . . .	53
52	Primera versió de la representació dels grups . . . . .	54
53	Exemple de polígon amb v1 . . . . .	54
54	Segona versió de la representació dels grups . . . . .	55
55	Exemple de polígon amb v2 . . . . .	55
56	Exemple de polígon amb v3 . . . . .	56
57	Petit algorisme per trobar les etiquetes. . . . .	57
58	Exemple d'etiquetes entre els grups. . . . .	57
59	Exemple de <i>brush</i> en un diagrama amb D3. . . . .	58
60	Exemple de fer <i>hover</i> sobre d'alguna trajectòria . . . . .	59
61	Ciclista seleccionat i tooltip activat . . . . .	60
62	Equip seleccionat i tooltip activat . . . . .	60
63	Diagrama de classes del fitxer index.html . . . . .	63
64	Diagrama de classes del fitxer stages.js . . . . .	63
65	Diagrama de classes del fitxer riders-info.js . . . . .	64
66	Diagrama de classes del fitxer riders-graph.js (1) . . . . .	65
67	Diagrama de classes del fitxer riders-graph.js (2) . . . . .	66
68	Diagrama de classes del fitxer riders-graph.js (3) . . . . .	66

69	Diagrama de classes del fitxer riders-graph.js (4)	67
70	Diagrama de classes del fitxer interaction.js (1)	68
71	Diagrama de classes del fitxer interaction.js (2)	68
72	Diagrama de classes del fitxer interaction.js (3)	69
73	Diagrama de classes del fitxer interaction.js (4)	69
74	Format d'entrada per els diferents fitxers	71
75	Exemple fitxer d'entrada de les etapes	72
76	Pseudocodi de generar els grups de ciclistes	73
77	Pseudocodi d'obtenir els poligons d'una etapa	74
78	Pseudocodi del comportament del zoom	77
79	Pseudocodi trobar els punts quilomètrics a mostrar	78
80	Visualització de l'aplicació en Firefox	79
81	Visualització de l'aplicació en Chrome	79
82	Visualització de l'aplicació en Safari	80
83	Captura de la terminal quan obté les dades	81
84	Ciclista líder	82
85	Ciclista no líder	82
86	Filtrar ciclista per nom	83
87	Filtrar equip per nom	83
88	Alert no existeixen dades de l'etapa seleccionada	84
89	Alert no participa el ciclista en l'etapa actual	84
90	Mostrar punts quilomètrics amb diferent distància entre trams (1)	85
91	Mostrar punts quilomètrics amb diferent distància entre trams (2)	85
92	Exemple d'instal·lació de Python	87
93	Exemple d'executar el servidor local amb Windows	87
94	Exemple d'executar el servidor local amb un Mac OS X	87
95	Visualització de la pàgina web	89
96	Visualització dels grups que es divideixen	90
97	Visualització dels grups que es fusionen	90
98	Visualització dels grups que es continuen	91
99	Visualització de seleccionar un ciclista	92
100	Visualització de seleccionar un equip	93
101	Exemple de zoom amb un equip seleccionat	94
102	Visualització de comparar dos ciclistes	95
103	Seleccionar etapa (1)	107

104	Seleccionar etapa (2) . . . . .	107
105	Gràfica sense fer zoom . . . . .	108
106	Gràfica fent zoom . . . . .	108
107	Seleccionar dues trajectòries . . . . .	109



# 1 Introducció

Durant el transcurs de la història les curses s'han convertit en una gran influència per tots nosaltres, directa o indirectament. De tots els tipus de curses que hi ha, les curses de llarga distància són aquelles en les que s'impliquen un gran nombre de corredors, aquesta distància es pot dividir en diferents trams, durant tots aquests trams hi han punts de control que tenen la funció d'analitzar l'estat del corredor i l'evolució que té durant la cursa. En aquests punts es pot fer un registre per exemple: de la velocitat mitjana que porta, el ritme (minuts necessaris per córrer un quilòmetre o milla), la cadència uniforme (nombre de passos per minut), entre altres dades de les que un pugui estar interessat.

Gràcies al paper *Group evolution patterns in running races* [1] amb la contribució de Y. Diez, M. Fort, M. Korman i J.A. Sellarès, s'ha fet una definició teòrica de quina mena de grups es poden formar durant una cursa, cada grup com és evident està compost per diferents concursants, i a més a més pot arribar a comportar-se de diferents formes: sobreviu, apareix, desapareix, s'expandeix, etc.. Són diferents patrons que estan ben definits en el paper esmentat abans. L'objectiu final és fer un anàlisi de: els diferents grups que es formen, la interacció que hi han entre aquests grups i entre els corredors de la cursa. En aquest paper es presenten diferents algorismes molt potents capaços d'executar-se en temps real sense que el temps d'execució afecti els resultats.

A conseqüència d'aquest estudi, en el departament d'IMAE, la doctora Marta Fort Masdevall <sup>1</sup>, juntament amb dos professors externs Pere Pau Vázquez <sup>2</sup> i Jose Diaz Iriberry <sup>3</sup>, de la UPC i UVic-UCC respectivament, va sorgir la idea de fer un anàlisi dels participants d'una cursa de recorregut prefixat en la que pogués haver-hi més d'una etapa amb participants potencialment diferents. Aquest interès es deu a que en Jose és un corredor habitual d'aquest tipus de cursa i juntament amb en Pere-Pau són experts en la visualització de les dades i la Marta tenia un article previ [1] sobre la detecció de grups en una etapa de curses d'aquest tipus. El projecte consisteix en utilitzar tota informació d'una cursa real, importar-la en una pàgina web, juntament amb la col·laboració de la meva tutora Marta Fort, amb la que es podrà fer un anàlisi més visual i explícit de totes les dades.

## 1.1 Motivacions

Els motius que han fet que em decidís en fer aquest projecte, són la meva passió per córrer, per les curses d'obstacles i per l'esport en general. Ja d'un bon principi tenia pensat que el meu projecte estigués una mica relacionat amb l'esport. Quan es va organitzar la xerrada del PFG, els diferents professors de

---

<sup>1</sup><http://ima.udg.edu/mfort/>

<sup>2</sup><https://www.cs.upc.edu/ppau/>

<sup>3</sup><https://www.cs.upc.edu/jdiriberri/>

GEINF van fer diferents propostes de projectes, relacionats amb els diferents itineraris que té la carrera. Entre les diferents possibilitats de projectes estava aquesta, la visualització de diferents etapes d'una cursa ciclista, projecte que va presentar la Dra. Marta Fort, on més endavant arribaria a ser la tutora del meu PFG. Si estava relacionat amb l'esport i a més a més podia col·laborar en una investigació millor que millor. Un altre aspecte a mencionar és que es tractava d'una pàgina web i sincerament no era un dels meus punts forts, però tampoc em desagradava la idea, com a repte personal era perfecte.

Vaig trobar la proposta interessant perquè, en l'actualitat hi ha dades de moltes curses, i l'anàlisi d'aquestes dades ajudaria a donar informació als participants sobre la seva evolució. Una pàgina web ajudaria molt a la visualització de les dades i a un futur anàlisi dels resultats. Per exemple, un atleta fent un anàlisi a posteriori de la carrera pot comprovar quina ha sigut la seva evolució durant la cursa, comparant les seves sensacions amb les dades reals d'aquell moment. Això pot fer el corredor en entrenaments futurs pugui treballar més en les seves debilitats fent que el resultats millorin considerablement.

També pot ser interessant pels organitzadors de les curses de manera que puguin trobar certes coincidències en moments concrets de la cursa. Per exemple, si analitzen una etapa de muntanya i en un punt de control quilomètric veuen que es formen dos o més tipus de grups diferents, dels quals uns segueixen el mateix ritme, però uns altres es van endarrerint. Caldria comprovar quin són els motius que fan que perdin el ritme. És possible que sigui un factor físic, per el gran desnivell de la muntanya, però també hi poden haver altres factors, especialment si es tracta d'una part plana. Podria ser, per exemple que la carretera fos massa estreta o que hi hagués un tram en què hi passin participants en els dos sentits de la marxa i que al coincidir corredors en els dos sentits faci alentir-los.

Per part d'un equip podria arribar a servir per a detectar problemes en el material de les bambes, rodes, del seient, de l'uniforme dels participants, o factors meteorològics, és possible que en aquell moment hagués plogut o hagués aparegut un cert tipus de vent. Pot arribar a utilitzar-se per dissenyar estratègies per edicions futures en el mateix recorregut, assumint que el desenvolupament de la cursa serà aproximadament el mateix, es poden planejar escapades o canvis en el ritme per no despenjar-se d'un grup o atrapar-ne un altre. També pot ajudar a detectar participants que fan el paper de *llebre* entre altres estratègies de grup no permeses.

És a dir, pot ajudar a arribar a la conclusió de que no tots els participants tenen les mateixes condicions en un moment puntual de la cursa. I com és en el cas de les curses de llargues distàncies, la suma de tots aquests esdeveniments pot arribar a generar uns resultats molt diferents al final de la cursa. Mirar de minimitzar aquesta diferència de condicions pot ser interessant pels organitzadors i per a la majoria

de participants. Així els únics paràmetres que farien decidir qui guanya la competició serien els factors físics i psicològics.

Així que tots aquest factors com realitzar un projecte en què es treballés amb dades esportives, l'aplicabilitat que veia en la pàgina web, el fet de poder aprendre web juntament amb la visualització de les dades i poder treballar en un projecte de recerca va fer que em decantés per aquest projecte.

## 1.2 Propòsit i objectius del projecte

La idea principal del projecte es pot utilitzar per a representar dades de qualsevol tipus de cursa (d'atletisme, ciclisme, raids de cavalls, de motos, cotxes, etc.), però per començar es tractaran dades d'una cursa ciclista, si ha sigut així era perquè eres les dades que disposàvem en aquell moment i a més a més, comptàvem amb en Jose, que té grans coneixement sobre aquestes dades, gràcies a la seva afició de fer curses. Amb aquest requisit el propòsit del projecte és implementar una pàgina web capaç de reflectir tota l'evolució dels ciclistes durant una cursa. D'aquesta cursa, mostrar els ciclistes i cada un dels diferents grups que es formen durant els punts de control que hi ha en les etapes de la carrera, de tal manera que sigui fàcil d'interactuar, facilitant així la futura recerca de possibles patrons de moviments que vagin apareixent durant la cursa.

Com a objectiu, volem representar tota la informació que tenim d'una cursa ciclista formada per diferents etapes de la millor manera possible. Volem poder-hi representar tant els ciclistes com els equips participants, així com la seu comportament al llarg de l'etapa. Per això volem poder representar els diferents grups de ciclistes que es formen al llarg d'una etapa de manera que també es vegi l'evolució dels diferents grups. Per altra banda també volem poder visualitzar la classificació, totes les dades d'un ciclista o equip i finalment poder comparar l'evolució de dos ciclistes concrets. Tot això ho volem fer de facilitant la interacció amb l'usuari per tal de que pugui visualitzar les dades que més li convinguin. Volem fer-ho de manera que visualitzar una etapa sigui ràpid per tal de que si mai es poden obtenir dades en temps reals es pugui anar mostrant amb els mateixos algorismes en temps real.

Per a poder-ho aconseguir cal fer un disseny previ de la pàgina web, decidir la millor estructura de dades possible i generar els algorismes que facin falta per fer la web. Cal determinar com visualitzar les dades per tal de que es vegi l'evolució temporal de la cursa. Cal representar visualment la generació dels diferents grups que es vagin formant al llarg de la cursa, però també ha de quedar representada visualment la seva evolució. Finalment també caldrà establir quina ha de ser la interacció de l'usuari amb la web.

## 2 Estudi de viabilitat

La viabilitat d'aquest projecte comença en obtenir les dades d'una cursa cursa per a poder-les visualitzar. Existeixen totes les dades del Tour de France 2018 a la web<sup>4</sup>. Per tant disposem de dades reals amb les quals treballar. Respecte a les dades de les etapes de la cursa només disposem de 3 d'elles (les etapes 11, 13 i 18). Vull agrair aquest fet al Dr. Jose Diaz, ja que va ser ell qui va fer una col·lecta d'aquestes dades mitjançant la web Pro Cycling Stats [5]. Actualment no disposem de les eines necessàries per a fer-lo funcionar en temps real durant el desenvolupament d'una cursa. Per això aquest aspecte es tradueix, tal i com es comenta a la secció anterior, a visualitzar les dades de forma ràpida i eficient perquè més endavant, si escau, es puguin visualitzar dades durant el transcurs d'una cursa.

Un altre factor molt important era trobar una eina que fos capaç d'implementar la visualització i també la interacció amb l'usuari. És aquí on el *core* de l'aplicació apareix, facilitant-nos una mica la feina, parlem d'una llibreria de JavaScript anomenada D3, aquesta llibreria ens ajudarà a fer la pàgina web, ja que està pensada per llegir unes dades i a partir d'aquestes poder produir uns gràfics dinàmics, amb la possibilitat d'interaccionar amb l'usuari. Més endavant argumentarem perquè aquesta llibreria i no altres opcions com per exemple el Qt, framework multiplataforma orientada a objectes i amb una interfície gràfica. També s'utilitzarà la llibreria jQuery, que ens ajudarà a manipular millor els objectes i les funcionalitats de la pàgina web.

Un altre dels recursos necessaris era utilitzar un servidor, en el que es pogués fer crides de tipus *GET* que et retornessin informació dels ciclistes, equips i dades de la cursa. Per facilitar-me la feina he optat per utilitzar un servidor local que simula aquest servei. Per tenir aquest petit servidor vaig decidir utilitzar un mòdul de *Python* anomenat *SimpleHTTPServer*, un senzill servidor al que es pot accedir a través de *localhost*. Aquesta decisió comporta un avantatge i un inconvenient: la part positiva és que en aquest projecte hi hauran peticions asíncrones i sinó s'utilitzés aquest servidor local aquesta funcionalitat no hi podria ser. Per això es recomana l'utilitzar d'un servidor [2]. La part negativa d'aquesta decisió és que aquest servidor no admet crides de tipus *POST* [3], aquest inconvenient no té cap efecte negatiu ja que en el nostre cas no necessitem modificar les dades de la cursa, ja que donem per suposat que s'han obtingut correctament. Però si de cares al futur, per exemple, es volgués utilitzar aquesta aplicació amb diferents usuaris, s'hauria d'adaptar el projecte per implementar aquest canvi.

Tots aquests recursos esmentats no tenen cap tipus de cost, ja que són de codi lliure i no tenen cap cost econòmic. Per altra banda podem calcular el pressupost econòmic que hagués costat contractar a una persona per realitzar aquest projecte, partint dels honoraris següents: 15€/hora aproximadament. La

---

<sup>4</sup><https://www.procyclingstats.com/race/tour-de-france/2018/gc/stages/all-stage-profiles>

feina final tindria un cost aproximat de:

<b>Pressupost inicial</b>		
<b>Tasca</b>	<b>Hores</b>	<b>Cost(€)</b>
Estudi del treball previ ja desenvolupat	15	225
Investigació de la viabilitat del projecte	15	225
Estudi de les llibreries necessàries	30	450
Disseny de la pàgina web	20	300
Desenvolupament de la pàgina web	300	4500
Documentació del projecte	70	1050
<b>TOTAL</b>	<b>450</b>	<b>6750</b>

Pressupost inicial sobre el cost del projecte.

### 3 Metodologia

La organització ha sigut primordial per arribar a satisfer les necessitats del projecte dintre del plaç encomanat. La metodologia que s'ha arribat a terme bé arrel de les idees bàsiques del que es coneix com a *Scrum* [4]. A grans trets *Scrum* és una metodologia de treball en la que és divideix un projecte en petites tasques, on cada una d'aquestes pot ser desenvolupada per qualsevol membre que estigui participant en l'aplicació, els quals poden tenir diferents rols. *Scrum* també implementa una nova idea de flux treball, en la que divideix tota la duració del projecte en diferents etapes, que anomena *Sprint*. Dintre de cada un d'aquests *Sprints*, que s'aniran tancant cronològicament, apareixen un conjunt de tasques que s'haurien de finalitzar abans de que s'acabés l'*Sprint*, si no fos així, la tasca es posposaria al següent.

Els beneficis d'aquesta pràctica són la gran flexibilitat que aporta durant el projecte, per exemple el client pot comprovar certes funcionalitats abans de que tot el projecte finalitzi, donant la possibilitat de fer canvis si fos necessaris. Un altre benefici a esmentar és l'augment de productivitat, al haver-hi bastants participants fa que l'evolució del projecte tingui un bon ritme, ja que teòricament cada una de les tasques és independent de les altres i per tant cada membre pot avançar sense interferir-se amb els altres.

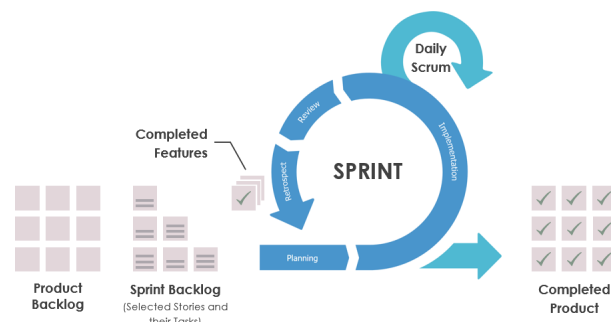


Figura 1: Diagrama de flux del mètode Scrum

Scrum està pensat per projectes on hi participin diferents persones, però això no vol dir que no es puguin utilitzar les idees principals d'aquesta metodologia i implementar-les en el nostre. En aquest cas les idees que s'han vist reflectides més són l'organització de tasques, l'idea de Sprint i el que s'anomena *to do*, *in progress* i *done*. Organitzar totes aquestes tasques en petits objectius setmanals i donar-los per finalitzats, pot comportar una gran tranquil·litat psicològica. Ja que a mesura que avança el temps, observes i ets conscient de com la pàgina web va agafant forma.

Pel que fa a la participació del projecte han estat involucrats els doctors Jose Diaz Iriberry i en Pere Pau Vázquez, de UVic-UCC i la UPC respectivament (que farien la part de client-expert), la doctora Marta

Fort Masdevall del departament d'IMAE de la UDG (la meva tutora) i jo mateix. Durant el transcurs del projecte es farien tutories, setmanals o quinzenals, per parlar sobre el disseny, dubtes o idees noves que vagin sorgint. Les reunions amb la meva tutora estaven pensades fer-se presencialment com és habitual, però degut a complicacions externes (COVID-19), ens vam tenir que adaptar a la situació per fer aquestes tutories en videoconferència, mitjançant Skype. Un cop donat el vist bo, es farien reunions amb en Pere Pau i en Jose perquè validessin aquestes noves propostes.

La metodologia que s'ha utilitzat per fer la documentació és totalment diferent de la idea de Scrum, ja que hem utilitzat una aplicació web (Overleaf), que comentarem més endavant, on per la meua part podia anar fent la documentació, i a la vegada la meua tutora podia fer-me comentaris respecte el que anava redactant. La veritat és que aquest paral·lelisme fa que la documentació tingui molt bon ritme.

## 4 Planificació

A continuació es veurà una visió general de la planificació que s'ha seguit per desenvolupar el projecte. Les tasques següents són una visió general dels objectius que es volien aconseguir durant tot els transcurso del treball.

1. **Estudi previ dels antecedents:** Donat que el projecte ja havia estat iniciat per altres investigadors, era una necessitat posar-se al dia amb les noves idees que s'havien dissenyat i els nous conceptes plantejats.
2. **Seguiment del projecte:** Des de ben al principi hi ha hagut reunions amb la meua tutora, ja sigui per comentar les nocions bàsiques del projecte, com per exemple les llibreries, el disseny web i on més endavant es parlaria sobre temes més concrets, com algorismes o funcionalitats de la pàgina web.
3. **Reunions amb els clients-experts:** Com s'ha comentat abans, el seguiment del projecte amb la meua tutora seria periòdicament, per comentar diferents aspectes i dubtes. Però a mesura que el projecte vagi agafant forma es necessari fer reunions amb els clients-experts (en el nostre cas en Pere-Pau i en Jose) que ens ajudarien acabar de matisar totes les idees.
4. **Disseny de la solució:** Vam començar a plantejar-nos com representar les dades que volíem representar de manera que visualment es poguessin interpretar correctament.
5. **Desenvolupament de l'aplicació web:** Si els passos tenen bons resultats, ja es pot donar inici a la codificació de la web, començant per la lectura de fitxers i finalitzant amb les funcionalitats interactives amb l'usuari. Aquest procés tindria una duració que gairebé arribaria fins al final del projecte, ja que posteriorment faltaria la documentació.
6. **Aprenentatge de les llibreries:** Els requisits que es plantejaven en l'aplicació web comportava haver de fer un estudi sobre les llibreries que s'anessin necessitant, ja que eren conceptes nous que encara no dominava. La duració d'aquesta tasca arribaria fins al final del projecte, encara que seria al principi on la corba d'aprenentatge seria més lenta, ja que s'havien d'aprendre conceptes totalment nous.
7. **Primers dissenys de la web:** Un cop s'han après els conceptes generals de les llibreries, ha arribat l'hora de començar a fer un petit esbós de l'idea general de com es vol estructurar la pàgina web, per més endavant mostrar el disseny al client i rebre l'aprovació. Aquest estudi comporta tenir un mínim de nocions bàsiques del disseny web d'una pàgina.



8. **Implementació d'algorismes:** Aquest projecte en concret, ha tingut un cost de temps en el que s'ha estat dissenyant diferents algorismes, ja sigui per obtenir o per mostrar dades en concret dintre de la pàgina.
9. **Documentació:** Si tots els passos anteriors es finalitzen satisfactòriament, es planteja fer la documentació a un mes vista del lliurament de l'entrega del projecte. Ja que per recomanació de la meua tutora, és un bon marge de temps per preparar la memòria. En aquest document s'espera explicar tot el treball realitzat durant tot aquest procés.

A continuació es mostrarà una gràfica on s'observa aproximadament les dates pensades per realitzar tota la planificació esmentada. Fa falta dir que la implementació dels algorismes es tenia pensat assignar-li una franja de temps més curta de la que al final s'ha necessitat.

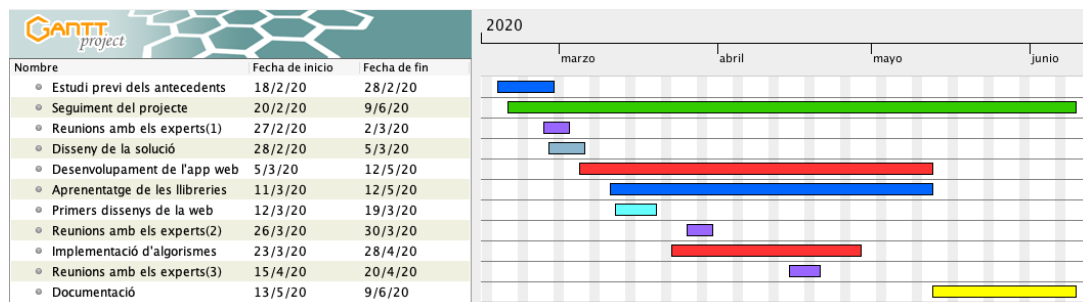


Figura 2: Diagrama de Grantt de la planificació del projecte

## 5 Marc de treball i conceptes previs

Aquest projecte parteix d'una idea inicial en la que es vol fer un anàlisi del participants d'una cursa ciclista. Si l'objectiu final és poder detectar patrons de moviments, concretament de la ciclista del Tour de France 2018, un dels passos previs és tenir una eina per visualitzar totes aquestes dades d'una manera adequada, precisa i sobretot "agradable" a la vista. Les dades de totes les etapes d'una cursa són considerablement grans perquè es puguin analitzar a mà. És aquí on usar una pàgina web ajuda a solucionar el problema. Ara bé si les dades no fossin d'una cursa ciclista sinó d'un altre tipus de cursa i es guardessin seguint l'estructura que usem, aquest projecte no només seria útil per analitzar curses de ciclistes, sinó per analitzar altres tipus de curses.

Per posar al lector una mica en situació es comentaran alguns conceptes previs que s'han de saber per arribar entendre el que es comentarà en els següents apartats. Concretament, parlarem de les persones amb qui s'ha desenvolupat el treball, dels conceptes relacionats amb curses ciclistes, amb la definició i detecció de grups, però també amb les tecnologies usades per elaborar la web on es visualitzar les dades del Tour de France 2018.

### 5.1 Marc de treball

Aquest projecte com ja s'ha dit amb anterioritat venen amb uns antecedents, és per això que com és evident en aquest treball han estat involucrades més persones: Jose Diaz Iriberry, de UVic-UCC, corredor habitual de curses i expert en visualització de la informació, té molt clar com a corredor què li agradaria veure a la web, com estan acostumats a veure la informació i on se'n pot trobar. En Pere Pau Vázquez de la UPC que treballa habitualment en temes de visualització de la informació. I la Marta Fort Masdevall del departament d'IMAE de la UDG que té experiència en la detecció de grups i de patrons de moviment.

### 5.2 Conceptes previs

En aquest apartat es descriuran diferents aspectes relacionats amb la temàtica del treball, com per exemple: la mesura de temps en una cursa ciclista, les diferents classificacions que hi ha i també es mencionaran estudis que s'han fet relacionat amb la visualització de dades de curses. A més a més per tal de dur a terme aquest projecte també s'han tingut que utilitzar diferents eines per facilitar la implementació d'aquest projecte, aquests estudis i decisions també s'explicaran en aquest apartat.

#### 5.2.1 Curses de ciclistes

En aquest apartat s'explicaran els conceptes bàsics per entendre com funciona una cursa ciclista, concretament la del Tour de France. Primer de tot s'ha de mencionar que la cursa està dividida en diferents

etapes, i la cursa de cada una d'aquestes etapes es realitzen en dies diferents. Per cada etapa hi han diferents punts de control, localitzats en diferents trams de l'etapa, d'on s'han obtingut diferents dades per cada un dels ciclistes. Entre aquestes dades hi ha el temps transcorregut i la posició, conceptes que expliquem a continuació.

### Temps transcorregut i posició

Definirem com *elapsed time*, *ET*, d'un ciclista en un punt de control (o punt quilomètric) com al temps transcorregut en segons, entre aquest ciclista i el primer en arribar al punt quilomètric. L'*ET* del primer ciclista en arribar un punt quilomètric és 0. A continuació s'adjunta una petita taula on s'explica millor aquesta definició. Suposant que hi han 3 participants a la cursa, que anomenarem A, B i C. Durant el transcurs de la cursa cada un d'aquests participants arribarà en un moment o altre a un punt quilomètric, i el primer que hi arribi tindrà un *ET* de 0 segons, ja que es el primer. En aquest exemple, en el Km1 es pot veure que el primer en arribar ha sigut el ciclista A (s'assigna *ET* = 0 segons), a continuació el ciclista B i per últim el C. El *ET* de cada un d'aquest darrers ciclistes seran els segons que han trigat en arribar en aquest punt de control en comparació amb ciclista A. Aquesta mateixa idea s'aplica als Km2 i Km3.

L'altre concepte a explicar és la posició, la posició ens ajuda a mantenir un ordre d'arribada per cada un dels ciclistes que van arribant en els punts de control. Aquesta informació és la que més endavant en servirà per representar millor l'evolució dels ciclistes. És en el moment d'arribar en els punts de control quan s'obté la informació de la posició juntament amb l'*ET*.

Ciclista	Km1		Km2		Km3	
	Posició	ET	Posició	ET	Posició	ET
A	1	0s	2	7s	3	4s
B	2	10s	1	0s	2	2s
C	3	20s	3	8s	1	0s

Figura 3: Exemple del transcurs del temps respecte el primer (1)

Un cop s'ha entès aquesta idea, podem exemplificar un cas que s'aproximi més a la realitat. En les curses de llarga distància hi participen molta gent, això fa que quan s'arriba a un punt de control l'aglomeració de participants en aquest punt pugui ser molt gran, és per això que de manera justa, s'assigna el mateix *ET* a tots els membres d'un grup que hagin passat aproximadament en el mateix instant de temps. En la taula següent veurem un petit exemple.

Ciclista	Km1		Km2		Km3	
	Posició	ET	Posició	ET	Posició	ET
A	1	0s	2	7s	3	4s
B	2	10s	1	0s	2	2s
C	8	20s	12	8s	1	0s
...	...	...	...	...	...	...
Z	4	10s	4	7s	5	4s
AA	6	10s	34	67s	4	4s
AB	10	23s	22	12s	7	8s
...	...	...	...	...	...	..

Figura 4: Exemple del transcurs del temps respecte el primer (2)

Si ens fixem en els participants B, Z i AA en el Km1 observem que tenen el mateix ET. El mateix passa en altres ciclistes en el Km2 i en el Km3. Per tant, per definició es pot dir que un grup el formaran aquells ciclistes que tinguin el mateix ET.

### Tipus de classificacions

En el Tour de França hi ha diferents tipus de classificacions, es pot dir llavors que hi ha més d'un guanyador. Per representar cada una d'aquestes classificacions s'utilitzen diferents tipus de mallots, on cada un d'aquests té la finalitat de diferenciar quin tipus de lider hi han en alguna de les etapes de la cursa. Aquest apartat consistirà en explicar els diferents mallots que hi ha.

- **Mallot groc:** Aquest mallot és el més reconegut de tots ja que representa el líder global provisional dintre de la classificació general. La cursa es divideix en moltes etapes, i en cada una d'aquestes etapes s'actualitzen els temps totals de cada un dels ciclistes. D'aquesta manera qui tingui el temps menor, serà el que provisionalment guanyaria la cursa, dintre de la classificació general
- **Mallot verd:** Aquest mallot representa el líder per punts. Aquests punts s'aconsegueixen durant el transcurs d'una etapa, en l'etapa hi han diferents sprints en els quals els participants poden aconseguir punts. Normalment els ciclistes que estan competint per aquest mallot, no competeixen per aconseguir el mallot groc.
- **Mallot de lunars:** Aquest mallot es conegut per tenir lunars de color vermell en una mallot de color blanc. De la mateixa manera que el mallot de color verd, aquest mallot de lunars també s'aconsegueix gràcies a partir de punts, però aquests punts s'obtenen en la cima de les muntanyes. Per tant, qui tingui el mallot de color verd, serà el que en l'etapa anterior aconseguí arribar primer a les cima.
- **Mallot blanc:** Aquest mallot representa aquells ciclistes menors de 25 anys dintre d'un altra tipus de classificació, aquesta classificació no va per punts, sinó que va per temps. Els ciclistes que solen

portar aquest mallot, solen ser futurs candidats a portar el mallot groc en futures carreres.

- **Mallot per equip:** Hi ha un altra classificació que s'anomena classificació per equips, l'equip líder es reconeix per tenir el millor temps respecte dels altres equips. Aquest sumatori de temps es calcula sumant el temps dels tres primers ciclistes de cada equip en la classificació general.

### 5.2.2 Visualització de curses

La temàtica que engloba aquest projecte, visualitzar l'evolució de ciclistes, ja ha sigut estudiat en el passat per diferents papers com per exemple [6], en que es fa un petit estudi sobre la representació dels ciclistes, o per exemple aquest altre paper [7] on es creen diferents visualitzacions per representar el comportament d'un cert grup de ciclistes (concretament el de les bicicletes llogades en Londres). Amb la idea d'aquests papers en ment es crear la definició de trajectòria. Una trajectòria serà la representació d'un ciclista al llarg de l'etapa, on cada una d'aquestes trajectòries servirà per comparar-se amb altres trajectòries que apareixen en l'etapa. Aquesta trajectòria serà dissenyada a partir d'una línia poligonal, on cada un dels segments que la formen serà la distància que hi haurà entre diferents punts quilomètrics representats en la gràfica.

### 5.2.3 Definició de grup

Un grup, d'acord amb el que es va definir a [1] és un conjunt de participants, en aquest cas ciclistes, que passen per un mateix punt quilomètric amb poca diferència de temps entre ells. Aquesta és la definició formal que es va donar aquest article, però com veurem a continuació coincideix amb el que intuïtivament es considera un grup en el món del ciclisme.

En algunes curses poden participar-hi equips, on tots els membres dels equips estiguin competint a la vegada, com es el cas del Tour de France. En aquest apartat es vol deixar clar que la definició d'equip i de grup són conceptes totalment diferents. En un grup, els membres integrants poden ser de diferents equips a la vegada o, per casualitat, d'un sol mateix equip. Un grup sempre constarà de com a mínim dos participants. La creació d'aquests grups es formaran a l'arribar als punts de control (punts quilomètrics o línies verticals) que hi han durant tota la cursa. És en aquests punts on s'obtenen les dades de cada un dels ciclistes. El número de grups de ciclistes que es generen en aquests punts de control no està condicionat per la quantitat de ciclistes que hi passen, sinó per la diferència de temps que hi ha entre ells, la formació d'aquests grups es realitzarà quan aquest temps entre ciclistes sigui suficientment gran. Un grup seran tots aquells ciclistes, independentment de l'equip, que hagin arribat al mateix temps, temps donat per *elapsed time*.

Tornant al paper [1] esmentat abans, també es van formalitzar diferents tipus de grups que es poden formar, aquests són: *survives*, *appears*, *disappears*, *expands*, *shrinks*, *merges*, *splits*, *coheres* i *disbands*. D'aquestes definicions en comentarem breument 3, ja que seran les mencionades en l'apartat 10.3.2 dels resultats.

- **Survives:** És el tipus de grup que d'un punt de control al següent, es manté amb un nombre suficientment gran d'atletes.. En el nostre cas seran els grups de tipus que continuen. poden haver perdut ciclistes o tenir-ne de nous.
- **Merges:** Quan dos o més grups d'un punt de control al següent s'ajunten en un de sol. Per nosaltres aquest tipus de grup seran els que es fusionen.
- **Splits:** Donat un grup en un punt de control, es dividirà en dos o més grups en el punt de control següent. Nosaltres hem definit aquest com els que divideixen.

Aquestes definicions de grups, juntament amb la resta estan exemplificades en la figura 5 extretes de [1].

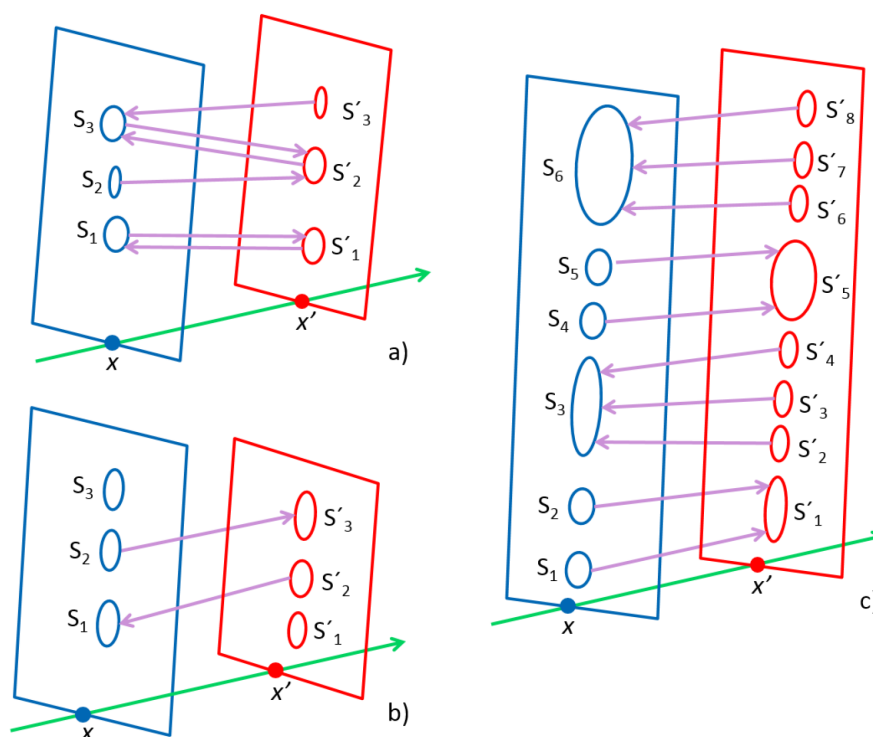


Figure 1: Examples of all possible group evolution patterns a) group  $S_1$  survives as group  $S'_1$ ; group  $S_3$  survives as group  $S'_2$  and, moreover, group  $S'_2$  absorbs group  $S_2$  and group  $S_3$  spawns group  $S'_3$ ; b) group  $S'_1$  appears and group  $S_3$  disappears; group  $S_2$  expands to group  $S'_3$  and group  $S_1$  shrinks into group  $S'_1$ . c) groups  $S_1$  and  $S_2$  merge into group  $S'_1$  and group  $S_3$  splits into groups  $S'_2$ ,  $S'_3$  and  $S'_4$ ; groups  $S_4$  and  $S_5$  cohere into group  $S'_5$  and group  $S_6$  disbands into groups  $S'_6$ ,  $S'_7$  and  $S'_8$ .

Figura 5: Exemple de l'evolució del grups (extreta de [1])

#### 5.2.4 Disseny d'una pàgina web

Alhora de crear una pàgina web és important tenir uns petits coneixements sobre el disseny que tindrà. És important tenir en compte quines són les reaccions dels usuaris al entrar en la pàgina. Això vol dir donar importància en el disseny, ja que abans de fixar-se en el detalls, els usuaris donen més importància a l'estructura i l'estil de la pàgina. Un altra cosa a tenir en compte és la interacció que es facilita a l'usuari, a un usuari se li ha de donar la màxima facilitat per interactuar amb la pàgina, com poden ser per exemple en el moment de seleccionar algun element o alhora de buscar per teclat alguna informació en concret. És important tenir en compte tots aquests petits detalls per més endavant fer un bon disseny.

#### 5.2.5 Servidor local vs Servidor en el Cloud vs Servidor Remote

Un servidor web pel que fa al hardware és un dispositiu que emmagatzema els fitxers que componen tot el lloc web, aquests fitxer ja són els documents HTML, imatges, fitxers CSS (que defineixen l'estil de la pàgina) i fitxers JavaScript, aquests fitxer estan connectats a internet i són accessibles a partir d'un nom de domini.

A més a més d'aquests fitxers també hi ha una base de dades, per més petita que sigui, amb la que es generarà totes les dades necessàries. És el servidor el que ens assegura que les dades estan ben protegides i tenen un constant manteniment. En l'actualitat hi ha tres tipus de servidors [9]:

- **Servidor local:** Ens referim a local quan el mateix servidor es troba en el mateix dispositiu que estem utilitzant.
  - Avantatges: gran velocitat alhora de consultar o modificar les dades, control complet i un increment de la seguretat.
  - Desavantatges: pot comportar un hardware més costós i un manteniment constant.
- **Servidor remot:** Quan es parla de remot significa que aquestes dades estan en un altre dispositiu, aquest és el cas de la majoria d'aplicacions web que podem conèixer. S'accedeix aquestes dades mitjançant la xarxa.
- **Servidor Cloud:** Un servidor Cloud és un tipus de servidor remot (normalment es troben en Data Centers), on s'accedeix mitjançant internet. Les empreses que s'encarreguen de gestionar aquest servei lloguen als usuaris un espai virtual on hi poden guardar diferents tipus de dades. En serien un exemple iCloud o Dropbox, entre molts altres.
  - Avantatges: Manteniment constant i actualitzacions, un emmagatzematge escalable, accedir a les dades en remot, accedir des de qualsevol dispositiu que tingui internet.

- Desavantatges: No s'hi pot accedir sense internet, la velocitat de la transferència de dades dependrà de la qualitat d'internet que es tingui.

En aquest projecte la quantitat de dades que s'havien de manejar no eren relativament grans, però si de cares al futur l'aplicació s'anés fent més complexa, sí que hi hauria una necessitat de tenir un servidor, per més petit que fos, on totes les dades estiguessin ben guardades. Donades les característiques dels diferents tipus de servidors em vaig decantar per utilitzar un servidor local.

Com que es tracta d'una aplicació relativament petita, em podia fer totalment responsable de el manteniment d'aquestes dades. Això implicava tenir que fer còpies de seguretat si fos necessari. I suposant que les dades no s'haguessin guardat correctament, fer les petites correccions que es convinguessin.

### 5.2.6 W3Schools

W3Schools és una web per desenvolupadors, on es poden trobar una gran quantitat de tutorials i referències de molts llenguatges de programació. Entre aquests llenguatges hi ha HTML, CSS, JavaScript i jQuery, aquests han sigut els llenguatges utilitzats en el projecte, juntament amb D3, que comentarem més endavant. Aquesta pàgina web, encara que no té una afiliació amb l'estàndard W3C (World Wide Web Consortium) té una gran influència en l'àmbit informàtic, ja que aporta una gran ajuda en aquells que comencen aprendre o necessiten consultar conceptes bàsics d'algun llenguatge. W3Schools es va crear al 1998 per una empresa noruega anomenada Refsnes Data [10].

### 5.2.7 D3.js

D3.js [11] és una llibreria de JavaScript que està dissenyada especialment per manipular dades a partir d'un fitxer. Un cop s'han llegit aquestes dades, D3 utilitza tècniques com HTML, SVG i CSS per crear nous objectes en la pàgina web i mostrar-los a l'usuari. Aquesta llibreria soluciona el problema de la manipulació eficient d'una pàgina web basada en dades. Com que pot suportar una gran quantitat d'informació, el comportament alhora de modificar i interactuar amb la pàgina, no té cap efecte significatiu a la velocitat de resposta. A continuació es comentaran certs aspectes importants sobre aquesta llibreria.

- **Selections:** Modificar el DOM d'una pàgina web utilitzant l'API de W3C pot arribar a ser molt tediós ja que el seu llenguatge imperatiu implica tenir que fer moltes iteracions. D3.js aporta una molt bona solució aquest problema.



```
var paragraphs = document.getElementsByTagName("p");
for (var i = 0; i < paragraphs.length; i++) {
  var paragraph = paragraphs.item(i);
  paragraph.style.setProperty("color", "blue", null);
}
```

Figura 6: Exemple canviar color dels paràgraf amb W3C DOM API

```
d3.selectAll("p").style("color", "blue");
```

Figura 7: Exemple canviar color dels paràgraf amb D3.js

- **Dynamic Properties:** Amb D3 l'estil, els atributs i altres propietats del elements del DOM poden ser definides a partir de funcions. Això fa que el llenguatge sembli molt simple però a la vegada molt potent. A més d'això D3 conté moltes funcions predefinides amb les que pot projectar les dades elements que necessitem com per exemple SVG. En aquest projecte s'utilitzen diferents funcions predefinides que comentarem més endavant.

```
d3.selectAll("p").style("color", function() {
  return "hsl(" + Math.random() * 360 + ",100%,50%";
});
```

Figura 8: Exemple canviar color dels paràgraf aleatòriament amb D3.js

- **Modify nodes:** A partir d'unes dades, la llibreria D3.js es capaç de crear nous objectes (o nodes), modificar-los i si fos necessari eliminar aquells que ja no siguin d'utilitat. Aquesta facilitat en el llenguatge, fa que es pugui precisar quina es la funció de cada un dels elements dintre de la pàgina. Això ens aporta una gran control en el que es refereix a tots element creats a partir d'unes dades.
- **Transformation, not Representation:** D3 no crea una nova representació visual dels objectes, sino que utilitza les eines estàndards que ja coneixem com són HTML, SVG i CSS. Aquesta característica fa que l'aprenentatge d'aquesta llibreria sigui molt més fàcil. Això fa que debuggar mitjançant l'inspector de navegador sigui idèntic al que hem fet fins al moment, ja que els nodes que es creen són els estàndards.
- **Transitions [12]:** El CSS és un llenguatge que descriu quin estil ha de tenir un document HTML. Entre totes les eines que hi ha les transicions permeten canviar les propietats dels atributs d'una manera senzilla i sempre dintre d'un marge de temps. D3 utilitza aquesta idea de transició per modificar les propietats i l'estil de la pàgina per donar animació a la pàgina. Per exemple pot canviar el color del fons d'una pàgina. O pot modificar el radi dels cercles que apareixen en una pàgina web. Veiem un parell d'exemples:

For example, to fade the background of the page to black:

```
d3.select("body").transition()
  .style("background-color", "black");
```

Or, to resize circles in a symbol map with a staggered delay:

```
d3.selectAll("circle").transition()
  .duration(750)
  .delay(function(d, i) { return i * 10; })
  .attr("r", function(d) { return Math.sqrt(d * scale); });
```

Figura 9: Exemple canviar color del fons i modificar el radi dels cercles

En el primer exemple es pot comprovar com d'una manera senzilla es modifica el color del fons a negre. Si analitzem el segon exemple veiem que hi aplica moltes propietats:

1. Selecciona tots els elements cercles que apareixen en la pàgina.
2. Aquesta transició tindrà una duració de 750 mili-segons
3. Per cada un dels elements del cercle que ha seleccionat s'esperarà  $i * 10$  mili-segons a fer la transició, on "d" representa la informació de l'element seleccionat i "i" és la posició en la que es troba aquest element dintre de la selecció dels cercles.
4. La propietat que modifica en cada un dels elements és el radi.

Modificant únicament els atributs, D3 redueix el cost que pot comportar aplicar totes aquestes accions, fent que la interpolació entre estils sigui molt més fluida. Aquestes transicions també poden veure afectades per *events* que configurem a la pàgina, per exemple *onclick*, *hover* o com veurem més endavant *events* més complexes com per exemple el zoom.

Per acabar vull fer menció als exemples de projectes D3 que es poden trobar a internet. En el procés d'aprenentatge una part molt important, en la meua opinió, es la visualització d'exemples. És per això que un cop llegida una mica la documentació sobre D3, estava bastant interessat en buscar projectes d'altres persones per veure l'estructura del codi i quins resultats oferien. Va ser quan vaig trobar la pàgina *The D3.js Graph Gallery* [15]. És una pàgina web on es pot trobar una gran col·lecció de gràfiques fetes amb D3.js. Tinc que donar els meus agraïments en Yan Holtz, expert en anàlisi en visualització de les dades, creador d'aquesta pàgina que he comentat.

### 5.2.8 Paleta de colors

En el disseny d'una pàgina web és molt important que tots elements que hi participen tinguin coherència. És a dir, la posició de cada un dels objectes a la pàgina, la grandària que han de tenir, els diferents colors que hi poden aparèixer, entre altres molts factors. És en la paleta de colors on vull agrair en Pere-Pau

per donar-me a conèixer el projecte Color Brewer [13]. És una eina en la et donen una orientació sobre l'elecció dels colors que més conjunyeixen entre ells, a partir de l'estudi de la Dra. Cynthia Brewer.

A continuació s'adjunta una imatge de com és la pàgina web. Hi han dos paràmetres que vull comentar, el primer és el que està a dalt a l'esquerra, el nombre de classes que t'interessen representar, venen a ser les diferents tonalitats de verds que hi han en el mapa, això simularia els diferents elements que tindriem a la pàgina. L'altre paràmetre són els *roads* en l'apartat *Context*, aquest element són les línies de color vermell que apareixen en el mapa. Aquest paràmetre ha sigut molt important ja que això simularia la visualització d'un ciclista un cop s'ha seleccionat, com veurem en exemples més endavant

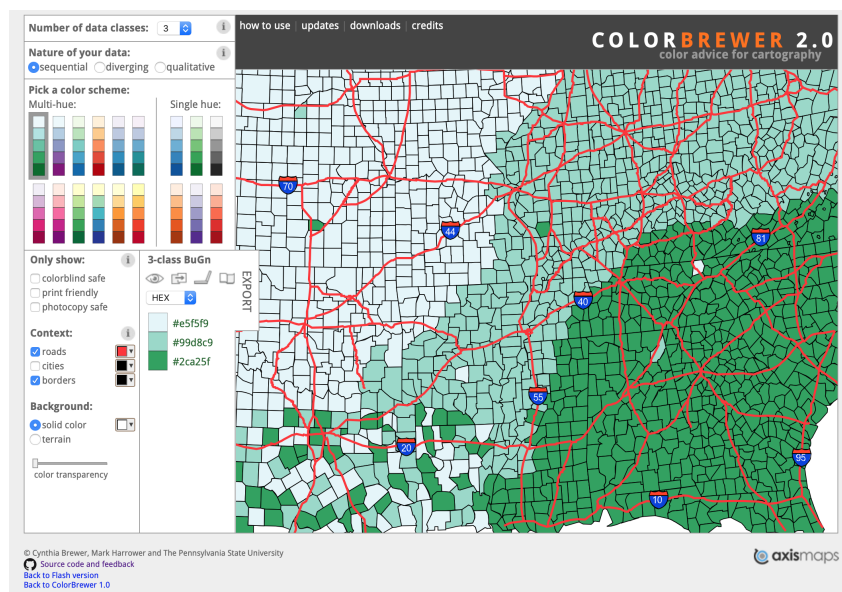


Figura 10: Pàgina web de Color Brewer

## 6 Requisits dels sistema

L'aplicació web ha de complir una sèrie de requisits funcionals i no funcionals perquè sigui accessible per els usuaris.

### 6.1 Requisits funcionals

- L'usuari ha de poder escollir un ciclista o un equip amb tots els membres d'alguna etapa que apareix a la cursa.
- L'usuari ha de veure, amb la màxima precisió, quina ha sigut l'evolució del ciclista o equip escollit en els trams/punts quilomètrics que hi ha en l'etapa.
- L'usuari ha de tenir la possibilitat de comparar dos ciclistes d'una mateixa etapa.
- L'usuari ha de poder saber quin corredor pertany a cada una de les trajectòries.
- L'usuari ha de poder veure per cada punt quilomètric el retard de temps que porta un ciclista respecte el primer ciclista que ha arribat en aquell tram.
- L'usuari en tot moment ha de veure quins són els diferents grups que s'han generat durant l'etapa d'una cursa i com han evolucionat.
- L'usuari en qualsevol moment pot fer zoom a la pàgina, la gran quantitat d'informació fa que es necessiti aquesta funcionalitat. Així permet veure amb més claredat la informació que es necessita.
- S'ha de donar informació sobre els ciclistes i els equips que hi participen.
- En tot moment s'ha de veure la classificació dels guanyadors de l'etapa i el líder de la classificació general de la cursa.
- Depenent de l'escala en la que es troba la pàgina, es a dir, quant de zoom s'està aplicant en aquell moment, apareixen unes etiquetes o unes altres informant sobre la diferència de temps entre grups consecutius en diferents punts quilomètrics. Les etiquetes es mostren en funció del zoom per evitar la sobreinformació que hi hauria si en tot moment es mostressin totes.
- L'usuari en tot moment pot escollir quina és l'etapa que vol visualitzar.
- S'ha de veure un perfil de l'etapa per identificar els desnivells durant tota la cursa, en aquest mateix perfil s'identificarà el tram de perfil que s'està visualitzant en el moment de fer zoom in.

## 6.2 Requisites no funcionals

- Disposar un servidor on emmagatzema les dades de les etapes, ciclistes i equips
- Tenir connexió a internet, al utilitzar-se llibreries com D3.js o jQuery de manera online és primordial tenir connexió a internet.
- El temps de resposta de l'aplicació ha de ser ràpid per a l'usuari que l'estigui utilitzant.

## 7 Estudis i decisions

En aquest apartat es descriurà els diferents llenguatges utilitzats, les llibreries importades per fer l'aplicació, i també diferents programes que s'han utilitzat per fer la documentació.

### 7.1 Llenguatges

#### 7.1.1 JavaScript

La generació dels elements a mostrar i la interacció amb l'usuari ha sigut gràcies aquest llenguatge de programació. JavaScript està pensat per utilitzar-se en navegadors web, per crear objectes dinàmics i la possible interacció amb l'usuari. Aquest llenguatge té la capacitat de crear tasques que només afectin al dispositiu del client, és redundat, però això fa que el servidor no hi hagi d'intervenir per res. Per exemple si es donés el cas de que durant l'execució d'un programa es perdés la connexió a internet, la pàgina web funcionaria parcialment, es possible que es pugui modificar l'estructura de la pàgina sense que el servidor intervingui. És un llenguatge asíncron, això vol dir que l'aplicació es pot comunicar amb el servidor sense que l'usuari es vegi afectat.



Figura 11: Icona de JavaScript

#### 7.1.2 HTML

Totes les pàgines web tenen una estructura preestablerta i aquest projecte no en seria l'excepció. HTML serveix per decidir quin ordre tindran tots els elements que conté la pàgina. Això es fa utilitzant unes marques *hypertext* que ajudarà a relacionar els elements de la pàgina, són unes etiquetes conegudes en anglès com a *tags*



Figura 12: Icona de HTML

### 7.1.3 CSS vs Bootstrap

En aquest projecte en concret la visualització de les dades és molt important, és per això que escollir un bon estil afectaria al resultat final. Les alternatives per treballar l'estil de la pàgina venen són CSS o Bootstrap.

- **CSS (Cascading Style Sheets):** és un llenguatge de programació que s'utilitza per descriure l'estil d'una pàgina web a partir de documents HTML. Des de 1996 que s'està utilitzant aquesta tècnica, i per ara ja van per la versió CSS3. Aquesta tècnica té l'avantatge de poder crear un estil propi, per contra, tota la feina d'estructurar l'estil (la mida dels elements, els colors, tipografies, formularis, botons, taules...) l'ha d'implementar el desenvolupador de la web.
- **Bootstrap:** Aquesta llibreria multiplataforma és relativament més nova ja que es va donar a conèixer al 2011. La seva funcionalitat bàsica és la d'oferir plantilles ja preestablertes. Fent que l'estructura de l'estil de la pàgina sigui menys costosa d'implementar.

Amb aquestes dues possibilitats a escollir, vaig decidir utilitzar CSS ja que era amb el que tenia més experiència i considerava que amb més flexibilitat ajudaria a visualitzar millor les dades. Un cop finalitzat el projecte m'agradaria esmentar que si pogués tornar a començar de nou segurament escolliria les dues eines, ja que durant el projecte m'he donat compte que he estat molt de temps aplicant un estil que ja fàcilment podia replicar amb Bootstrap. Segurament agafar el millor de les dues tècniques hagués sigut la millor opció.



Figura 13: Icona de CSS



Figura 14: Icona de Bootstrap

## 7.2 Llibreries

### 7.2.1 jQuery

És una llibreria multi-plataforma de JavaScript iniciada al 2006, que permet modificar els elements d'una pàgina web d'una forma senzilla. La funció principal que ha tingut jQuery ha sigut la de acabar d'implementar totes les funcionalitats d'interacció amb l'usuari.



Figura 15: Icona de jQuery

### 7.2.2 D3 vs Qt

El projecte ha de permetre un anàlisi còmode de les dades i per fer això s'havia de buscar un software que resolgués aquest propòsit. Les dues alternatives que tenia eren:

- D3: És una llibreria de JavaScript pensada per crear una representació visual a partir d'unes dades. Aquestes dades es poden manipular segons les necessitats de cada objectiu. S'utilitzen els conceptes bàsics de HTML, CSS i SVG.
- Qt: Per altra banda és un framework multiplataforma orientada a a objectes, utilitza una interfície gràfica encara que també pot ser utilitzar per aplicacions CLI (Command Line Applications). El llenguatge natiu es C++ encara que es poden incorporar nous llenguatges fent adaptacions de llibreries (binding)

Amb aquestes dues visions, al final D3 va ser l'opció guanyadora ja que esta ambientat en pàgines web i juntament amb JavaScript tenia l'oportunitat d'aprendre nous conceptes. En canvi amb Qt ho més probable era que programés quasi tota l'estona amb C++, no tinc res en contra d'aquest llenguatge, però ja l'hem tingut que aprendre bastant durant tota la carrera i la veritat estava més interessat en ampliar una mica més els meus coneixements.

En el moment de començar a utilitzar D3 vaig utilitzar idees d'altres exemples per tenir una lleugera idea de com començar a "picarçodi, la majoria d'exemples que es poden trobar per internet estan implementades amb la versió 4, es per això que la llibreria implementada en aquest projecte és d'aquesta versió. Però la més recent és la número 5, on la novetat més important és la compatibilitat d'utilitzar les *Promises* [16].



Figura 16: Icona de D3



Figura 17: Icona de Qt

## 7.3 Programari utilitzat

Per fer aquest projecte he utilitzat el meu portàtil personal, durant tota la carrera l'he estat utilitzant per treballs de la universitat i la veritat sempre m'ha anat molt bé. Té un sistema operatiu macOS i es totalment compatible amb el projecte. Com hem tractat una aplicació web, el sistema operatiu no té cap inconvenient.



### 7.3.1 Sublime Text

És essencial utilitzar un editor de text amb el que poder programar el projecte, la meva elecció ha sigut Sublime Text, ja que durant tota la carrera he anat provant diferents editors de text com Notepad++, TextWrangler, Leafpad i al final la meva elecció i amb la que em sento més còmode és Sublime, tinc la sensació de que va una mica més ràpid que les altres opcions, aparentment sembla un editor bàsic però realment té moltes funcionalitats bastant interessants, com per exemple buscar contingut en tots els fitxers oberts, amb el que es coneix *Goto Anything*, amb aquesta funció no només pots buscar text, sinó també un fitxer, una línia en concret o un mètode creat. El codi font de Sublime Text està escrit amb C++ i Python. Aquest editor de text es pot descarregar i avaluar de forma gratuïta. De totes maneres no es de software lliure o de codi obert i s'ha d'obtenir una llicència pel seu us continu, encara que la versió bàsica és plenament funcional i no té data de caducitat.



Figura 18: Icona de Sublime

### 7.3.2 GitHub

Com que el desenvolupador de l'aplicació era jo, el responsable del codi i del seu manteniment eren cosa meva. Un bon hàbit, alhora de programar codi, és tenir un control de versions sobre tot el codi. Això fa tenir prevenir possibles mal de caps que vinguin en el futur, com per exemple la pèrdua o l'eliminació del codi o voler tornar cap enrere ja que la codificació que que es tenia abans era millor. Una eina molt utilitzada en la informàtica es el GitHub, és un espai web on pots guardar tots els teus projectes en diferents repositoris. Aquests repositoris fer-los públics o privats, i si fos necessari també poden ser compartits. Hi ha altres alternatives que et faciliten aquest control de versions, com per exemple: GitLab o Bitbucket. Però pel que a mi respecta, GitHub es l'eina que domino més i és per això que em vaig decantar per aquesta opció.



Figura 19: Icona de GitHub

### 7.3.3 Wavi

La decisió de programar el projecte amb un editor de text ha comportat haver d'utilitzar un software extern per generar les diferents classes que s'implementen. La veritat és que ha sigut de gran utilitat ja que implementar totes les taules a mà hagués implicat un gran quantitat de temps. És per això que per fer la representació de les classes de la nostra aplicació web s'ha utilitzat un repositori de Github anomenat Wavi (Web Application Viewer) [20]. Aquesta aplicació està dissenyada per generar diagrames de classes amb format SVG, per aquells desenvolupadors que vulguin documentar les seves aplicacions web. Per generar aquests diagrames es pot fer instalant el wavi a l'ordinador o utilitzar la seva pàgina web i fer-ho online.



Figura 20: Icona de Wavi

### 7.3.4 GanttProject

Com s'ha comentat en l'apartat 4, la planificació és molt important per fer un bon projecte. Gantt és un tipus de diagrama on l'objectiu és exposar el temps la dedicació que es té pensat dedicar en les diferents tasques. GanttProject és una eina de codi obert programat en Java amb el que es pot crear aquest tipus de gràfica i veure l'evolució de les diferents tasques programades.



Figura 21: Icona de GanttProject

### 7.3.5 Draw.io

És un recurs online gratuït que permet generar tot tipus de diagrama. El projectes creats es poden guardar al núvol (Google Drive o One Drive) o es pot guardar el mateix projecte en el dispositiu que es convingui. Els diagrames que veuran en el transcurs de la memòria s'han fet amb aquesta eina.



Figura 22: Icona de Draw.io

### 7.3.6 Skype

El procés més habitual que es dur a terme per fer el seguiment del PFG és fer reunions presencials amb el tutor, cada cert temps establert. A menys que es decideixi fer el projecte en època d'estiu. Però degut al problemes sorgits amb el COVID-19 vam haver d'adaptar les tutories mitjançant videoconferència, l'opció més generalitzada i més còmode tan per mi com per la meva tutora va ser utilitzar Skype.



Figura 23: Icona de Skype

### 7.3.7 Overleaf

Per fer la documentació el recurs que he utilitzat és Overleaf, un startup fundada al 2012, on el seu propòsit és documentar online amb format LaTeX. Aquesta eina l'he començat a utilitzar aquest darrer any per documentar projectes d'altres assignatures i personalment fa que el flux del procés sigui una mica més amè. Aquesta startup està també pensada per projectes col·laboratius, la meva tutora i jo hem utilitzat aquesta avantatge per dur a terme la revisió i redacció de la memòria en paral·lel. Per la meua part jo desenvolupava la documentació i quan donava un apartat per acabat, a la vegada que jo n'escrivia un altra, ella feia petits comentaris i consells de com poder millorar la part de la memòria del que ja donava per acabada. Overleaf proporciona un xat per cada projecte que es realitza, aquesta eina també l'hem utilitzat ja que per dubtes o comentaris petits surt més a compte que tenir que fer un skype tota l'estona.



Figura 24: Icona de l'Overleaf

### 7.3.8 Color Brewer

Per fer un bon disseny de l'estil de colors que tindrà la pàgina, i sobretot de quins colors tindran els elements de la gràfica s'ha utilitzat un repositori de GitHub el qual ens aporta informació de el conjunt de colors que millor es combinen alhora d'implementar dissenys d'una web.



Figura 25: Icona de Color Brewer

## 8 Anàlisi i disseny del sistema

Com s'ha argumentat en apartats anteriors, aquest projecte té ben definides certes funcionalitats que analitzarem a continuació. Per aconseguir aquestes funcionalitats, cal realitzar certes tasques que es poden dividir en dos blocs. Per una banda cal analitzar i dissenyar la interfície per l'usuari. Aquest disseny s'ha discutit amb els membres del projecte (Pere-Pau, Jose, Marta i jo) per tal de determinar com s'han d'estructurar els elements de la pàgina perquè l'usuari pugui interpretar bé les dades. Per altra banda, un cop es té la idea general de l'estructura, s'analitzaran individualment els elements més importants de la pàgina, com són per exemple la representació dels ciclistes o grups generats.

### 8.1 Anàlisi i disseny de la interfície usuari

Abans de començar a codificar és molt important tenir una idea general de com s'havia d'estructurar la pàgina i més endavant saber com representar cada element, és per això que a continuació és mostra la primera idea que vam tenir per estructurar l'aplicació. El grup d'investigació Pere-Pau, Jose i Marta ja en van començar a fer un model fa un temps i la veritat he d'agrair que hagin compartit amb mi els seus coneixements, de cares a estructurar millor cada un dels elements. Es per això que es va concretar un dia i una hora per parlar sobre el disseny i acabar de decidir la visualització de la pàgina. S'adjunta una imatge sobre l'estructura dissenyada de la pàgina web.

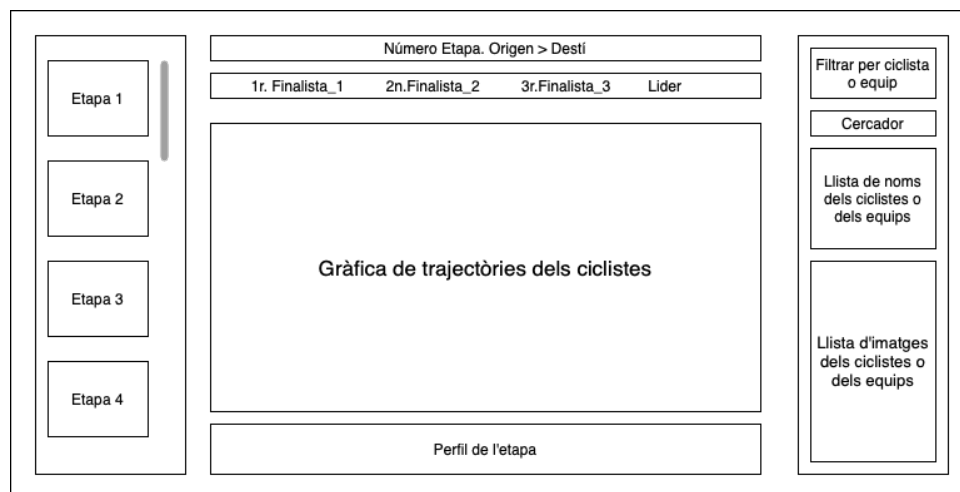


Figura 26: Esbós de la visualització de curses

La part més important de l'aplicació és la visualització de les dades d'una etapa, anomenada *etapa actual*, que seleccionarà l'usuari. És per això que la gràfica ocupa una gran part respecte la resta. A continuació es comentarà breument els elements que es poden veure.

- A l'esquerra es pot veure una llista de totes les etapes que formen la cursa. En aquest exemple hi

hauran un total de 21 etapes, on l'etapa actual quedarà ressaltada.

- A la part superior es donarà informació de 3 dades de l'etapa actual que s'està visualitzant:
  - El número d'etapa actual amb el seu origen i destí
  - La classificació dels 3 primers en acabar l'etapa actual
  - El líder de la classificació general provisional en el moment de disputar l'etapa, en aquest se li dona un mallot de color groc per diferenciar-lo
- A la part inferior s'observa un perfil de l'etapa actual, aquest perfil dona informació sobre el desnivell que hi ha durant el transcurs de la cursa.
- A la part de la dreta, es podrà seleccionar quin tipus d'informació es vol veure, si la dels ciclistes o la dels equips. A més a més en tot moment hi haurà un cercador amb el que es podrà filtrar per nom per trobar amb més facilitat un ciclista o un equip, ja que hi hauran molts participants.
- Per últim, en el centre de la pàgina es veurà un gràfic, on s'hi hauran de representar diferents elements:
  - Totes les trajectòries que representaran cada un dels ciclistes
  - Línies verticals que representaran punts quilomètrics de l'etapa
  - Els polígons que utilitzarem per representar els grups formats en cada punt quilomètric i la seva evolució
  - Etiquetes que mostraran la diferència de temps entre dos grups consecutius d'un mateix punt quilomètric
  - Si es selecciona alguna trajectòria, aquesta es dibuixarà d'un color diferent i es podrà obtenir per aquell ciclista i cada punt quilomètric la següent informació: el retard de temps respecte al primer ciclista en arribar al punt quilomètric, la seva posició provisional a la classificació general en aquell punt quilomètric i el retard provisional respecte el líder de la classificació general en aquell punt quilomètric.

Un cop s'ha definit el disseny de la pàgina web, s'ha de dissenyar com s'executaran els diferents requeriments que s'han comentat en punt 6, s'ha de començar a fer un anàlisi de cada una de les funcionalitats que la pàgina web, d'aquesta manera quan arribi el moment de codificar es tindrà ben definida cada una de les tasques. En l'apartat següent es comentaran aquests requeriments.

### 8.1.1 Anàlisi dels requeriments

Un usuari en qualsevol moment ha de poder seleccionar qualsevol etapa de la cursa, per visualitzar unes altres dades.

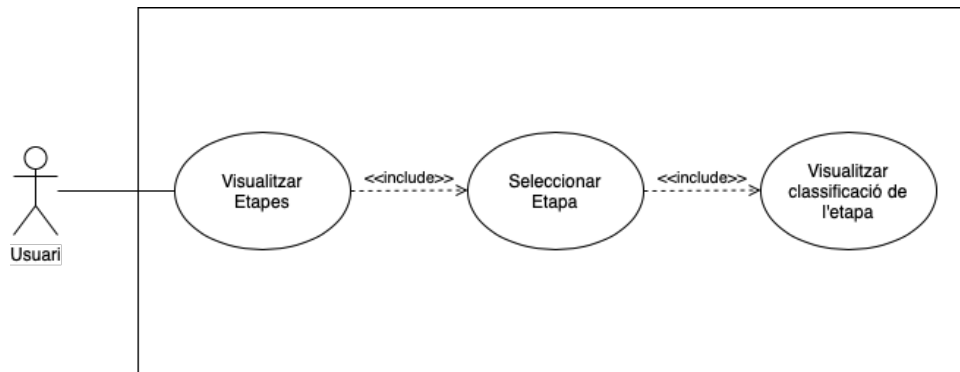


Figura 27: Diagrama cas d'ús de Gestionar etapes

Cas d'ús	Seleccionar Etapa
Versió	Visió anàlisi
Descripció	Un usuari vol seleccionar una etapa diferent de la cursa
Actors	Usuari
Precondició	Tenir les dades de les diferents etapes de la cursa
Flux principal	<ol style="list-style-type: none"> <li>1. <b>Scroll</b> en la barra lateral esquerra de la pàgina</li> <li>2. <b>Clicar</b> en una de les etapes</li> <li>3. <b>Mostrar</b> gràfica de la nova etapa</li> </ol>
Flux alternatiu	Cap
Postcondició	La pàgina mostra la nova etapa amb la seva classificació

Figura 28: Fitxa de cas d'ús de Seleccionar etapa

Un usuari ha de poder buscar entre tots els participants de la cursa a un ciclista.

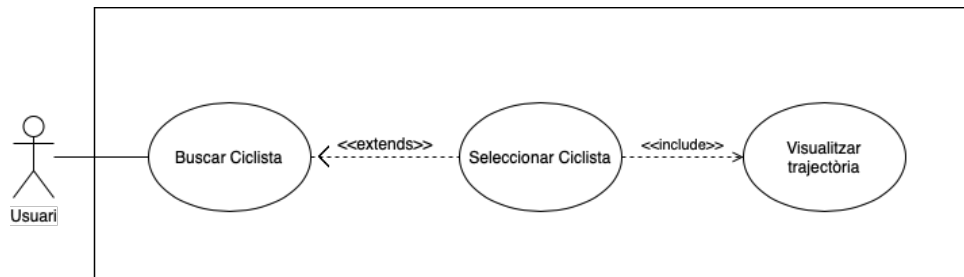


Figura 29: Diagrama cas d'ús de Buscar ciclista

Cas d'ús	Buscar ciclista
Versió	Visió anàlisi
Descripció	Un usuari vol buscar un ciclista en concret
Actors	Usuari
Precondició	Cap
Flux principal	<ol style="list-style-type: none"> <li>1. <b>Filtrar</b> per ciclistes en la barra lateral dreta</li> <li>2. <b>Clicar</b> en el cercador</li> <li>3. <b>Entrar</b> nom del ciclista</li> <li>3.1 <b>Mostrar</b> ciclistes que coincideixen amb els caràcters introduïts.</li> </ol>
Flux alternatiu	Cap
Postcondició	S'han filtrat els ciclistes amb el nom introduït

Figura 30: Fitxa de cas d'ús de Buscar ciclista



Un usuari ha de poder buscar en qualsevol moment algun equip en concret.

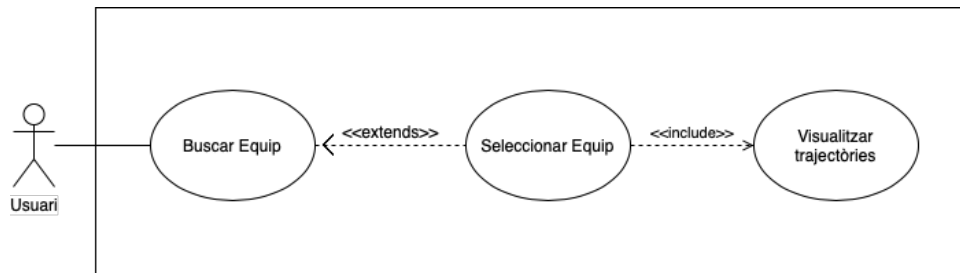


Figura 31: Diagrama cas d'ús de Buscar equip

Cas d'ús	Busca equip
Versió	Visió anàlisi
Descripció	Un usuari vol buscar un equip en concret
Actors	Usuari
Precondició	Cap
Flux principal	<ol style="list-style-type: none"> <li>1. <b>Filtrar</b> per equips en la barra lateral dreta</li> <li>2. <b>Clicar</b> en el cercador</li> <li>3. <b>Entrar</b> nom de l'equip <ol style="list-style-type: none"> <li>3.1 <b>Mostrar</b> equips que coincideixen amb els caràcters introduïts.</li> </ol> </li> </ol>
Flux alternatiu	Cap
Postcondició	S'han filtrat els equips amb el nom introduït

Figura 32: Fitxa de cas d'ús de Buscar equip

Un usuari ha de poder visualitzar en la pàgina web a un ciclista.

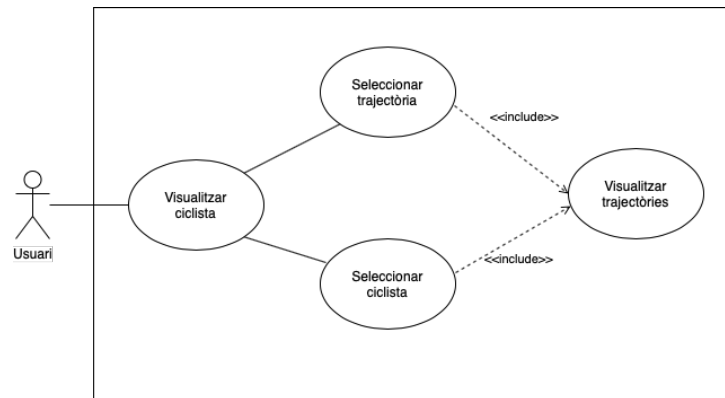


Figura 33: Diagrama cas d'ús de Visualitzar ciclista

Cas d'ús	Seleccionar trajectòria
Versió	Visió anàlisi
Descripció	Un usuari vol seleccionar una de les trajectòries que apareixen
Actors	Usuari
Precondició	Cap
Flux principal	<ol style="list-style-type: none"> <li>1. <b>Mouse over</b> per les trajectòries de la gràfica               <ol style="list-style-type: none"> <li>1.1 <b>Mostrar</b> hover sobre les trajectòries que es fa <b>mouse over</b>.</li> </ol> </li> <li>2. <b>Clicar</b> en alguna de les trajectòries.</li> <li>3. <b>Mostrar</b> ciclista seleccionat</li> </ol>
Flux alternatiu	Cap
Postcondició	Es visualitza la trajectòria en un color diferent i el ciclista al que pertany

Figura 34: Fitxa de cas d'ús de Seleccionar trajectòria

Cas d'ús	Seleccionar ciclista
Versió	Visió anàlisi
Descripció	Un usuari vol seleccionar un ciclista
Actors	Usuari
Precondició	Els ciclista ha d'estar participant en l'etapa actual
Flux principal	<ol style="list-style-type: none"> <li>1. <b>Filtrar</b> per ciclistes en la barra lateral dreta</li> <li>2. Es pot buscar ciclista pel cercador               <ol style="list-style-type: none"> <li>2.1 <b>Mostrar</b> els ciclistes que coincideixen amb el nom entrat</li> </ol> </li> <li>3. <b>Scroll</b> en la llista de ciclistes</li> <li>4. <b>Escollir</b> i fer clic en el ciclista</li> <li>5. <b>Mostrar</b> ciclista seleccionat</li> </ol>
Flux alternatiu	Cap
Postcondició	Es visualitza la trajectòria en un color diferent i el ciclista al que pertany

Figura 35: Fitxa de cas d'ús de Seleccionar un ciclista

Un usuari ha de poder visualitzar en la pàgina web a un equip en concret.

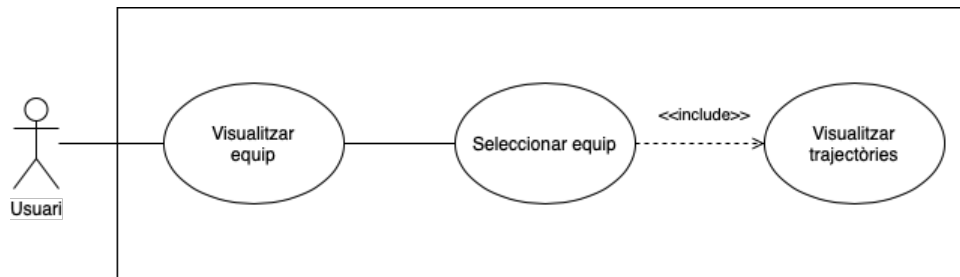


Figura 36: Diagrama cas d'ús de Visualitzar un equip

Cas d'ús	Seleccionar equip
Versió	Visió anàlisi
Descripció	Un usuari vol seleccionar un equip
Actors	Usuari
Precondició	Cap
Flux principal	<ol style="list-style-type: none"> <li>1. <b>Filtrar</b> per equips en la barra lateral dreta</li> <li>2. Es pot buscar equip pel cercador               <ol style="list-style-type: none"> <li>2.1 <b>Mostrar</b> els equips que coincideixen amb el nom entrat</li> </ol> </li> <li>3. <b>Scroll</b> en la llista de ciclistes</li> <li>4. <b>Escollir</b> i fer clic en l'equip</li> <li>5. <b>Mostrar</b> equip seleccionat</li> </ol>
Flux alternatiu	Cap
Postcondició	Es visualitza les trajectòries dels membres de l'equip en un color diferent i la informació sobre l'equip

Figura 37: Fitxa de cas d'ús de Seleccionar equip

Donat un punt quilomètric un usuari ha de poder visualitzar quin és el retard de temps que hi hagut entre un ciclista i el primer ciclista en arribar en aquell punt quilomètric.

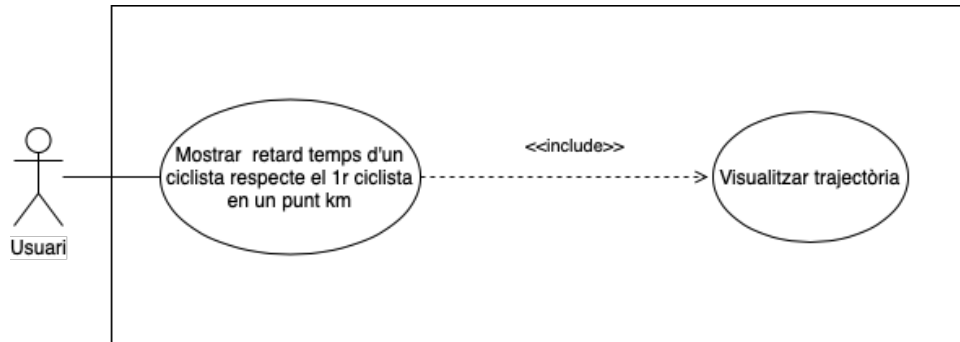


Figura 38: Diagrama cas d'ús de Mostrar elapsed time

Cas d'ús	Mostrar retard d'un ciclista respecte al 1r que arribat en un punt km
Versió	Visió anàlisi
Descripció	Donat un punt quilomètric, un usuari vol veure la diferència de temps que hi ha entre un ciclista i el 1r ciclista que hi ha arribat
Actors	Usuari
Precondició	S'ha de visualitzar alguna trajectòria seleccionada
Flux principal	<ol style="list-style-type: none"> <li>1. <b>Mouse over</b> per sobre d'alguns del punts que apareixen en la trajectòria</li> <li>2. <b>Mostrar</b> informació sobre el ciclista: retard de temps respecte el 1r, en aquell punt quilomètric, posició en la classificació general i retard de temps en la classificació en general</li> </ol>
Flux alternatiu	Cap
Postcondició	Es mostra la informació del retard

Figura 39: Fitxa de cas d'ús de Mostrar retard

Un usuari ha de poder escollir entre dos ciclistes i comparar les dades de cada un d'ells

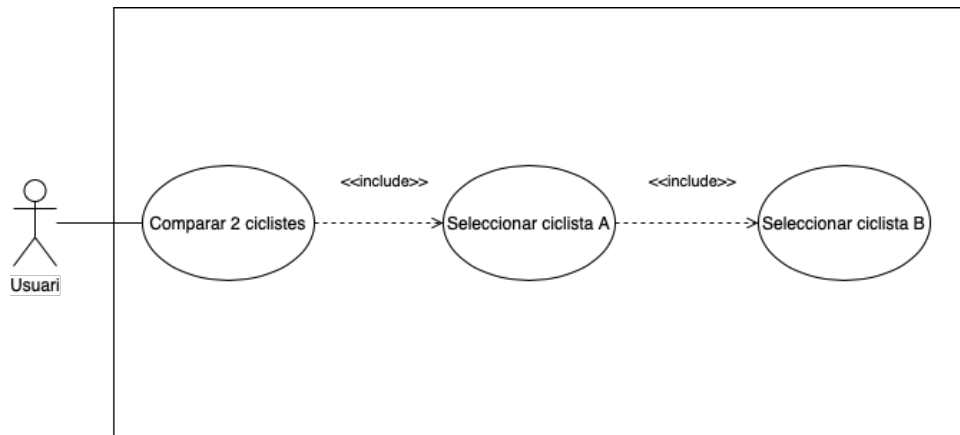


Figura 40: Diagrama cas d'ús de Comparar dos ciclistes

Cas d'ús	Comparar 2 ciclistes
Versió	Visió anàlisi
Descripció	Un usuari vol comparar 2 ciclistes
Actors	Usuari
Precondició	Cap
Flux principal	<ol style="list-style-type: none"> <li>1. <b>Filtrar</b> per ciclistes en la barra lateral dreta</li> <li>2. <b>Mantenir</b> pressionat el botó "<u>cmd</u>" en Mac o "<u>ctrl</u>" en Windows.               <ol style="list-style-type: none"> <li>2.1 <b>Clicar</b> un dels ciclistes de la barra lateral</li> <li>2.2 <b>Clicar</b> un altre ciclista de la barra lateral</li> </ol> </li> <li>3. <b>Mostrar</b> resultat en la pàgina</li> </ol>
Flux alternatiu	<ol style="list-style-type: none"> <li>1. <b>Filtrar</b> per ciclistes en la barra lateral dreta</li> <li>2. <b>Mantenir</b> pressionat el botó "alt"               <ol style="list-style-type: none"> <li>2.1 <b>Clicar</b> una de les trajectòries de la gràfica</li> <li>2.2 <b>Clicar</b> una de les altres trajectòries de la gràfica</li> </ol> </li> <li>3. <b>Mostrar</b> resultat en la pàgina</li> </ol>
Postcondició	Es visualitzen dues trajectòries seleccionades i respectius ciclistes

Figura 41: Fitxa de cas d'ús de Comparar ciclistes

L'usuari ha de poder fer zoom en la gràfica on es mostren totes les trajectòries dels ciclistes, d'aquesta manera podrà veure amb més detalls les dades.

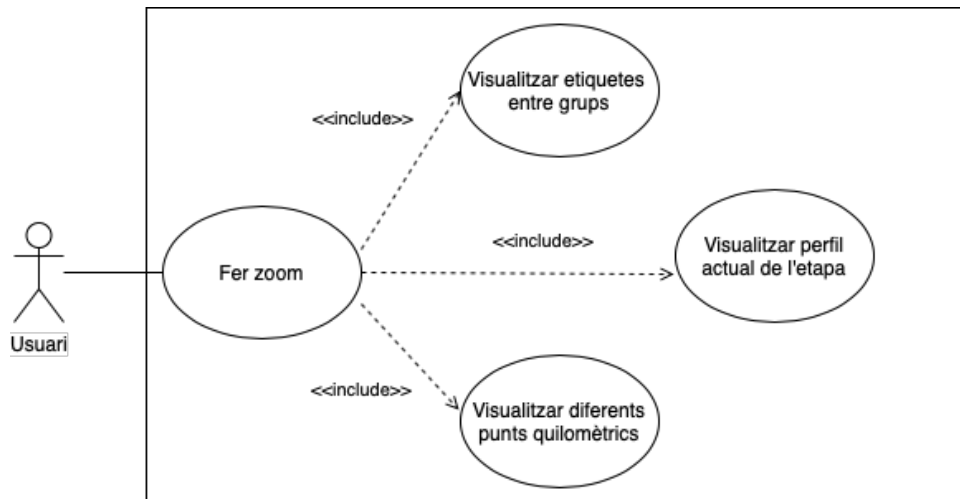


Figura 42: Diagrama cas d'ús de Fer zoom

Cas d'ús	Fer zoom
Versió	Visió anàlisi
Descripció	Un usuari vol fer zoom la gràfica de l'etapa
Actors	Usuari
Precondició	Cap
Flux principal	<ol style="list-style-type: none"> <li>1. Moure la rodeta del ratolí</li> <li>2. Mostrar el resultat en la pàgina</li> </ol>
Flux alternatiu	<ol style="list-style-type: none"> <li>1. Moure amb dos dit amb el touchpad del portàtil</li> <li>2. Mostrar el resultat en la pàgina</li> </ol>
Postcondició	A la gràfica s'aplica l'escala que ha provocat el zoom. Es visualitzen les etiquetes entre grups, el perfil de l'etapa actual i el punts quilomètrics pertinents.

Figura 43: Fitxa de cas d'ús de Fer zoom

L'usuari si ho creu convenient pot fer *drag* en el graf de les trajectòries per visualitzar unes altres dades.

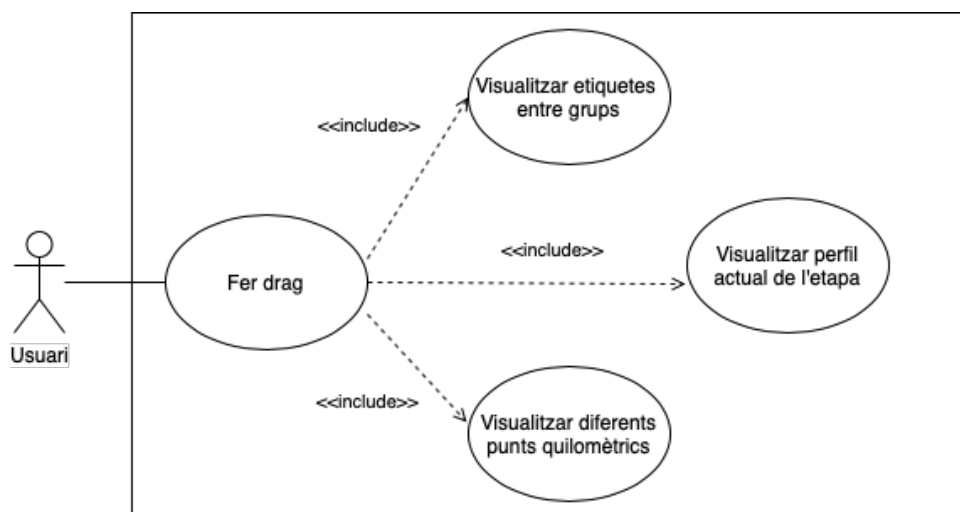


Figura 44: Diagrama cas d'ús de Fer drag

Cas d'ús	Fer drag
Versió	Visió anàlisi
Descripció	Un usuari vol fer drag en la gràfica de l'etapa
Actors	Usuari
Precondició	Cap
Flux principal	<ol style="list-style-type: none"> <li>1. Fer clic esquerra amb el ratolí i mantenir clicat</li> <li>2. <b>Moure</b> ratolí</li> <li>3. <b>Mostrar</b> el resultat en la pàgina</li> </ol>
Flux alternatiu	<ol style="list-style-type: none"> <li>1. <b>Clicar</b> amb el botó esquerra en el <i>touchpad</i> del portàtil i <b>mantenir</b> clicat</li> <li>2. <b>Moure's</b> pel <i>touchpad</i> amb un altre dit</li> <li>3. <b>Mostrar</b> el resultat en la pàgina</li> </ol>
Postcondició	La gràfica ajusta la posició que ha sigut afectada al fer drag. Es visualitzen les etiquetes entre grups, el perfil de l'etapa actual i els punts quilomètrics pertinents.

Figura 45: Fitxa de cas d'ús de Fer drag

## 8.2 Anàlisi i disseny dels elements de la gràfica

En aquest apartat s'argumentarà com s'han arribat a dissenyar i implementar els diferents elements que apareixen a la pàgina web. Aquests elements s'han dissenyat amb l'idea de poder fer zoom, és per això que el disseny de les trajectòries o els polígons s'han tingut que adaptar aquesta funcionalitat.

### 8.2.1 Punts quilomètrics

Identifiquem els punts quilomètrics de la cursa en els quals tenim dades de tots els ciclistes amb una línia vertical amb una etiqueta indicant els quilòmetres restants per finalitzar l'etapa. Un exemple d'això el podem trobar en la figura 46 on hi han un total de 6 línies verticals. Aquesta quantitat de línies verticals en la pàgina web pot ser molt més gran, i això afectarà a la visualització de les dades ja que hi hauria molts masses elements a la vista. La solució que s'ha adoptat ha sigut mostrar un màxim de 4 línies verticals en tot moment, per no donar sobre informació a l'usuari. L'algorisme que s'ha aplicat, que suporta el zoom, s'explicarà més endavant, en l'apartat 9.1.4, un cop tinguem entès com funciona el zoom.

### 8.2.2 Trajectòries

Com s'ha comentat en els apartats anteriors, els ciclistes es representaran com a línies poligonals representant les trajectòries, amb un gruix de 1px. S'ha representat amb aquesta resolució ja que el nombre de participants és molt gran i si s'hagués dibuixat amb més píxels la visualització hagués empitjorat, a continuació en la figura 46 mostrem una petita idea del disseny que s'està plantejant.

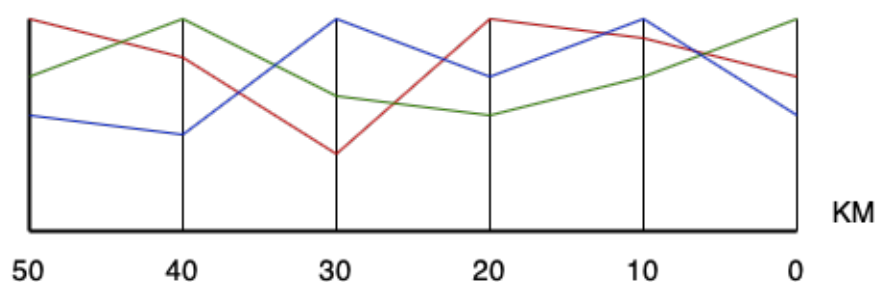


Figura 46: Exemple de les trajectòries de cada ciclista en la web

Aquest exemple serveix perfectament per explicar l'idea que es vol dissenyar e implementar per les trajectòries. L'eix de les  $x$  representa el nombre de quilòmetres que falten per acabar l'etapa. Per tant, comença en 50 quilòmetres i acaba en 0. L'eix de les  $y$  ens hem de quedar amb l'idea que estarà associat al temps però també amb la classificació/posició dels ciclistes en cadascun dels punts quilomètrics. Aquesta etapa té 50 quilòmetres en el que hi ha 3 ciclistes, representats per colors. Cada un d'aquests colors és una trajectòria totalment diferent, i aquesta trajectòria es pot dividir en 5 segments diferents.



Cada un d'aquests 5 trams (segments) són les que s'haurà de generar a partir del fitxer de les dades. Com és evident els extrems de cada segment tindran els punts del tram anterior i del tram següent.

Un altre matís que es vol mencionar és que normalment la unitat de temps en una gràfica ve condicionada de manera ascendent. Però en aquesta gràfica no és el cas, a propòsit es va definir el temps de manera descendent, ja que representava millor les posicions actuals dels ciclistes durant la cursa. Per tant les trajectòries que es troben més a dalt de la gràfica representen els ciclistes que van per endavant de la carrera. En canvi, els que estan més a sota representen aquells participants que tenen un retard de temps més gran en comparació al 1r d'aquell punt quilomètric.

Ara bé, aquest és el disseny de la web, però realment els valors que hi han darrere no són aquests, l'eix de les  $y$  representarà d'alguna manera el temps, aquest augmenta quan més aprop s'està de l'eix de les  $x$ . I per altra banda que l'eix de les  $x$  vagi de més a menys no ens serveix per definir unes coordenades cartesianes vàlides. És per aquesta raó que per codificar s'han utilitzat uns altres valors. En l'imatge que s'adjunta a continuació (figura 47) es pot veure quins són els orígens de coordenada que s'han utilitzat per guardar les dades de les trajectòries. Aquests són els valors inicials un cop es carrega la pàgina.

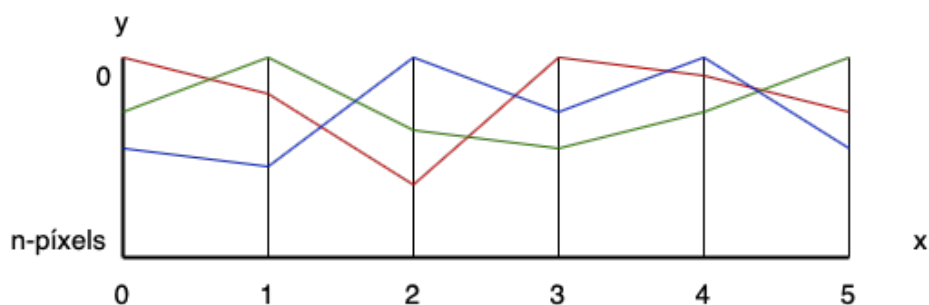


Figura 47: Exemple de les coordenades cartesianes de cada ciclista en la web

Un cop ha quedat clar com s'han dissenyat les trajectòries ja es pot arribar a implementar a la pàgina. En l'apartat 5.2.3 s'ha comentat quines dades s'obtenen dels ciclistes, si aquestes dades que es volen representar les representem en una gràfica, diferenciant cada un dels ciclistes i visualitzant la seva evolució està clar que no seran dades que vegin reflectides correctament. Vegem un exemple a continuació.

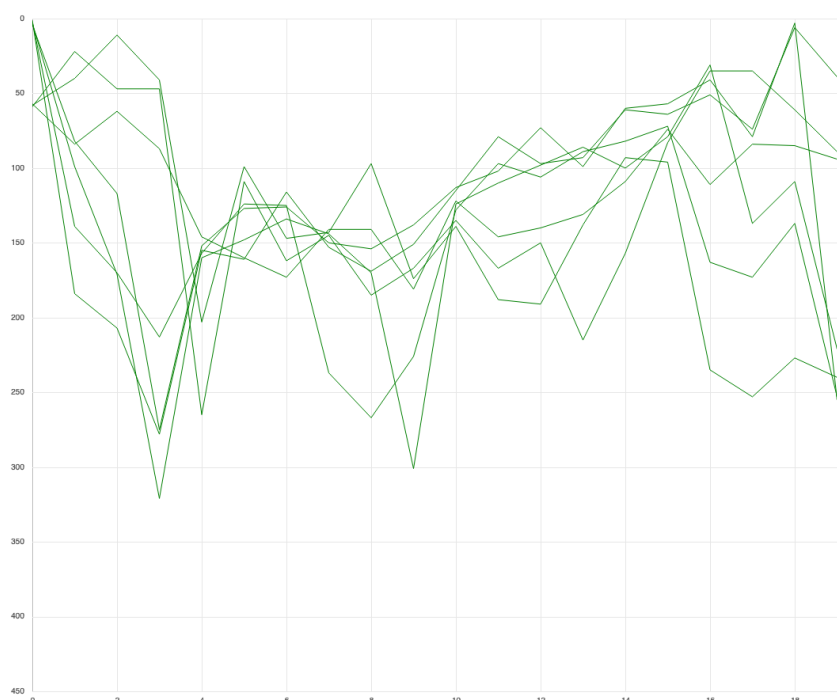


Figura 48: Gràfica de les trajectòries de cada ciclista, v1

Si ens fixem en aquesta figura podem pensar que el nombre de participants o nombre de trajectòries no es tan gran com ens pensàvem. Però això es mentida, ja que les trajectòries hi són, però estan superposades una a sobre de l'altre. Per això hem de buscar una solució per resoldre aquest conflicte.

És en aquest punt on la posició d'un ciclista en un punt quilomètric juga un paper elemental en la representació. En una cursa, la posició de cada un dels participants pot variar a mesura que avança el recorregut. Si a més a més es tracta d'una cursa de llarga distància, aquesta variació es pot veure modificada constantment, però per més que variï cap d'aquestes posicions es veurà repetida en el mateix punt quilomètric. Això es una gran avantatge, ja que ens solucionarà el petit problema que em comentat. Sumant el *ET* i la posició del ciclista en aquell punt de control, aquestes repeticions desapareixeran. En la taula següent es mostra amb més claredat l'idea esmentada.

Ciclista	Km1			Km2			Km3		
	Posició	ET	Pos + ET	Posició	ET	Pos + ET	Posició	ET	Pos + ET
A	1	0s	1	2	7s	9	3	4s	7
B	2	10s	12	1	0s	1	2	2s	4
C	8	20s	28	12	8s	20	1	0s	1
...	...	...	...	...	...	...	...	...	...
Z	4	10s	14	4	7s	11	5	4s	9
AA	6	10s	16	34	67s	101	4	4s	8
AB	10	23s	33	22	12s	34	7	8s	15
...	...	...	...	...	...	...	...	..	...

Figura 49: Exemple del transcurs del temps respecte el primer

Un cop es té aquesta idea, s'ha de modificar el codi per implementar la nova gràfica que veiem a continuació.

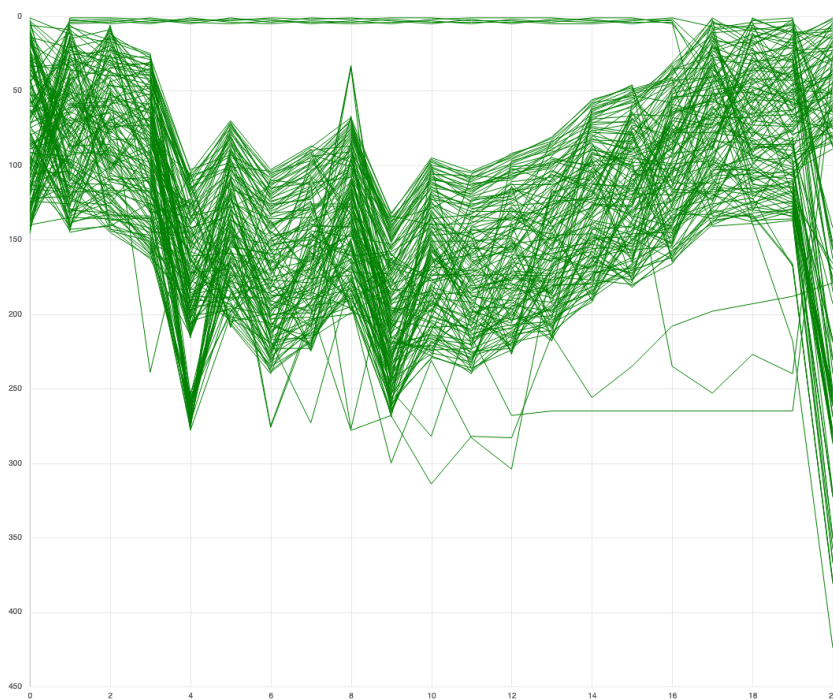


Figura 50: Gràfica de les trajectòries de cada ciclista, v2

En aquesta nova gràfica podem veure quina és la nova representació de les trajectòries, aquestes no estan juntes sinó que ara apareixen una sota de l'altre utilitzant la posició dels ciclistes per cada un dels punts quilomètric.

Ens podem fixar que durant tota l'etapa les trajectòries es van creuen unes amb les altres. En una de les reunions amb els membres del projecte vam arribar a pensar de mantenir un cert ordre en els segments de manera que no es creuessin, això implicava canviar les coordenades dels extrems dels seg-

ments, i en cert punt també la posició del ciclista en un punt quilomètric concret. Amb aquest nou canvi la visualització de les trajectòries hagués estat més ordenada, però per una altra banda les dades no mostrarien els valors exactes de cada ciclista. Per aquesta raó vam arribar a la conclusió de que l'idea original, d'utilitzar les dades de posició en cada punt quilomètric i l'ET, era una millor representació.

Tornant a la figura 50, es pot comentar que en gran part de l'etapa sempre hi hagut els mateixos ciclistes liderant les primeres posicions. Aquests són els que en l'eix de les  $y$  sempre han estat dintre del rang 0 i 25 aproximadament. Pel que fa a l'eix  $x$  i  $y$  farem un petit comentari:

- L'eix  $x$  té un rang de 0-20, en total hi han 21 eixos verticals, on cada un d'aquests representa un punt quilomètric. Més endavant es canviarà per la numeració de quilometratge pertinent.
- L'eix  $y$  té un rang de 0-450. Més endavant vam arribar a la decisió de no mostrar aquests valors ja que podrien arribar a la confusió, perquè no representen ni els segons transcorreguts ni tampoc la posició. La informació de posició i  $ET$  es mostrarà en *hovers* que apareixen un cop es selecciona una trajectòria, disseny que explicarem més endavant.

Ara que tenim aquesta idea clara podem explicar com es detecten tots aquests grups. Per cada punt quilomètric es formaran un conjunt de grups els quals es generaran de la següent manera:

- Es llegirà seqüencialment de dalt cap abaix cada un dels  $ET+Posició$  de cada ciclista.
- Es crearà un grup i s'aniran afegint aquests valors  $ET+Posicio$ , en seqüencial, un darrere de l'altre, sempre i quan aquests valors siguin consecutius, és a dir, que la diferència de  $(ET + Posicio)_{i+1}$  i  $(ET + Posicio)_i$  sigui igual a 1.
- Quan la condició esmentada abans es deixa de complir, llavors es crea un nou grup i es van afegint els valors restants de  $ET+Posició$  usant la mateixa condició d'abans
- Aquest procés es repeteix fins arribar a l'últim valor  $ET + Posicio$  d'aquest punt quilomètric

Un petit exemple de valors  $ET+Posicio$  podria ser 2,3,4,5,8,9, aquests valors correspondrien a dos grups acceptables(2,3,4,5), (8,9). En canvi si obtinguéssim els valors 2,3,4,5,6 no es generarien dos grups, ja que no trenquen la cadena, es generaria un únic grup definit pels valors (2,3,4,5,6).

Si tornem a la figura 50 ara ja ens podem fer una lleugera idea de com es van formant els grups. Amb aquesta metodologia, totes les etapes tindran una característica en comú i es que al principi de l'etapa sempre hi haurà un únic grup format per tots els ciclistes. Ja que per tots es complirà  $ET = 0$ , per tant l'únic condicionant és la seva posició.

### 8.2.3 Representació del grups

En una de les reunions amb els membres del projecte és va plantejar la idea de representar els grups de dues maneres, amb diagrames Sankey o amb representació poligonal de 4 costats. En la figura 51 es pot veure un exemple de diagrama de Sankey. Aquest disseny es va descartar ja que es volia mantenir l'ordre d'arribada dels ciclistes en cada punt quilomètric, i com que el Sankey uneix els grups d'un punt quilomètric amb el següent de la millor manera que considera, això comportaria que les trajectòries i els grups fossin independents, cosa que visualment no ens convé. Aquest inconvenient s'hagués pogut arreglar fent que les trajectòries no es creuessin, per fer això s'hauria de modificar les posicions dels ciclistes en cada punt quilomètric, i com ja s'ha comentat prèviament aquesta idea no ens acabava de convèncer ja que no representava correctament l'evolució dels ciclistes durant l'etapa. Per això vam descartar la idea de diagrames de Sankey i utilitzar la representació amb polígons.

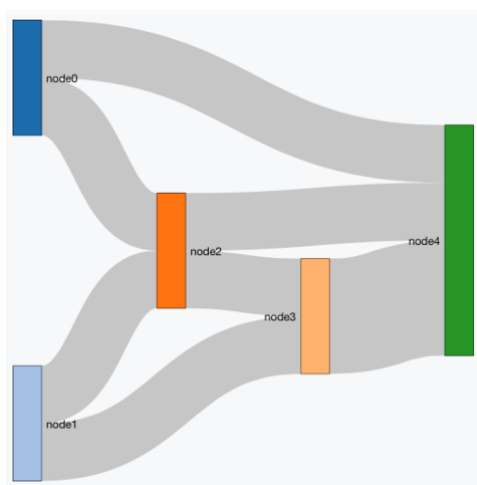


Figura 51: Exemple de diagrama Sankey

Com hem comentat l'element visual que hem escollit ha sigut un polígon de quatre costats. On els costat de l'esquerra i de la dreta coincideixen amb l'eix de les  $y$ . Els vèrtex d'aquest polígon vindrien condicionats per el primer i per l'últim ciclista que hi hagués en el grup. En aquest exemple que veiem a continuació (figura 52) observem que es formen dos polígons, un de color taronja i l'altre de color verd. Això és perquè del grup que hi havia en el tram anterior s'han format dos grups nous. Aquests dos polígons estan superposats, aquest fet es repetirà durant tota la gràfica ja que els grups que es formen es poden dividir o fusionar, i aquest efecte és inevitable.

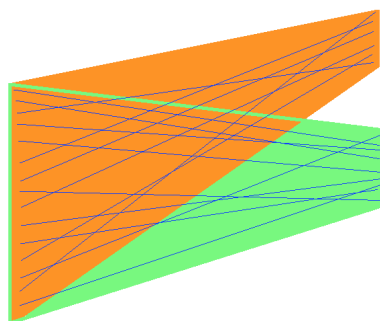


Figura 52: Primera versió de la representació dels grups

Un cop s'ha dissenyat la idea de com representar-ho és el moment d'implementar-ho. En la figura 53 es mostra una captura de la generació dels polígons. A diferència del disseny anterior que té colors diferents, en la pàgina web hem volgut implementar-ho només amb un color ja que representa millor el flux de l'evolució dels grups. És per això que trobar els polígons amb els respectius vèrtex és molt difícil ja que es solapa un polígon amb un altre, però com s'ha dit l'objectiu no es representen els polígons, sinó visualitzar l'evolució dels grups que es formen.

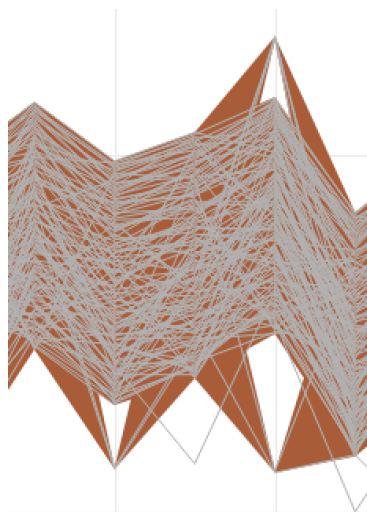


Figura 53: Exemple de polígon amb v1

Després de codificar aquesta representació, ens vam donar compte que la visualització no acabava d'ajustar-se del tot, ja que l'àrea de cada polígon que es dibuixava era més gran del que hauria de ser. És per això que vam començar a fer una segona versió per poder ajustar una mica més els polígons.

En la següent versió es parteix de l'exemple anterior, s'observa que els polígons originals són els de color taronja i verd. I els polígons que es defineixen al final són el de color groc i verd. En aquesta nova

versió els vèrtex (per cada polígon) del costat esquerre no serien el primer i l'últim de cada grup. Sinó que com a 1r vèrtex del polígon (a dalt a l'esquerra) s'escolliria la primera trajectòria que en el punt quilomètric següent formés un nou grup. I el 2n vèrtex (abaix a l'esquerra) s'escolliria l'últim ciclista que formés part del nou grup que la primera trajectòria a generat. Aquesta nova idea i la seva aplicació en el graf queda de la següent manera.

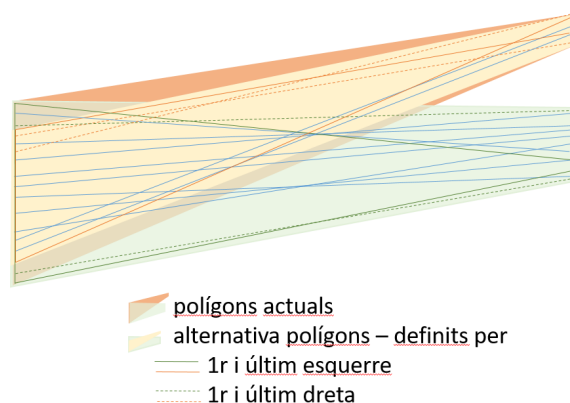


Figura 54: Segona versió de la representació dels grups

En la figura 55 és pot veure quina és la diferència que hi hagué en la representació dels polígons. El canvi no es molt exagerat però això és un exemple puntual, en el moment de representar la gràfica sencera llavors sí que haurà una canvi notable.

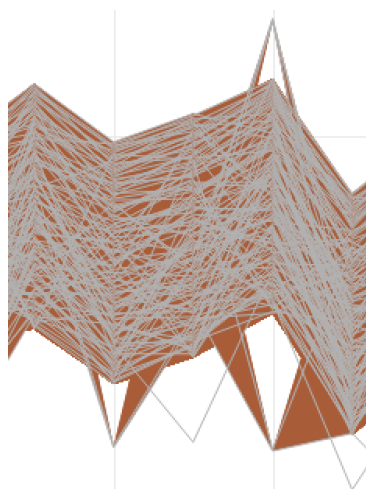


Figura 55: Exemple de polígon amb v2

Un cop s'ha implementat l'algorisme correcte per formar correctament els polígons, vam pensar en fer una petita modificació en el color amb el que es pinta. Fins el moment hem pintat els polígons amb el mateix color i per tant encara que es solapessin no hi veuríem cap diferència, aquesta va ser la primera

idea, però parlant-ho amb els membres del projecte es va decidir en pintar-ho amb una mica d'opacitat, d'aquesta manera es veuria quina ha sigut l'evolució dels grups durant els trams de l'etapa, en la figura següent podem veure quin és el resultat final.

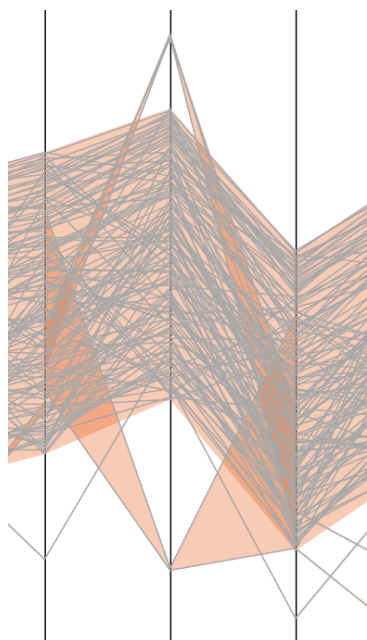


Figura 56: Exemple de polígon amb v3

En aquest exemple és pot diferenciar que en el primer eix vertical hi ha un sol grup, en canvi en el tram següent veiem que s'han format tres grups (la punxa d'amunt, la part del mig i la punxa d'abaix), per tant s'han generat tres polígons. Del segon eix vertical cap al tercer podem observar que es tornen a unificar en un de sol. En el darrer eix vertical també s'observa que abaix de tot hi arriba una única trajectòria que no forma grups amb ningú, per tant tampoc es visualitza cap polígon.

#### 8.2.4 Informació temporal

Un cop es té informació visual del grups que es formen, també era important mostrar quina diferència de temps hi havia entre dos grups consecutius d'un mateix tram quilomètric. El fet que l'eix y no correspongui al temps fa que no hi hagi cap informació relativa al temps transcorregut o diferència en temps entre dos grups de ciclistes diferent. Per això es va decidir afegir informació addicional a la gràfica usant etiquetes.

Per dissenyar aquesta tasca es va decidir pintar la diferència de temps just en mig dels dos grups. Per tant, es va dissenyar un petit algorisme per trobar totes aquestes etiquetes. El primer que es fa es crear un *array* buit on es guardaran totes les etiquetes, seguidament s'obté un grup de ciclista i el següent grup



que ve després d'aquest. Quan es tenen aquests dos grups, s'ha de trobar les coordenades cartesianes on es dibuixarà l'etiqueta. El valor de la  $x$  és l'índex del punt quilomètric que s'està tractant i el càlcul per trobar el valor de la  $y$  el podem observar a la línia 10 de la figura 57. I per últim només s'ha de buscar quin és el valor de diferència de l'eix de les  $y$  entre el primer ciclista del grup següent i l'últim ciclista del primer grup.

```

1 // Buscar les etiquetes
2
3 Etiquetes = []
4 i = 0
5 PER cada grup de ciclista que hi ha en els punts quilomètrics
6
7     G1 = grup[i]
8     G2 = grup[i+1]
9     x = index del punt quilomètric
10    y = G1.ultimCiclista.y + ((G2.primerCiclista.y - G1.ultimCiclista.y) / 2)
11    TempsEtiqueta = G2.primerCiclista.y - G1.ultimCiclista.y
12    Afegir a Etiquetes x, y i TempsEtiqueta
13    i++
14
15 FIPER

```

Figura 57: Petit algorisme per trobar les etiquetes.

Un cop s'obtenen totes les etiquetes que hi ha entre els grups, ja es poden dibuixar a la gràfica (figura 58). En aquest exemple en concret podem veure que s'han pintat molts elements a la pàgina, això dona una sobre informació, ja que hi han moltes etiquetes i això dificulta l'anàlisi i a l'usuari no li beneficia tantes dades de cop. És per això que parlant amb els membre del projecte vam arribar a la conclusió de mostrar només aquelles etiquetes que complien dues condicions:

- Que la relació del temps de diferència entre grups (el nombre de píxels que hi ha entre els grups) i el zoom sigui l'adequat per mostrar l'etiqueta. És a dir, mostrar aquelles etiquetes que tinguin un espai considerable entre grup i grup.
- L'etiqueta es mostrarà únicament si l'eix de les  $y$  en aquell punt quilomètric també es dibuixa en la gràfica.

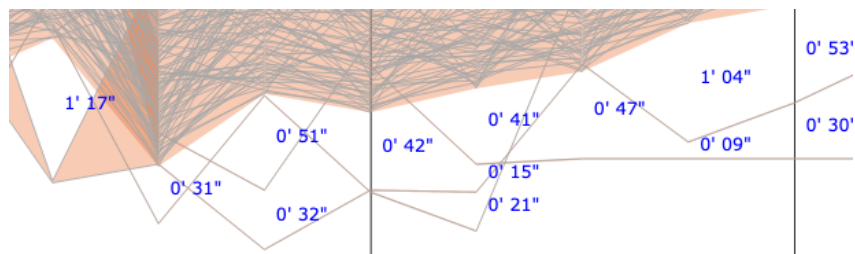


Figura 58: Exemple d'etiquetes entre els grups.

### 8.2.5 Perfil de l'etapa

Un dels altres requisits que necessita la pàgina és mostrar en tot moment el perfil de l'etapa, entenem per perfil de l'etapa una imatge que s'afegirà a sota de la gràfica i mostrarà quin és el desnivell que hi ha durant tota la cursa. Si hem escollit una imatge és perquè en la pàgina de Pro Cycling [17] es pot trobar fàcilment aquesta informació. Una altra funcionalitat que ha de tenir aquest perfil de l'etapa és donar informació sobre quin és el tram de la cursa que s'està visualitzant en aquell moment, ja que depenent del zoom, serà un tram o un altre.

La primera opció era utilitzar una eina de D3 anomenada *brush* [18]. Aquesta eina el que permet és adaptar la mida de la zona ombrejada de manera que a la vegada la visualització de la gràfica s'adapta en el nou estil. De la mateixa manera, si fa zoom o es mou la gràfica, la zona ombrejada també s'adapta en aquesta nova vista. A més a més els requisits visuals que necessitàvem era justament aquests.

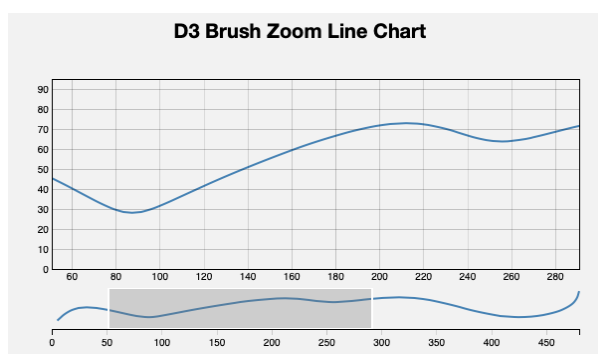


Figura 59: Exemple de *brush* en un diagrama amb D3.

Però quan es va començar a veure la documentació i com s'havia d'implementar ens vam donar compte de que no disposàvem amb les dades necessàries per implementar el *brush*. Aquesta eina utilitza dades en comú per la gràfica de dalt, com la gràfica de sota. És per això que no podem implementar-ho de la mateixa manera, ja que nosaltres disposem d'una gràfica (les trajectòries) i d'una imatge (el perfil de l'etapa). A més a més, són conceptes totalment diferents, ja que en una es mostra l'evolució del ciclistes i en el perfil es mostra el desnivell de l'etapa. És per això que la nostra solució ha sigut utilitzar un element SVG per representar aquesta ombra, en concret un rectangle. I a partir de jQuery i CSS modificar els elements necessaris:

1. Crear un contenidor SVG, amb les mides necessàries on es crearà el rectangle
2. Afegir el rectangle dintre del contenidor creat, amb les mides i l'estil que necessitem. En concret deixar en translúcid el rectangle, perquè la imatge del perfil de l'etapa sigui visible.

- $x$  és la posició on es començarà a dibuixar el rectangle.
  - $width$  és l'amplitud del rectangle
3. Cada cop que s'aplica zoom a la pàgina actualitzar els valors  $x$  i  $width$  utilitzant la informació que ens aporta D3 sobre l'escala actual que s'està aplicant.

### 8.2.6 Informació d'una trajectòria

Un cop els elements principals, com les trajectòries, polígons, punts quilomètrics (línies verticals) i la informació principal dels ciclistes i les etapes ja estan implementades en la web. És el moment de començar a facilitar petites tasques que l'usuari pugui executar. Les trajectòries en la pàgina web estan configurades amb un *hover*, d'aquesta forma cada cop que un usuari faci *mouse over* per les trajectòries podrà veure amb exactitud quina és la trajectòria que selecciona, en la figura 60 en podem veure un exemple.

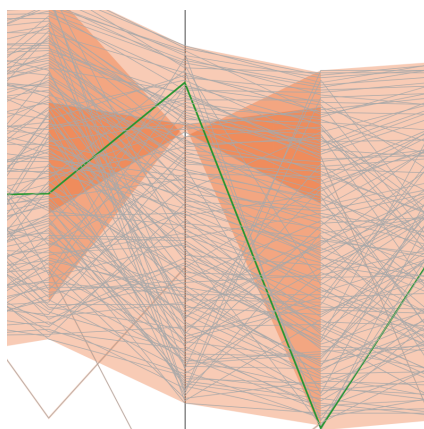


Figura 60: Exemple de fer *hover* sobre d'alguna trajectòria

Un cop s'ha seleccionat, ja sigui amb el mouse o elegint un ciclista amb el cercador el nou estil de la trajectòria quedarà fixe i automàticament apareixeran nous elements a la trajectòria i estaran situats exactament on es creuen el punts quilomètrics. Aquests punts creats s'utilitzaran per donar més informació del ciclista. S'utilitzarà un *tooltip* per cada punt, *tooltip* és bàsicament un estil de CSS creat expressament per donar més informació un cop es fa *mouse over* sobre de l'element. La informació que s'ha volgut donar ha sigut: el retard respecte al primer de l'etapa en aquell quilòmetre (Elapsed time), la posició provisional a la classificació general en aquell quilòmetre (Provisional Position) i el retard provisional respecte el líder de la classificació general en aquell quilòmetre(Provisional Delay).

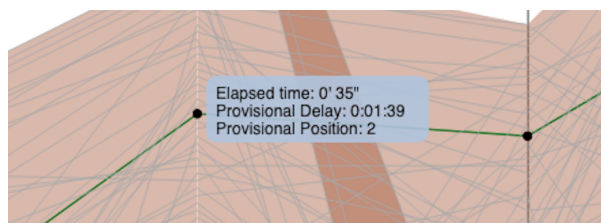


Figura 61: Ciclista seleccionat i tooltip activat

En la selecció d'un equip s'ha aplicat aquesta mateixa idea del *tooltip*. Però en aquest cas hi ha més d'una trajectòria seleccionada, i encara que es tingui informació de quin equip es tracta, també es necessari saber quin ciclista representa cada trajectòria, juntament amb les dades extra que té cada trajectòria. És per això que la solució que vam trobar era crear un nou element *tooltip* on es mostra quin és el ciclista que es selecciona, les dades a mostrar són: el nom del ciclista, el número de dorsal i les inicials de l'equip. A continuació es motra una petita captura del resultat final.

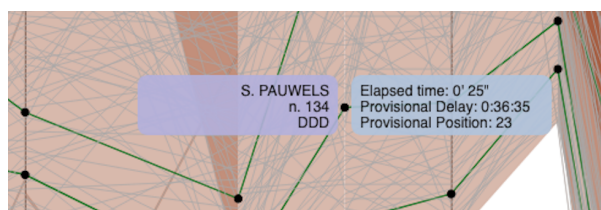


Figura 62: Equip seleccionat i tooltip activat

### 8.2.7 Zoom de la pàgina

Una de les funcionalitats més importants que té aquesta aplicació és la de tenir la possibilitat de fer zoom en els elements que apareixen en la pàgina web. Quan s'inicia la pàgina web, automàticament s'ha de carregar la pàgina i mostrar tota la informació de totes les trajectòries, al haver-hi una gran quantitat de participants, el número de trajectòries que hi ha és molt gran, això fa que no es pugui veure amb detall quina és l'evolució del ciclista en tota l'etapa. És per això que poder fer zoom és primordial per fer un anàlisi correcte de les dades. Com ja s'ha comentat anteriorment, la llibreria que s'ha utilitzat per aplicar això ha sigut D3. Amb aquesta mateixa llibreria s'han pogut afegir i configurar el comportament de altres elements que formen l'aplicació, com són: les trajectòries, els punts quilomètrics, etiquetes, polígons, *tooltips* i perfil de l'etapa.

La funcionalitat del zoom aplicant D3 és ideal perquè un usuari pugui interactuar de manera més còmode en l'aplicació web. Aquesta llibreria està dissenyada per respondre als moviments bàsics que coneixem per fer zoom, com per exemple moure la rodeta del ratolí o utilitzar el *trackpad* del portàtil. A més a més del zoom també incorpora el *panning*, que és conegut com mantenir clicat el botó del ratolí i moure'l

al gust de l'usuari, fent que la gràfica es mogui en el mateix sentit. Totes aquestes funcionalitats són les que es volen aplicar al nostre projecte. És per això que s'ha de configurar aquesta llibreria perquè modifiqui tots els elements que apareixen a la gràfica i a més a més la visualització del perfil de l'etapa.

La llista d'elements a tractar són:

- Eixos de les x i de les y, de la gràfica.
- Col·locació dels diferents eixos verticals que representen els punts quilomètrics.
- Les trajectòries de cada un dels ciclistes.
- Els *tooltips* que es situen en les trajectòries.
- Polígons que representen la generació dels grups de ciclistes durant l'etapa.
- Etiquetes que mostren la diferència de temps que hi hagut entre els grups d'un mateix punt quilomètric.
- Zona ombrejada del perfil de l'etapa, que no forma part de la gràfica però igualment s'ha d'actualitzar la mida

### 8.3 Anàlisi i disseny del codi

Quan es va dur a terme el procés de codificació del projecte, vaig a començar a seguir l'estructura dels models de projectes D3 que anava trobant per internet. Tots els projectes parteixen d'un fitxer on es codifica les funcionalitats del D3 i d'un altre fitxer .csv que conté totes les dades necessàries.

Per això vaig seguir el mateix procediment creant un fitxer per crear el graf. A mesura que anava avançant en el projecte em vaig donar que implementar-ho tot en un mateix fitxer no era gens òptim ni còmode, és per això que vaig decidir seguir una estructura i utilitzar diferents de fitxers per diferents funcionalitats. Els fitxers que al final s'han implementat són:

- visualitza.html: fitxer que serveix per estructurar els elements en la pàgina web.
- style.css: fitxer que defineix l'estil de la pàgina.
- stages.js: fitxer que gestiona la barra lateral esquerra on apareixen les diferents etapes que es poden seleccionar.
- riders-info.js: fitxer que gestiona les dades dels ciclistes, nom, número, icona..
- riders-graph.js: fitxer que genera el graf i crea totes les funcionalitats que es necessiten.
- interaction.js: fitxer que gestiona la interacció amb l'usuari. Un cop es carreguen tots els elements de la pàgina es necessari gestionar totes les interaccions que té l'usuari amb els elements de la pàgina.

#### 8.3.1 Diagrama de classes

A continuació es mostrarà quin és el disseny dels diferents fitxers que s'han utilitzar per implementar el projecte, juntament amb les seves funcions. Concretament s'han creat diferents fitxers JavaScript on cada un s'encarrega de llegir les dades necessaris i de representar gràficament els elements a la web. També es mostra quina és l'estructura dels elements de la pàgina web a partir del fitxer HTML. És important mantenir un cert ordre d'execució dels fitxers ja que hi ha variables globals que s'utilitzen durant el procés d'inicialització de la pàgina. A continuació es mostra les diferents captures d'imatges que s'han generat amb el projecte Wavi que s'ha comentat en l'apartat 7.3.3

En la figura 63 que mostrem a continuació es mostra l'estructura dels diferents elements de la pàgina web. Aquest ordre és molt important ja que tots els objectes que es vagin afegint a partir dels fitxers JavaScript aniran en la posició on s'ha definit en aquest fitxer, juntament amb l'estil del fitxer CSS.

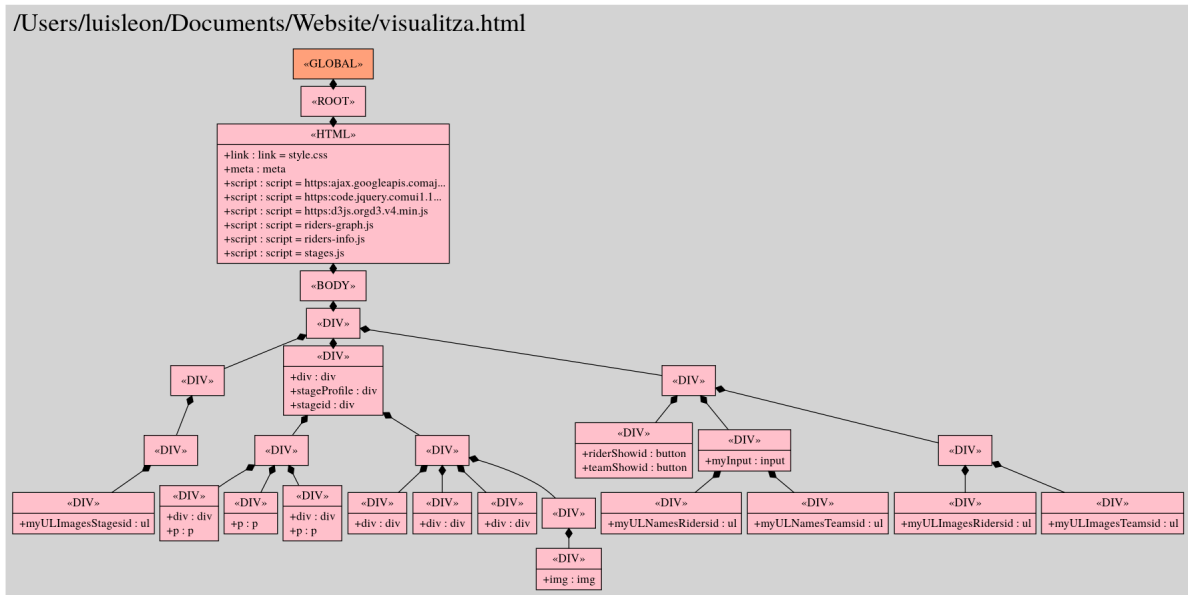


Figura 63: Diagrama de classes del fitxer index.html

En la figura 64 es mostra quin és el disseny de classes del fitxer stages.js. Aquest s'encarrega primer de tot de definir quines són les etapes que participen en la cursa, amb les dades de cada una d'aquestes (númeroEtapa, origen, destí i distància que hi ha). Un cop es tenim definides aquestes dades, es llegeixen totes les imatges dels perfils de l'etapa, que aniran en la barra lateral dreta, juntament amb la seva informació. Apareixen algunes variables globals com per exemple:

- currentStage = 13 que és l'etapa actual que es mostrarà quan s'inicia l'aplicació.
- folder, és el path on estan guardades les diferents imatges de ciclistes i equips, variable que utilitzaran més endavant altres fitxer
- folderStages = path on hi han les imatges dels perfils de l'etapa

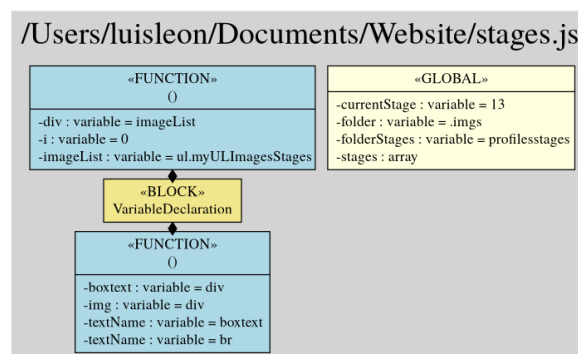


Figura 64: Diagrama de classes del fitxer stages.js

El fitxer riders-info.js crea alguns elements que apareixen en la barra lateral dreta, com per exemple: el cercador, mostrar els noms i informació dels ciclistes o equips. El procés que es segueix és: primer de tot es llegeixen les dades del fitxer Rider.csv, juntament amb les seves imatges i es carreguen a la pàgina, si tot ha anat correctament fa el mateix procés per la informació dels equips i pels respectius noms. Com a variables globals podem veure que la gran majoria són imatges, aquestes imatges representen els diferents tipus de mallots que hi han per cada tipus de líder (explicació dels diferents tipus de lider en l'apartat 5.2.1 . En l'apartat 9.2.3 es mostra un exemple del que s'està esmentant.

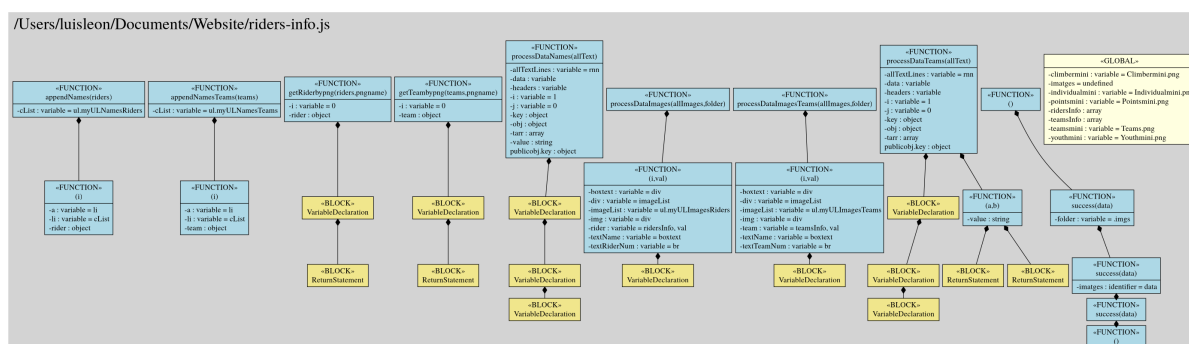


Figura 65: Diagrama de classes del fitxer riders-info.js

A continuació es comentarà el fitxer riders-graph.js que s'encarrega de gestionar la part central de la pàgina web. És aquest fitxer el que implementarà el gràfic i tots els seus elements. Aquest tipus de fitxer s'utilitzarà per qualsevol de les etapes que seleccioni l'usuari. En el moment d'iniciar l'aplicació web, per defecte, es visualitzarà l'etapa número 13, ja que així ho hem definit, però cada cop que es seleccioni qualsevol altre etapa, sempre i quan es tinguin dades d'aquella etapa, es tornarà a executar el fitxer riders-graph.js, però aquest cop es mostrarà un altre gràfic amb les dades de l'etapa que s'hagi seleccionat. Aquest fitxer conté una gran quantitat de funcionalitats, és per això que el nombre de classes és molt gran. S'adjuntaran 4 imatges que encapsularan el fitxer riders-graph.js (figures 66–69).

En la figura 66 per exemple podem veure la funció de zoomed que és on s'ha implementat el comportament dels elements alhora de fer zoom.



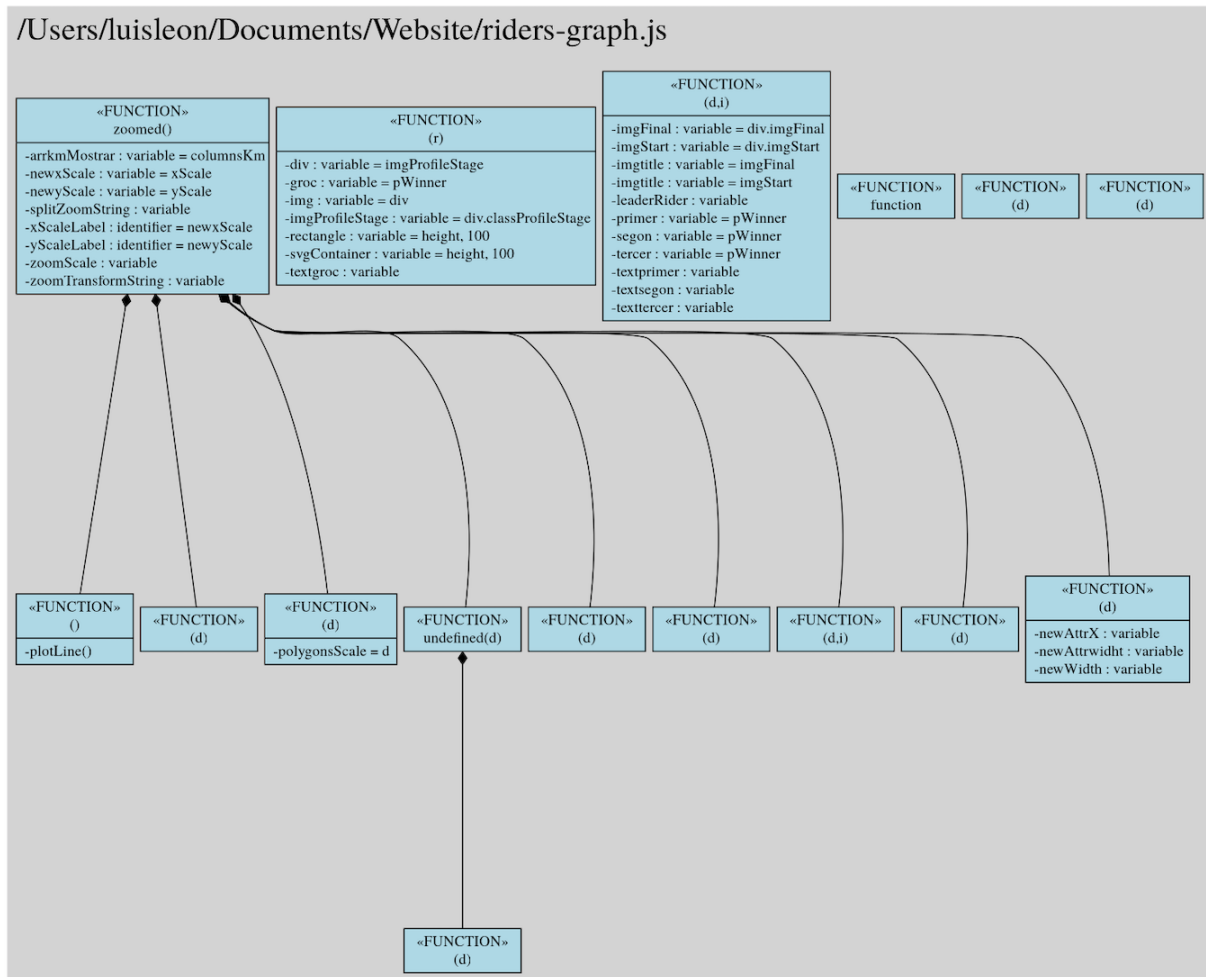


Figura 66: Diagrama de classes del fitxer riders-graph.js (1)

En la imatge següent, figura 67, es pot veure quines són les variables globals que s'utilitzaran, algunes d'aquestes són:

- `elapsedDataRider`: un array que guardarà totes les trajectòries de tots els ciclistes en els diferents punts quilomètrics d'una etapa
- `separacioEntreKM`: aquesta variable definirà quina és la distància que hi ha entre punts quilomètrics. S'utilitza en el moment de buscar quins punts quilomètrics s'han de mostrar. I s'ha hagut de crear, ja que aquesta distància entre quilòmetres és diferent segons en quina etapa.
- `showNAxisChart`: nombre màxim de línies verticals que es volen mostrar

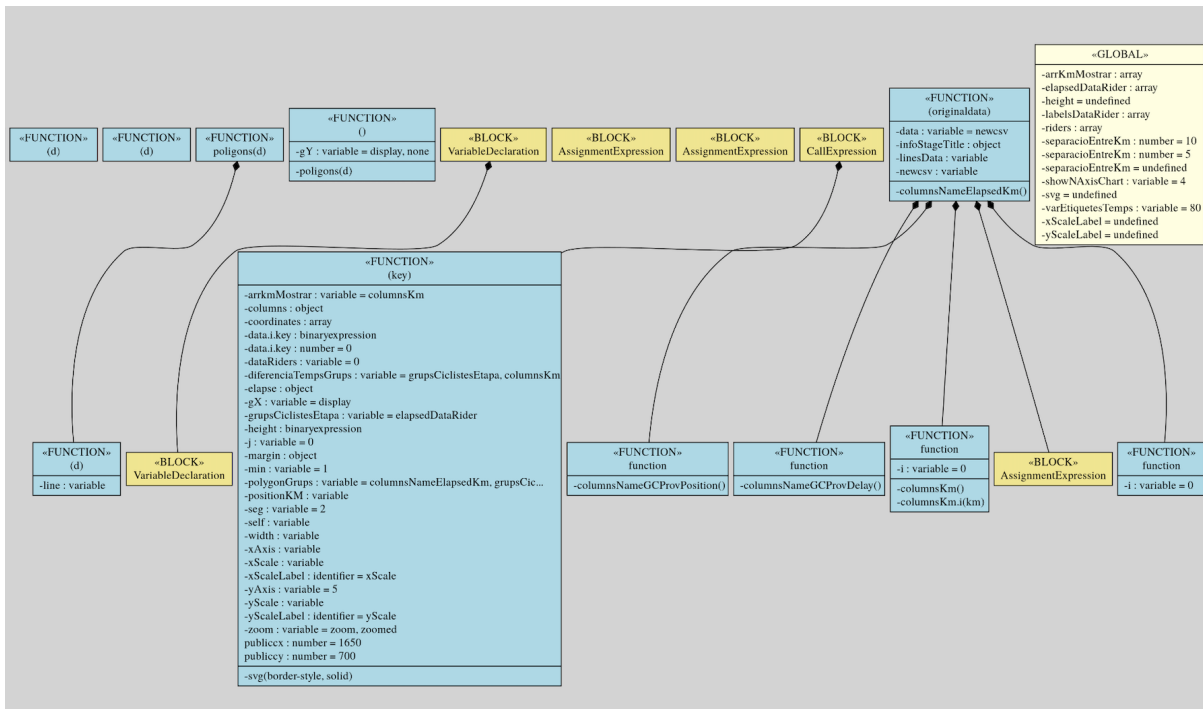


Figura 67: Diagrama de classes del fitxer riders-graph.js (2)

En la següent captura d'imatge, la figura 68, comentarem dues funcions, la primera "buscaDiferenciaTempsGrup", aquest mètode s'utilitza per buscar quina es la diferència de temps que hi ha entre dos grups consecutius d'un mateix punt quilomètric i també per saber en quina posició de la gràfica pintar aquest valor. L'altre funció a comentar és "valorElapsedMax", aquesta funció retorna quin ha sigut el valor de *ET + Posicio* més gran que hi hagué en tota l'etapa. Aquest valor s'utilitza per generar la mida del graf i a més a més per calcular quines etiquetes o línies verticals es mostraran quan es faci zoom.

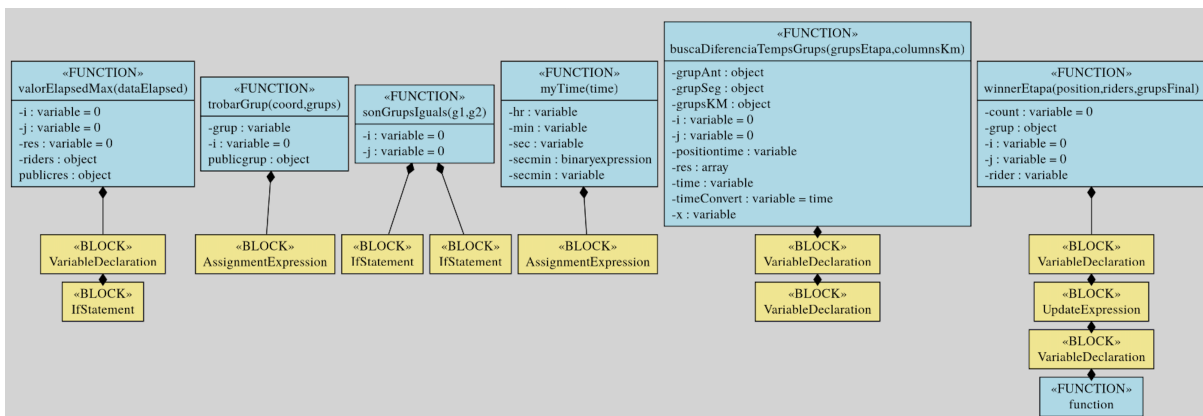


Figura 68: Diagrama de classes del fitxer riders-graph.js (3)

En l'última captura que s'adjunta del fitxer riders-graph.js podem veure alguns algorismes que s'han

tingut que implementar: "getGroupRidersStage", "getPolygonsGroups", "updateAxisXKM" aquests diferents algorismes serveixen per trobar els grups que es generen en l'etapa actual, els polígons que es formen en els grups i saber quins eixos verticals mostrar respectivament. Aquests mètodes es comentaran en el punt 9.1.

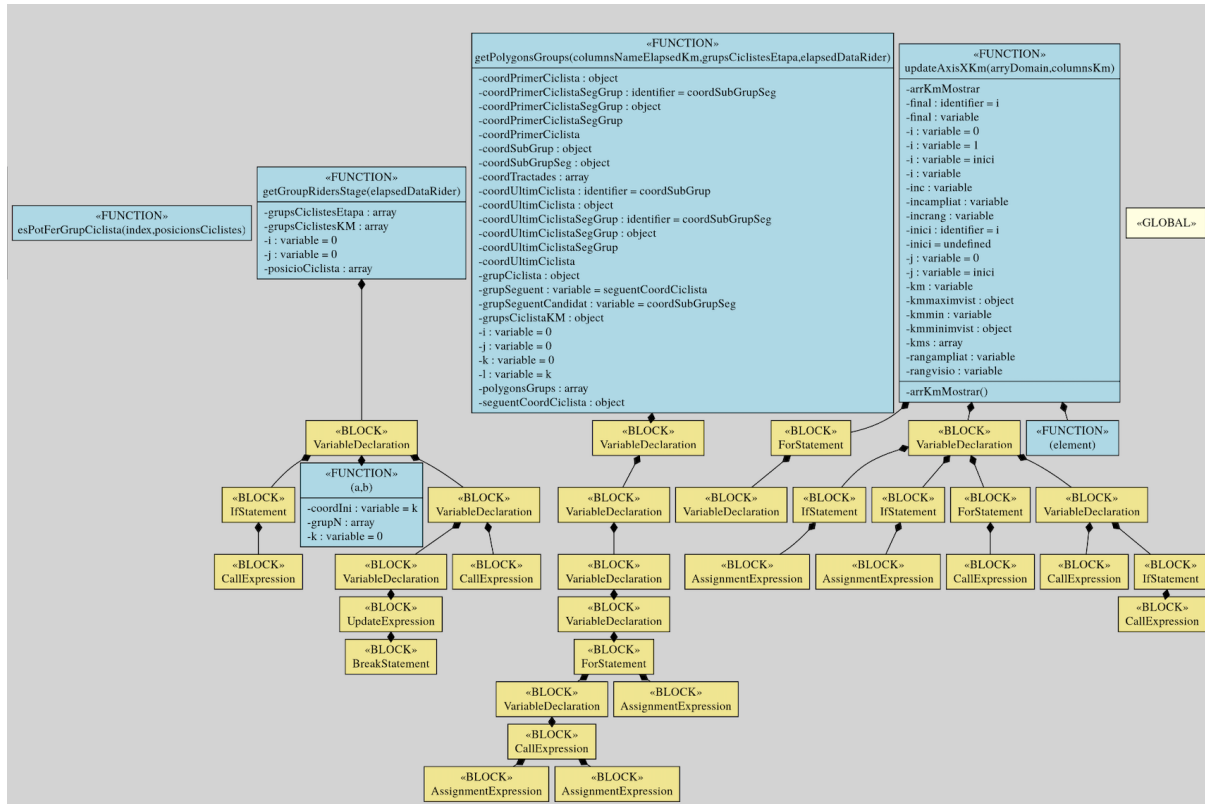


Figura 69: Diagrama de classes del fitxer riders-graph.js (4)

Un cop s'han generat tots elements de la pàgina, és el moment de llegir l'últim fitxer que s'ha anomenat interaction.js. Aquest fitxer és el que s'encarrega de controlar totes les interaccions que hi ha amb l'usuari. És important que aquest fitxer sigui l'últim en executar-se ja que totes les funcionalitats que s'utilitzen es fan amb la llibreria jQuery. Aquesta llibreria s'utilitza per controlar el comportament de la pàgina, com per exemple si fa clic a una trajectòria o a un ciclista. L'únic event que en aquest fitxer no apareix, és el de la funcionalitat de zoom. El zoom recordem que el gestiona el mateix fitxer (riders-graph.js) que ha iniciat el mètode i que ha declarat el seu comportament. Aquest fitxer com passava amb l'anterior també té un gran nombre de funcions és per això que les classes s'han dividit en diferents captures perquè és es pugui visualitzar millor (figures 70 – 73).

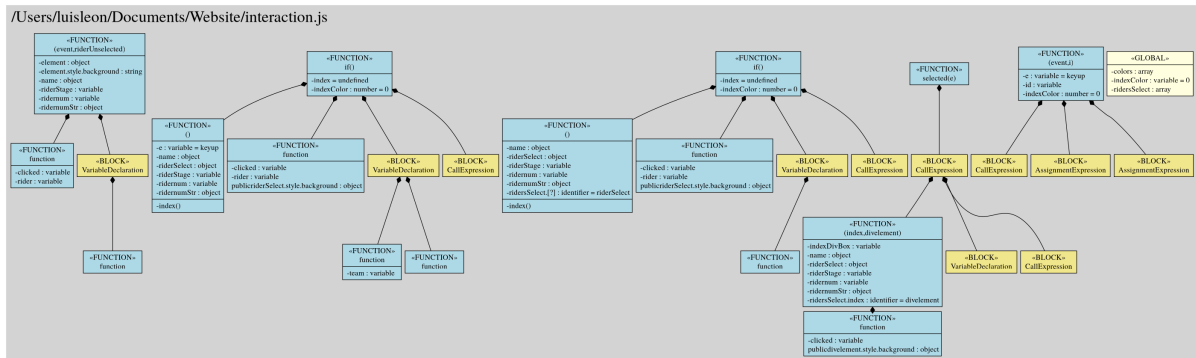


Figura 70: Diagrama de classes del fitxer interaction.js (1)

En la figura 71 es pot veure que s'han implementat un parell de mètodes que serveixen per mostrar o ocultar la informació temporal de les trajectòries, "showLabelInformation" i "hideLabelInformation" respectivament. Mostrarà aquesta informació sempre que s'hagi seleccionat la trajectòria i quan es deixi de seleccionar aquesta informació s'ocultarà.

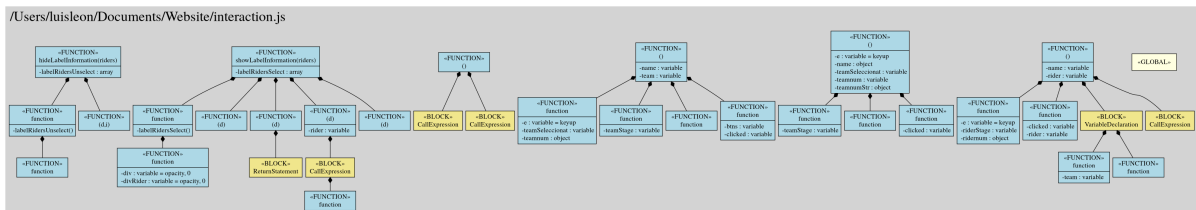


Figura 71: Diagrama de classes del fitxer interaction.js (2)

En la captura de la figura 72 es poden veure unes quantes funcions que s'executaran quan un usuari interactuï amb algun element de la pàgina, en comentarem un parell:

- `mostrarRiderInfo`: aquesta funció s'executarà quan es seleccioni una única trajectòria o ciclista. En la pàgina es mostrarà la informació detallada del ciclista que es selecciona o del ciclista al que li pertany la trajectòria seleccionada.
- `mostrarRiderLinea`: aquesta funció canviarà l'estil d'una trajectòria, que representa l'evolució d'un ciclista, mostrant també la informació temporal

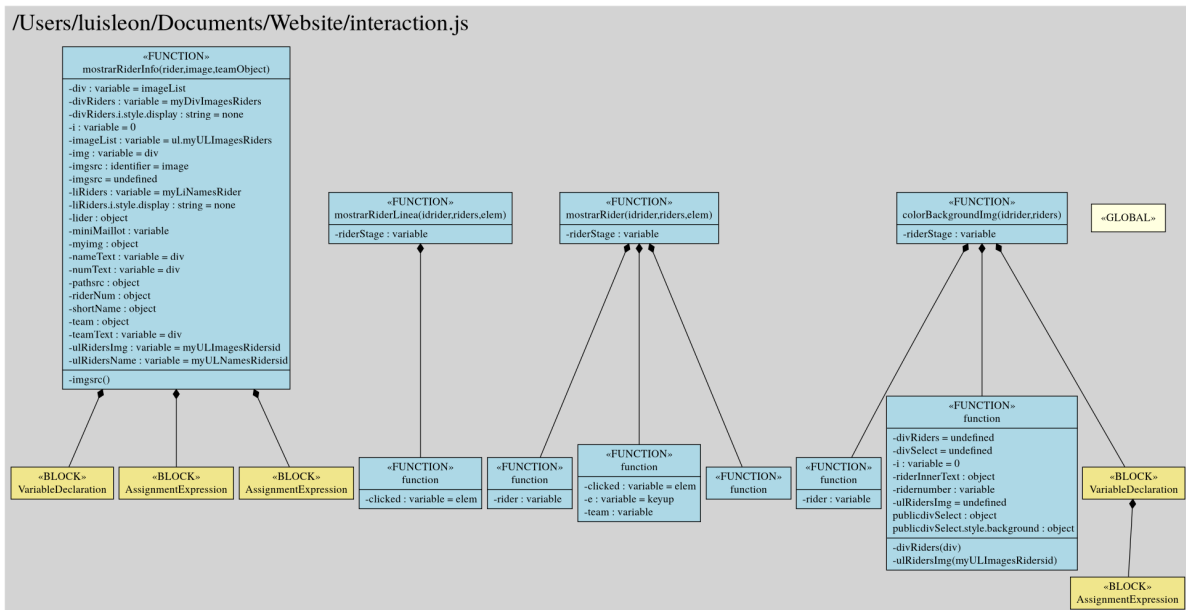


Figura 72: Diagrama de classes del fitxer interaction.js (3)

En la darrera captura, figura 73, la funció que més importància té és "search" aquesta funció representa el buscador de la barra lateral dreta. Aquesta funció filtra els ciclistes o equips que es mostraran segons el que s'entra per teclat.

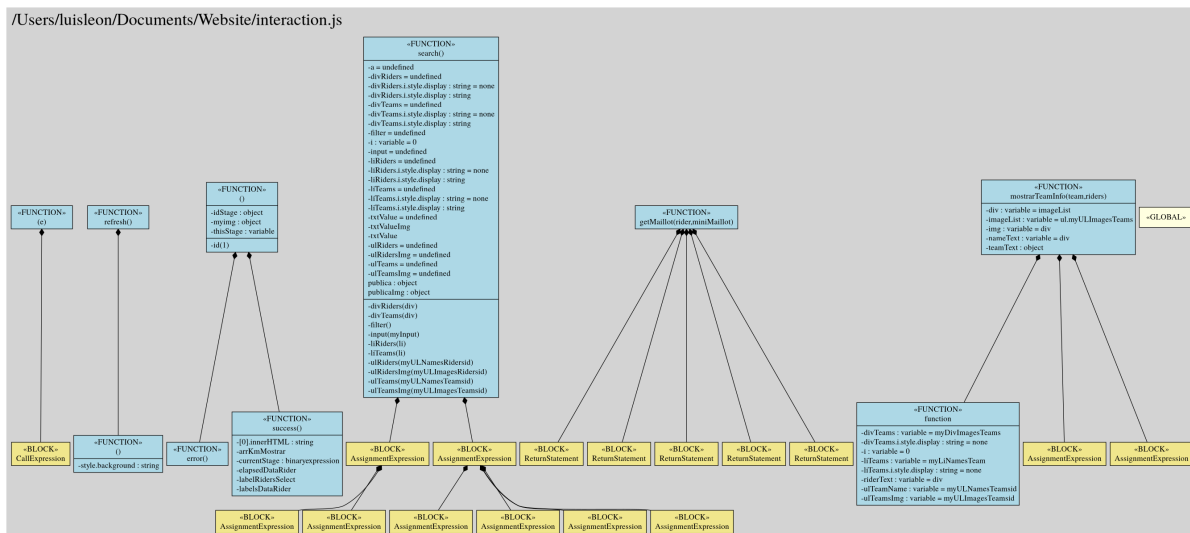


Figura 73: Diagrama de classes del fitxer interaction.js (4)

Un cop generades aquestes classes, he arribat a la conclusió un únic editor de text no es sufficient per fer aplicacions grans. Els exemples dels projectes que es poden trobar per internet són relativament més petit i es per això que amb un parell de fitxers és sufficient per executar-ho, però en el nostre la cas tenim

moltes més dades que hem de gestionar de la millor manera, és per això que considero que si pogués a tornar a reestructurar el codi de zero, segurament hagués utilitzat un altre software com per exemple WebStorm, un entorn de programació per codificar amb JavaScript. Utilitzar aquesta opció comportaria a recodificar tot el codi, crear diferents classes per els diferents elements de la web, probablement crear un altre tipus de servidor i crear totes les configuracions necessàries per executar-se en la web. De cares a un treball futur, seria molt recomanable que el primer pas a seguir sigués començar a fer aquesta migració a una altra plataforma.

## 8.4 Fitxer d'entrada

Com ja s'ha comentat prèviament les dades de cada etapa de la cursa es van obtenir gràcies a un membre del projecte, en Jose. Aquestes dades s'han guardat en diferents fitxers .csv (Comma-Separated Values): un fitxer per cada etapa de la cursa, un fitxer per la informació de cada ciclista i un fitxer per la informació dels equips. Per tal d'implementar els objectes de la pàgina i crear tots elements, prèviament s'havien de tractar totes les dades del fitxer, fer una relació entre els equips i els ciclistes, i sobretot estructurar i guardar totes les dades que es necessitàvem.

En els fitxers originals de les etapes es donava més informació de la que es necessitava per crear la pàgina, és per això que en l'exemple de la figura 74 només es mostren les columnes que realment es necessiten. En la figura 75 es pot observar quin és el fitxer original, obert amb l'excel, les columnes de color blau són les dades que necessitem guardar, la resta en aquest projecte no s'han tingut que fer servir.

```

1 //Format input etapes
2 numeroEtapa,OrigenEtapa,DestiEtapa,KM_Totals_Etapa
3 Rider_name,Rider_number,Team,Leader,kmX_Position,kmX_Elapsed_time,kmX_GC_prov_position,kmX_GC_prov_delay,kmXX_Position,kmXX_Elapsed_time,kmXX_GC_prov_posit
ion,kmXX_GC_prov_delay,kmXXX_Position,kmXXX_Elapsed_time,kmXXX_GC_prov_position,kmXXX_GC_prov_delay...
4 NOM_R,NUM_R,NOM_T,si/no,POS_X,TempsET_X,POS_PROV_X,DELAY_PROV_X,POS_XX,TempsET_XX,POS_PROV_XX,DELAY_PROV_XX,POS_XXX,TempsET_XXX,POS_PROV_XXX,DELAY_PROV_XXX
5 NOM_R,NUM_R,NOM_T,si/no,POS_X,TempsET_X,POS_PROV_X,DELAY_PROV_X,POS_XX,TempsET_XX,POS_PROV_XX,DELAY_PROV_XX,POS_XXX,TempsET_XXX,POS_PROV_XXX,DELAY_PROV_XXX
6 NOM_R,NUM_R,NOM_T,si/no,POS_X,TempsET_X,POS_PROV_X,DELAY_PROV_X,POS_XX,TempsET_XX,POS_PROV_XX,DELAY_PROV_XX,POS_XXX,TempsET_XXX,POS_PROV_XXX,DELAY_PROV_XXX
7 NOM_R,NUM_R,NOM_T,si/no,POS_X,TempsET_X,POS_PROV_X,DELAY_PROV_X,POS_XX,TempsET_XX,POS_PROV_XX,DELAY_PROV_XX,POS_XXX,TempsET_XXX,POS_PROV_XXX,DELAY_PROV_XXX
8 ....
9
10 //Exemple de l'input d'etapes
11 13,Bourg d'Oisans,Valence,"169,5"
12 Rider_name,Rider_number,Team,Leader,km169.5_Position,km169.5_Elapsed_time,km169.5_GC_prov_position,km169.5_GC_prov_delay,km160_Position,km160_Elapsed_time,
km160_GC_prov_position,km160_GC_prov_delay,km150_Position,km150_Elapsed_time,km150_GC_prov_position,km150_GC_prov_delay...
13 C. FROOME,1,SKY,no,1,+0'00'',2,0:01:39,126,+0'58'',2,0:01:39,102,+1'45'',2,0:01:39...
14 E. BERNAL GÓMEZ,2,SKY,no,2,+0'00'',19,0:21:22,81,+0'58'',18,0:21:22,65,+1'45'',18,0:21:22...
15 J. CASTROVIEJO,3,SKY,no,3,+0'00'',103,1:44:35,41,+0'58'',102,1:44:35,66,+1'45'',102,1:44:35...
16 M. KWIATKOWSKI,4,SKY,no,4,+0'00'',59,1:18:59,24,+0'58'',58,1:18:59,12,+1'45'',58,1:18:59...
17 ...
18
19 //Format input equips
20 Team_Num,Team_id,Team_Name,Logo,Shirt,Icon_Shirt
21 NUM_T,ID_T,NOM_T,LOGO_NAME,SHIRT_NAME,ICON_SHIRT_NAME
22 NUM_T,ID_T,NOM_T,LOGO_NAME,SHIRT_NAME,ICON_SHIRT_NAME
23 NUM_T,ID_T,NOM_T,LOGO_NAME,SHIRT_NAME,ICON_SHIRT_NAME
24 ...
25
26 //Exemple de l'input d'equips
27 Team_Num,Team_id,Team_Name,Logo,Shirt,Icon_Shirt
28 0,SKY,TEAM SKY ,Sky_logo.png,Sky_shirt.png,Sky_shirt_mini.png
29 1,EFD,TEAM EF EDUCATION FIRST - DRAPAC P/B CANNONDALE ,Education_First_logo.png,Education_First_shirt.png,Education_First_shirt_mini.png
30 2,ALH,AG2R LA MONDIALE ,AG2R_logo.png,AG2R_shirt.png,AG2R_shirt_mini.png
31 3,SUN,TEAM SUNWEB ,Sunweb_logo.png,Sunweb_shirt.png,Sunweb_shirt_mini.png
32 ....
33
34
35 //Format input equips
36 Rider_Num,Name,Short_Name,Team_Num,Photo
37 NUM_R,NOM_R,SHORT_NOM_R,NUM_T,PHOTO
38 NUM_R,NOM_R,SHORT_NOM_R,NUM_T,PHOTO
39 NUM_R,NOM_R,SHORT_NOM_R,NUM_T,PHOTO
40 ...
41
42 //Exemple de l'input d'equips
43 1,CHRIS FROOME ,C. FROOME,0,001.png
44 2,EGAN ARLEY BERNAL GOMEZ ,E. BERNAL GOMEZ,0,002.png
45 3,JONATHAN CASTROVIEJO ,J. CASTROVIEJO,0,003.png
46 4,MICHAL KWIATKOWSKI ,M. KWIATKOWSKI,0,004.png
47 ...

```

Figura 74: Format d'entrada per els diferents fitxers

20 July 2018 profile13.svg																			
Rider_name	Rider_numbr	Team	Leader	GC_at_start	GC_Total_ric	GC_Gap_at	Finish_Time	km169.5_Gr	km169.5_Position	km169.5_Elapsed_time	km169.5_Ga	km169.5_Avi	km169.5_Cu	km169.5_Gr	km169.5_GC_prov_position	km169.5_GC_prov_delay	km169.5_Gi	km160_Grou	km160_Position
C. FROOME	1	SKY	no	2 49h 26' 22"	0:01:39	03h 45' 55"	Main group		1	+0:00"	+0:00"	0 km/h	0 km/h	0%	2	0:01:39	49:26:22	Main group	126
E. BERNAL	2	SKY	no	19 49h 46' 05"	0:21:22	03h 48' 03"	Main group		2	+0:00"	+0:00"	0 km/h	0 km/h	0%	15	0:21:22	49:46:05	Main group	81
J. CASTROVIE	3	SKY	no	103 51h 09' 18"	1:44:35	03h 47' 51"	Main group		3	+0:00"	+0:00"	0 km/h	0 km/h	0%	103	1:44:35	51:09:18	Main group	41
M. KWATKQ	4	SKY	no	59 50h 43' 42"	1:18:59	03h 48' 03"	Main group		4	+0:00"	+0:00"	0 km/h	0 km/h	0%	59	1:18:59	50:43:42	Main group	24
G. MOSCON	5	SKY	no	114 51h 16' 13"	1:51:30	03h 48' 03"	Main group		5	+0:00"	+0:00"	0 km/h	0 km/h	0%	114	1:51:30	51:16:13	Main group	22
W. POELS	6	SKY	no	60 50h 45' 58"	1:21:15	03h 48' 03"	Main group		6	+0:00"	+0:00"	0 km/h	0 km/h	0%	60	1:21:15	50:45:58	Main group	11
L. BOWE	7	SKY	no	130 51h 27' 03"	2:02:20	03h 48' 03"	Main group		7	+0:00"	+0:00"	0 km/h	0 km/h	0%	130	2:02:20	51:27:03	Main group	15
G. THOMAS	8	SKY	GC	1 49h 24' 43"	0:00:00	03h 45' 55"	Main group		8	+0:00"	+0:00"	0 km/h	0 km/h	0%	1	0:00:00	49:24:43	Main group	114
S. CLARKE	12	EFD	no	113 51h 14' 51"	1:50:08	03h 47' 51"	Main group		9	+0:00"	+0:00"	0 km/h	0 km/h	0%	113	1:50:08	51:14:51	Main group	39
L. CRADDOCK	13	EFD	no	153 51h 58' 34"	2:33:51	03h 48' 40"	Main group		10	+0:00"	+0:00"	0 km/h	0 km/h	0%	153	2:33:51	51:58:34	Main group	94

Figura 75: Exemple fitxer d'entrada de les etapes



## 9 Implementació i proves

En aquesta secció es comentaran els conceptes i els successos més rellevants que han aparegut durant la implementació del projecte. Primer de tot es comentarà una sèrie d'implementacions que s'ha dut a terme. Més endavant es comentaran diferents proves del funcionament de la pàgina web, juntament amb les respectives captures d'imatges.

### 9.1 Implementacions

#### 9.1.1 Generar els grups de ciclistes

Un cop s'han llegit les dades dels fitxers d'entrada i s'han guardat correctament, és el moment de buscar quins són els grups que es formen en cada punt quilomètric. Per fer això es mostra a continuació el mètode en pseudocodi que bàsicament el que fa és un recorregut per cada un dels punts quilomètrics creats, ordenar pel valor de les coordenades  $y$  dels punts (on cada punt representa un ciclista), seguidament un cop s'han ordenat s'ha de buscar quins grups es van formant (explicació de com es formen els grups explicat en l'apartat 5). El codi complet es pot trobar en el punt 14.1 de l'annex.

```

1 // Obtenir els grups
2
3 GrupsCiclistes = []
4 PER cada KM que hi ha en l'etapa
5
6     GrupsCiclistesKM = []
7
8     PuntsCiclistesKM = []
9
10    Afegir tots els punts dels ciclistes del KM actual a PuntsCiclistesKM
11    Ordenar per "y" els punts de PuntsCiclistesKM
12
13    PER cada un dels punts de PuntsCiclistesKM
14
15        grupN = []
16        SI es pot fer grup de ciclistes LLAVORS
17
18            Afegir el puntActual a grupN
19            MENTRE el valor "y" del punts següents siguin consecutius FER
20
21                Afegir el punt a grupN
22                Actualitzar index de PuntsCiclistesKM
23
24            FIMENTRE
25
26            Afegir grupN a GrupsClistesKM
27
28        ALTRAMENT // es tractar d'un ciclista que no forma grup
29
30            Afegir el puntActual9 a grupN
31            Afegir grupN a GrupsClistesKM
32        FSI
33    FIPER
34
35    Afegir GrupsClistesKM a GrupsCiclistes
36
37 FIPER

```

Figura 76: Pseudocodi de generar els grups de ciclistes

### 9.1.2 Generació dels polígons per representar els grups

Per representar l'evolució dels grups dels ciclistes es generen un seguit de polígons. Per fer-ho s'ha utilitzat un algorisme que fa un recorregut per tots els grups generats en els punts quilomètrics, i per cada un dels grups comprova les diferents coordenades cartesianes de la trajectòria. En la figura 77 és pot observar un pseudocodi de com s'ha implementat. Cal remarcar que és molt important la línia número 7, on es declara quin serà els conjunts de punts que es van tractant. S'han de comprovar aquests punts ja que és possible que s'hagi creat un polígon on aquesta punt ja hi formi part. Si no es comprovés, es crearien molts més polígons dels que realment es necessiten, i a més a més la pàgina al iniciar-se aniria molt lent, perquè hauria de fer molts càlculs innecessàries i generaria més polígons dels que realment en necessitem. Per veure el codi complet de l'algorisme es pot accedir al punt 14.2 de l'annex.

```

1 // Obtenir poligons
2
3 Poligons = []
4 PER cada KM que hi ha en l'etapa
5
6   G_KM_X = Grups de ciclista del KM actual
7   PuntsTractats = [] *
8
9   PER cada grup de ciclistes de G_KM_X
10
11     G_X = Grup de ciclista actual
12     PER cada ciclista que hi ha a G_X
13
14       P1, P2, P3, P4 = Inicialitzar punts del poligon
15       SI els punts del ciclista actual no existeixen a PuntsTractats LLAVORS
16
17         G_SEG = Trobar el grup de ciclistes al que pertenyarà el ciclista actual en el KM següent
18         PER cada un dels ciclistes restants del grup actual
19
20           G_SEG_CANDIDAT = Buscar el grup de ciclistes al que pertenyarà el ciclista en el KM següent
21           SI G_SEG = G_SEG_CANDIDAT LLAVORS
22
23             Afegir el punt del ciclista a PuntsTractats
24             Tractar i actualitzar els punts P2, P3, P4
25
26             FSI
27           FPER
28
29       Tractar si el ciclista no te grup en el KM següent
30       Afegir [P1, P2, P3, P4] a Poligons
31     FSI
32   FPER
33 FIPER

```

Figura 77: Pseudocodi d'obtenir els polígons d'una etapa

### 9.1.3 Actualitzar posicions dels elements al fer zoom

La configuració del d3-zoom [19] és molt àmplia i la veritat és que si un no està acostumat al JavaScript, ni a la programació en entorn web, pot ser bastant farragós començar a utilitzar aquesta llibreria, com va ser en el meu cas. És per això que a part de llegir la documentació i entendre els diferents paràmetres que s'apliquen, el que em va ajudar sobretot va ser mirar exemples de projectes que hi havien per la web. Primer de tot vaig haver de analitzar els diferents exemples i poder modificar els paràmetres dels diferents mètodes, per veure com afecta en la implementació. Amb l'experiència que anava agafant i a base de

prova i error al final vaig aconseguir aplicar el zoom en el nostre projecte. Ara que tinc els coneixements si pogués generalitzar-lo ho dividiria en tres passos: definir la funció de zoom amb els paràmetres que t'interessen, escollir en quin element vols que s'apliqui zoom i actualitzar tots aquells elements on la posició sigui relativa a l'element esmentat abans.

1. **Definir funció de zoom:** Es defineix quin és el comportament que hi haurà en el moment que es faci zoom. Com s'ha comentat anteriorment, la possible configuració de la funcionalitat de zoom és molt gran, però pel nostre projecte hi aplicarem els següents comportament:

- *zoom.scaleExtent*([*k0*, *k1*]): Especifiquem quan de rang d'escala es podrà aplicar. On *k0* serà el valor mínim que es podrà fer escala i *k1* el valor màxim. Si aquests valors no s'apliquessin els valors predefinitos són 0 i infinit. En el nostra pàgina web s'ha decidit aplicar un rang de [1,20], ja que són valors que per la nostra finalitat són acceptables.
- *zoom.translateExtent*([*x0*, *y0*], [*x1*, *y1*]): *x0* i *y0* són els valors de la cantonada de a dalt a l'esquerra. *x1* i *y1* són els valors de la cantonada de a baix a la dreta. En el nostre cas els valors són: [[0, 0], [width, height]]. La primera tuple és 0, 0 ja que recordem que per codificar i guardar les punts de les trajectòries, el nostre punt d'origen es a dalt a la dreta. Podem dir que és el límit on ha d'aplicar el zoom. La segona tuple són l'amplitud i l'altura de l'àrea del nostre gràfic. Si aquests valors no s'apliquessin els valors per defecte serien [-infinit, -infinit], [+infinit, +infinit]
- *zoom.extent*([*x0*, *y0*], [*x1*, *y1*]): S'especifica els valors dels límits de la finestra (que en el nostre cas seria la gràfica) que es pot aplicar zoom. El valor per defecte si no s'aplica la configuració, són els valors del seu "pare". I en el nostre projecte aquest element "pare" és un objecte SVG que engloba tots elements de la gràfica (línies verticals, etiquetes, trajectòries..). On la seva posició i dimensió han sigut especificats a partir d'un *viewBox* i aquests valors no són els que a nosaltres ens interessa, ja que *margins* i *padding*s farien que la gràfica no es visualitzes correctament. Els valors que té l'*extent* són els mateixos que *translateExtent*.
- *duration*: S'especifica la duració en mili-segons, que trigarà la transició de fer zoom. Hem escollit 750 mili-segons, quan per defecte està a 250.
- *.on("zoom", functionName)*: Aquest comportament que s'ha definit comportarà un *trigger* a executar una funció, en el nostre cas hem escollit de nom zoomed. On més endavant hi veurem que s'ha declarat.

2. **Escollir l'element on aplicar el zoom:** Un cop s'ha definit quin comportament tindrà aplicar fer zoom, s'ha d'escollir el selector (o element) que en el nostre cas serà tota l'àrea de la gràfica. Un exemple senzill d'això que s'ha explicat seria: `d3.select(#stageId).call(zoom)`

3. **Actualitzar els altres valors de la pàgina que siguin necessaris:** En aquest últim pas s'ha de tenir en compte quins són tots els elements que formen part de la gràfica i com s'ha d'actualitzar cada una de les posicions.

(a) Trobar la nova escala que ha generat el zoom:

- *d3.event.transform.rescaleX(escala\_X)*: Aquest mètode retorna quina es la nova escala en l'eix de les x, que ha aplicat el zoom.
- *d3.event.transform.rescaleY(escala\_Y)*: Aquest mètode retorna quina es la nova escala en l'eix de les y, que ha aplicat el zoom.

(b) Un cop es té els nous valors de l'escala s'ha d'actualitzar tots els elements que es veuen afectats pel comportament de la gràfica que són els següents:

- Eixos de les x i de les y, de la gràfica.
- Col·locació dels diferents línies verticals que representen els punts quilomètrics.
- Les trajectòries de cada un dels ciclistes.
- Els *tooltips* que es situen en les trajectòries.
- Polígons que representen la generació dels grups de ciclistes durant l'etapa.
- Etiquetes que mostren la diferència de temps que hi hagut entre els grups d'un mateix punt quilomètric.
- Zona ombrejada del perfil de l'etapa, que no forma part de la gràfica però igualment s'ha d'actualitzar la mida

La funció que s'ha adjunta a continuació és el pseudocodi per actualitzar tots els elements que apareixen en la gràfica un cop s'aplica zoom. Els atributs dels objectes de la gràfica són actualitzats gràcies a la llibreria D3. Però hi han 3 elements que hem tingut que gestionar nosaltres per decidir exactament com s'haurien de mostrar. Aquests elements són els eixos quilomètrics, les etiquetes que es mostren entre els grups d'un mateix punt quilomètric i el perfil de l'etapa. Per veure el codi complet es pot anar a la secció 14.3 de l'annex.

```
1 // Comportament al fer zoom
2
3 newScale = Crear la nova escala de les coordenades "x", "y"
4
5 Assignar newScale com l'escala actual
6
7 valueScale = Trobar el valor numèric del zoom que s'aplica
8
9 puntsKM_mostrar = Trobar quins són els nous punts quilomètrics a mostrar a partir de valueScale
10
11 reescalar l'eix de les "x"
12 reescalar l'eix de les "y"
13
14 Tornar a dibuixar la trajectòria a partir de newScale
15
16 Tornar a posicionar els punts que hi han sobre les trajectòries a partir de newScale
17
18 Tornar a dibuixar els polígons a partir de newScale
19
20 Dibuixar les etiquetes entre grups que pertoquin
21
22 Redimensionar l'ombrejat del perfil de l'etapa a partir de newScale
```

Figura 78: Pseudocodi del comportament del zoom

#### 9.1.4 Trobar els punts quilomètrics a mostrar

Com s'ha explicat amb anterioritat un dels objectius de disseny que ha tingut el projecte és mostrar un nombre màxim de línies verticals en tot moment. En el nostre cas vam decidir que 4 línies verticals eren els idonis per no donar sobreinformació a l'usuari. Aquest valor l'hem anomenat "nAxis". Una altra variable que s'ha d'explicar és "distanciaEntreKM" aquest valor representa quina és la distància que hi ha entre dos punts quilomètrics (recordem que és en aquests punts on s'han recollit les dades del ciclista). Aquesta variable pot ser un valor o un altre, depenent de l'etapa que s'estigui mostrant, ja que aquesta separació no sempre es constant en totes les etapes.

Aquest algorisme l'hem tingut que adaptar un parell de vegades ja que al principi no teníem en compte que la separació que hi ha entre punts quilomètrics no era constant, però al final, amb l'ajuda de la meva tutora, l'algorisme trobava el punts quilomètrics correctes. A continuació es mostra un pseudocodi, si es vol veure el codi original està en el punt 14.4 de l'annex.

```

1 // Trobar punts quilomètrics a mostrar
2
3 nAxis = Definir el nombre de punts quilomètrics màxims que es mostraran
4 rangVisio = Trobar quin és el nou rang que es mostra (amb km)
5 distanciaEntreKM = Definir la distància que hi ha entre punts quilomètrics
6 columnsKM = Tots el punts quilomètrics
7
8 arrKMmostrar = []
9
10 SI rangVisio > nAxis * distanciaEntreKM LLAVORS
11
12     PER i = 1 a nAxis
13         arrKMmostrar = Afegir el punt quilomètric pertinenent de columnsKM
14     FIPER
15
16 ALTRAMENT
17
18     indexPrimerKMvist = Buscar posicio del primer km vist
19     indexUltimKMvist = Buscar posicio de l'últim km vist
20
21     SI indexUltimKMvist - indexPrimerKMvist < nAxis LLAVORS
22
23         arrKMmostrar = Afegir tots el punts quilomètrics de columnsKM, que estiguin entre indexPrimerKMvist i indexUltimKMvist
24
25     ALTRAMENT
26
27         arrKMmostrarSuggerits = Buscar els punts km que s'haurien de mostrar
28
29         arrKMmostrar = Afegir nAxis punts quilomètrics a partir de arrKMmostrarSuggerits i columnsKM
30     FSI
31 FSI

```

Figura 79: Pseudocodi trobar els punts quilomètrics a mostrar

### 9.1.5 Visualització en diferents navegadors

Durant la implementació del projecte s'ha estat utilitzant el navegador de Firefox per fer les diferents proves d'execució. No va ser fins quasi arribant al final del projecte quan es va començar a testejar l'aplicació en diferents navegadors, com per exemple Chrome o Safari. La primera execució en aquests navegadors va ser que no es mostrava la gràfica generada pel D3.

Es va començar a fer una cerca d'aquest error i la primera hipòtesi que vam tenir va ser que generar elements SVG amb els atributs de la mida (*height*, *width*) amb unitat de tan per cent no era compatible amb altres navegadors que no fossin Firefox. Aquesta hipòtesi va sorgir perquè modificant aquests atributs i aplicant un valor de píxels, llavors sí que funcionava en tots els navegadors. Finalment aquesta idea es va descartar gràcies a un anàlisi més profund del codi. Ens vam donar compte que aquest error es podria arreglar bàsicament modificant unes línies de codi de l'HTML generat pel D3 fent que l'estructura del gràfic dins de la web sigués més senzilla. Amb aquest nou canvi ja es podia deixar les unitats en tant per cent, que és com havien d'estar. A continuació es mostra unes captures de l'aplicació executant-se en diferents navegadors (figures 80–82).

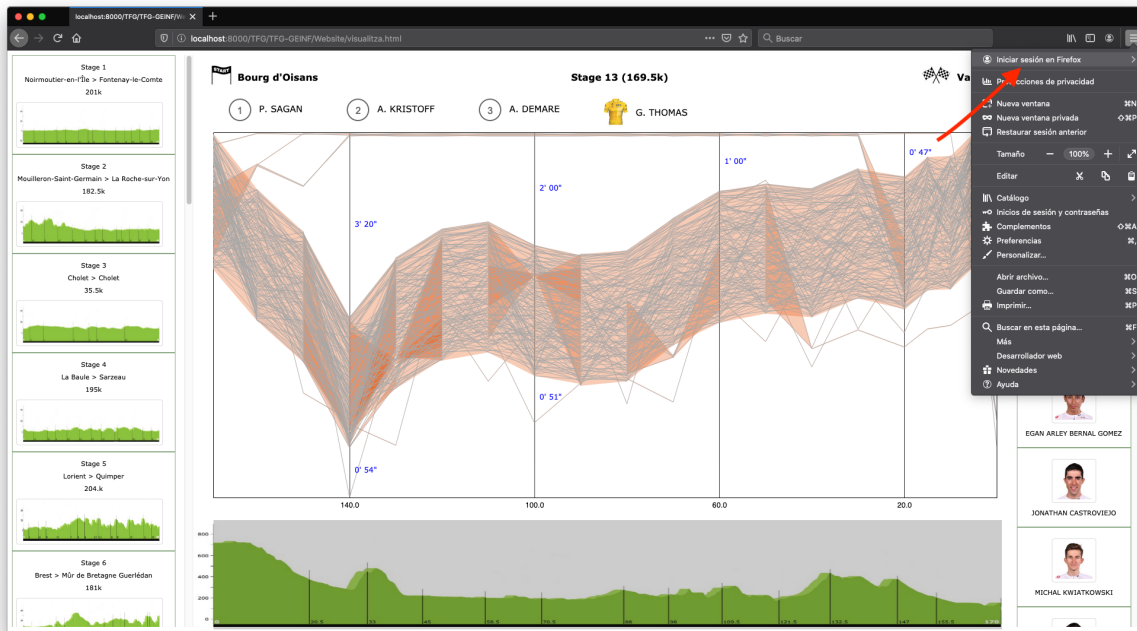


Figura 80: Visualització de l'aplicació en Firefox

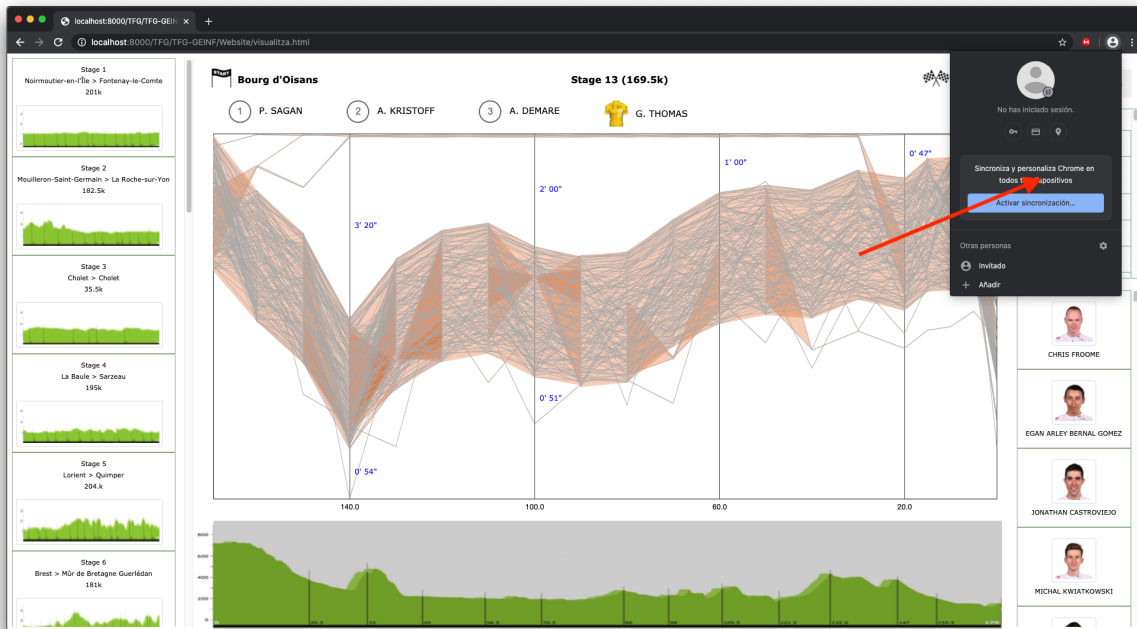


Figura 81: Visualització de l'aplicació en Chrome

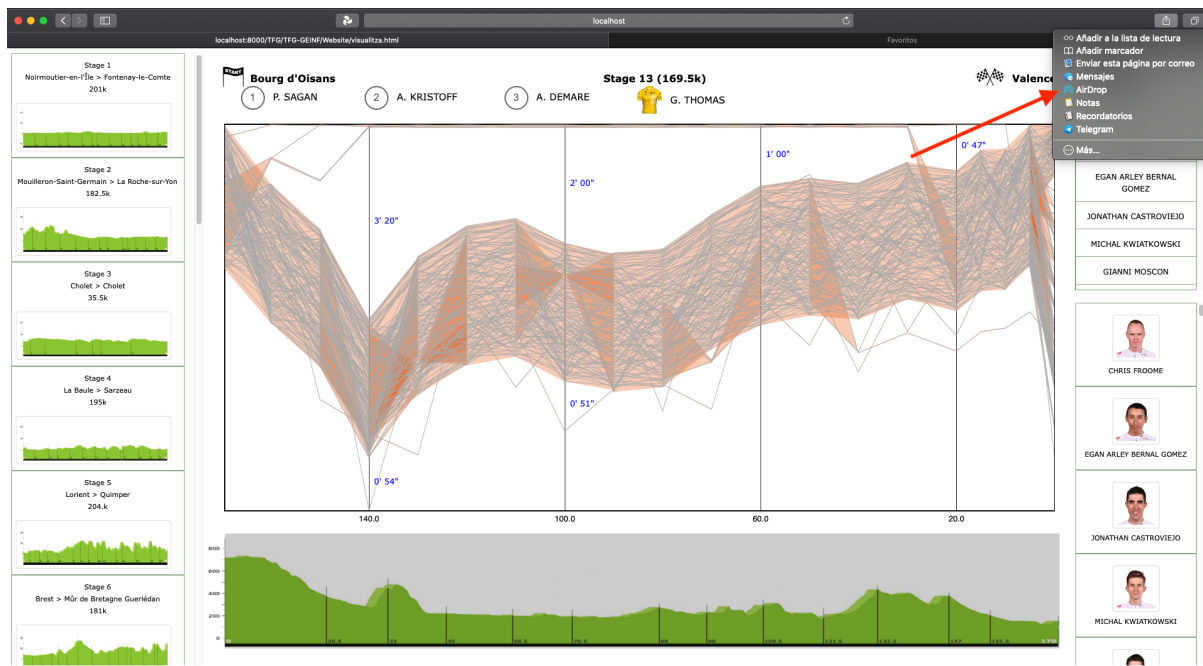


Figura 82: Visualització de l'aplicació en Safari

## 9.2 Proves

En aquesta secció es mostraran certes captures per verificar que la implementació del projecte funciona correctament.

### 9.2.1 Obtenir dades del servidor

En la figura 83 és mostren els logs que s'executen en el moment de executar l'aplicació, concretament en aquest exemple es mostra les crides GET de les imatges dels ciclistes i de les diferents imatges dels mallots.



```

127.0.0.1 - - [06/Jun/2020 21:12:17] "GET /TFG/TFG-GEINF/Website/v2/imgs/207.png HTTP/1.1" 200 -
127.0.0.1 - - [06/Jun/2020 21:12:17] "GET /TFG/TFG-GEINF/Website/v2/imgs/208.png HTTP/1.1" 200 -
127.0.0.1 - - [06/Jun/2020 21:12:17] "GET /TFG/TFG-GEINF/Website/v2/imgs/211.png HTTP/1.1" 200 -
127.0.0.1 - - [06/Jun/2020 21:12:17] "GET /TFG/TFG-GEINF/Website/v2/imgs/213.png HTTP/1.1" 200 -
127.0.0.1 - - [06/Jun/2020 21:12:17] "GET /TFG/TFG-GEINF/Website/v2/imgs/212.png HTTP/1.1" 200 -
127.0.0.1 - - [06/Jun/2020 21:12:17] "GET /TFG/TFG-GEINF/Website/v2/imgs/214.png HTTP/1.1" 200 -
127.0.0.1 - - [06/Jun/2020 21:12:17] "GET /TFG/TFG-GEINF/Website/v2/imgs/215.png HTTP/1.1" 200 -
127.0.0.1 - - [06/Jun/2020 21:12:17] "GET /TFG/TFG-GEINF/Website/v2/imgs/216.png HTTP/1.1" 200 -
127.0.0.1 - - [06/Jun/2020 21:12:17] "GET /TFG/TFG-GEINF/Website/v2/imgs/218.png HTTP/1.1" 200 -
127.0.0.1 - - [06/Jun/2020 21:12:17] "GET /TFG/TFG-GEINF/Website/v2/imgs/217.png HTTP/1.1" 200 -
127.0.0.1 - - [06/Jun/2020 21:12:18] "GET /TFG/TFG-GEINF/Website/v2/imgs/AG2R_shirt.png HTTP/1.1" 200 -
127.0.0.1 - - [06/Jun/2020 21:12:18] "GET /TFG/TFG-GEINF/Website/v2/imgs/Astana_shirt.png HTTP/1.1" 200 -
127.0.0.1 - - [06/Jun/2020 21:12:18] "GET /TFG/TFG-GEINF/Website/v2/imgs/Bahrain_shirt.png HTTP/1.1" 200 -
127.0.0.1 - - [06/Jun/2020 21:12:18] "GET /TFG/TFG-GEINF/Website/v2/imgs/BMC_shirt.png HTTP/1.1" 200 -
127.0.0.1 - - [06/Jun/2020 21:12:18] "GET /TFG/TFG-GEINF/Website/v2/imgs/Bora_shirt.png HTTP/1.1" 200 -
127.0.0.1 - - [06/Jun/2020 21:12:18] "GET /TFG/TFG-GEINF/Website/v2/imgs/Cofidis_shirt.png HTTP/1.1" 200 -
127.0.0.1 - - [06/Jun/2020 21:12:18] "GET /TFG/TFG-GEINF/Website/v2/imgs/Dimension_shirt.png HTTP/1.1" 200 -
127.0.0.1 - - [06/Jun/2020 21:12:18] "GET /TFG/TFG-GEINF/Website/v2/imgs/Direct_Energie_shirt.png HTTP/1.1" 200 -
127.0.0.1 - - [06/Jun/2020 21:12:18] "GET /TFG/TFG-GEINF/Website/v2/imgs/Education_First_shirt.png HTTP/1.1" 200 -
127.0.0.1 - - [06/Jun/2020 21:12:18] "GET /TFG/TFG-GEINF/Website/v2/imgs/FDJ_shirt.png HTTP/1.1" 200 -
127.0.0.1 - - [06/Jun/2020 21:12:18] "GET /TFG/TFG-GEINF/Website/v2/imgs/Emirates_shirt.png HTTP/1.1" 200 -
127.0.0.1 - - [06/Jun/2020 21:12:18] "GET /TFG/TFG-GEINF/Website/v2/imgs/Fortuneo_shirt.png HTTP/1.1" 200 -
127.0.0.1 - - [06/Jun/2020 21:12:18] "GET /TFG/TFG-GEINF/Website/v2/imgs/Katusha_shirt.png HTTP/1.1" 200 -

```

Figura 83: Captura de la terminal quan obté les dades

## 9.2.2 Ordre d'execució dels fitxers del projecte

Com s'ha esmentat anteriorment s'ha de mantenir un ordre dels diferents fitxers JavaScript ja que hi han variables globals que es comparteixen en diferents fitxers. Però també s'ha de tenir en compte que hi ha un sol fitxer que s'encarrega de gestionar la interacció de l'usuari amb la pàgina web. És per això que ens hem d'assegurar de carregar totes les dades i tots els elements del DOM abans de començar a interactuar amb ells, ja que pot donar error. En el codi que es mostra a continuació podem veure com es van fent crides GET una darrere de l'altre, assegurant-nos de que la petició anterior ha anat bé, és llavors quan en les línies 24 i 25 es poden executar els fitxers `riders-graph.js` i `interaction.js`. Això ens assegura que els primers elements que es crearan seran: barra lateral esquerra, barra lateral dreta i perfil de l'etapa. Només llavors quan es poden llegir els altres dos fitxers.

```

1 $(document).ready(function() {
2     $.ajax({
3         type: "GET",
4         url: "./csv/Riders.csv",
5         dataType: "text",
6         success: function(data) {
7             processDataNames(data);
8             var folder = "./imgs/";
9             $.ajax({
10                type: "GET",
11                url: folder,
12                dataType: "text",
13                success: function (data) {
14                    imatges = data;

```

```

15     processDataImages(data, folder);
16     $.ajax({
17         type: "GET",
18         url: "../csv/Teams.csv",
19         dataType: "text",
20         success: function(data){
21             processDataTeams(data);
22             processDataImagesTeams(imatges, folder);
23             $(function () {
24                 $.getScript('riders-graph.js');
25                 $.getScript('interaction.js');
26             });
27         }
28     });
29 }
30 });
31 }
32 });
33 });

```

Listing 1: Peticions GET de les dades de cada fitxer

### 9.2.3 Mostrar mallot d'un ciclista líder

Com s'ha comentat en l'apartat 5.2.1 hi ha diferents tipus de líder en les etapes de la cursa. Aquests es diferencien pel mallot que porten. En la figura 84 es mostra la informació d'un ciclista un cop s'ha seleccionat. Aquest ciclista en concret és líder de muntanya en l'etapa actual, ho podem saber gràcies a la miniatura del mallot que es visualitza, té punts impresos, en canvi en la figura 85 es mostra que aquell ciclista és líder i per tant, la miniatura del mallot és el mateix que la de la foto de perfil.

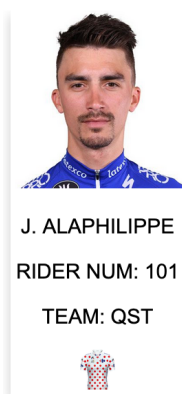


Figura 84: Ciclista líder

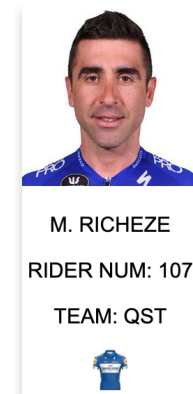


Figura 85: Ciclista no líder

### 9.2.4 Filtrar per nom de ciclista o equip

El parell d'imatges (figures 86 – 87) que es mostra a continuació s'observa quin és el comportament de la barra de cercador. A mesura que escrius per teclat el nom que t'interessa es van mostrant els ciclistes o els equips que coincideixen amb el nom escrit.

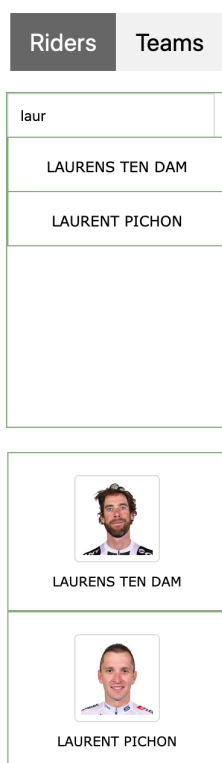


Figura 86: Filtrar ciclista per nom



Figura 87: Filtrar equip per nom

### 9.2.5 No hi han dades d'etapa o ciclista en l'etapa actual

Com s'ha comentat anteriorment, aquest projecte únicament disposa de les dades de 3 etapes, que són la número 11, 13 i 18. Encara que només tinguem aquestes dades, la pàgina web s'ha estructurat de forma que es pugui seleccionar qualsevol de les etapes que es mostren de la cursa. És per això que s'ha hagut d'implementar un alert per informar a l'usuari de que les dades que vol obtenir no existeixen. En la figura 88 en podem veure un exemple.

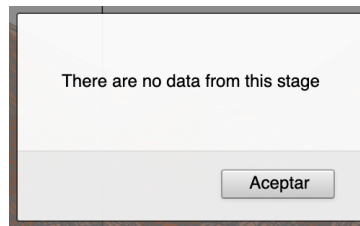


Figura 88: Alert no existeixen dades de l'etapa seleccionada

L'aplicació web està estructurada de manera que en la barra lateral de la dreta en tot moment hi hagi l'opció de seleccionar qualsevol dels ciclista de la cursa. Però és possible que en l'etapa actual que s'està mostrant aquell ciclista en concret no hi estiguin participant, és per això que també s'ha d'avisar a l'usuari cada cop que vulgui seleccionar aquest tipus de ciclistes. En la figura 89 es pot veure un exemple de l'alert que es mostra a l'usuari.

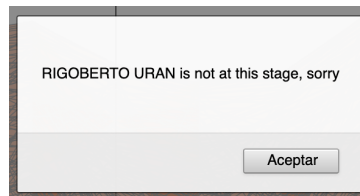


Figura 89: Alert no participa el ciclista en l'etapa actual

### 9.2.6 Mostrar punts quilomètrics

Un dels algorismes que s'han comentat anteriorment en l'apartat 9.1.4 és el de trobar els punts quilomètrics a mostrar en la gràfica. Ja s'ha comentat que la distància entre punts quilomètrics no sempre és constant, és per això que vam haver de dissenyar aquell algorisme, per trobar les línies que tocaven. Les dades obtingudes de l'etapa número 11 són les menys constants (en distància entre punts quilomètrics) de les que disposàvem, és per això que era un bon fitxer de prova per provar aquest algorisme. En les figures 90 i 91 es mostra el resultat de fer diferent quantitat de zoom en aquesta etapa.

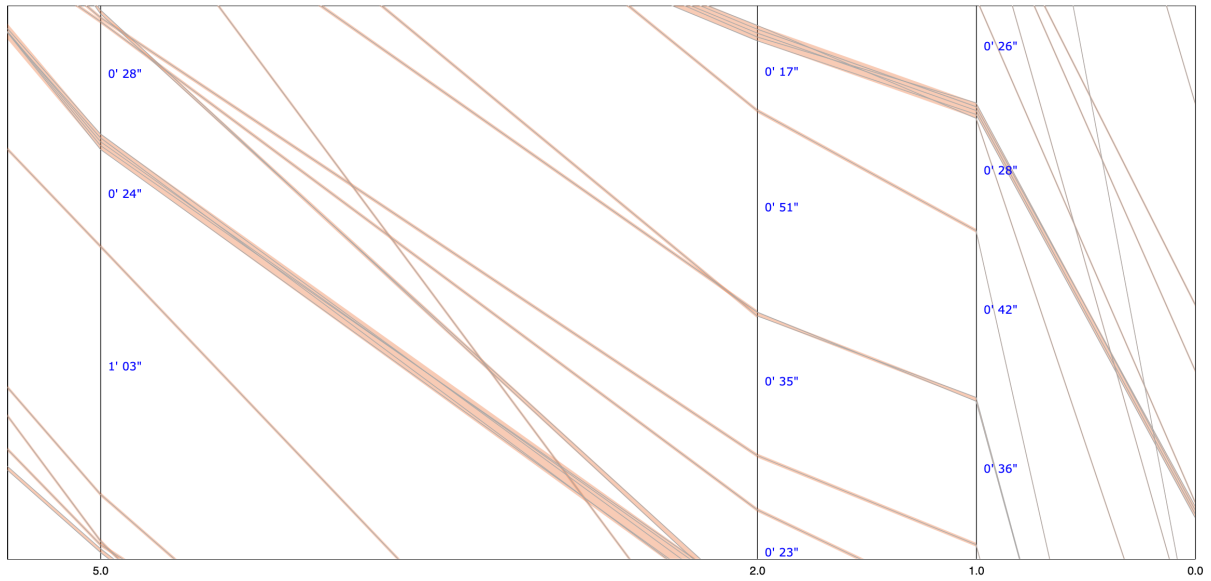


Figura 90: Mostrar punts quilomètrics amb diferent distància entre trams (1)

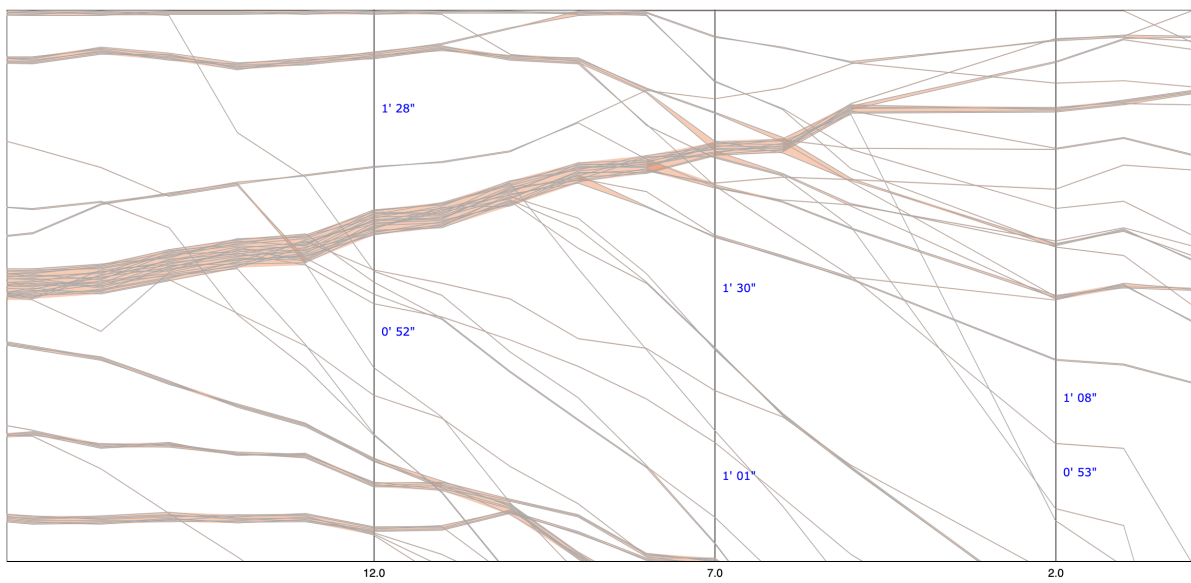


Figura 91: Mostrar punts quilomètrics amb diferent distància entre trams (2)

## 10 Implantació i resultats

En aquest apartat s'explicarà detalladament quins són els passos que s'ha de dur a terme per executar l'aplicació, si s'executen els passos correctament qualsevol usuari pot executar l'aplicació sempre i quan disposi de dades d'una cursa amb el format que s'ha esmentat en l'apartat 8.4. També es comentaran i s'adjuntaran algunes captures d'imatges que serviran per demostrar que els requeriments del projecte s'han assolit satisfactòriament.

### 10.1 Implantació d'un servidor local

Un cop sabem quina és la definició de servidor, esmentat en l'apartat 5.2.5, és el moment d'explicar com executar un servidor HTTP local.

1. Instal·lar Python. Si el dispositiu de treball és un sistema operatiu Linux o Mac OS X, el més probable és que ja estigui instal·lat. Si el sistema operatiu és Windows s'ha d'instal·lar anant a la pàgina web oficial.
  - Anar a [www.python.org](http://www.python.org)
  - Anar a la secció de Downloads i fer click sobre Python "3.xxx"
  - En la part superior de la pàgina escollir l'executable windows x86-64 i descarregar-lo
  - Executar l'instal·lador
  - En la primera pàgina de l'instal·lació assegurar-se de fer tick a "Add Python 3.xxx to PATH"(figura 92)
  - Fer clic a instal·lar
2. Obrir el terminal i entrar la comanda "python -V"per assegurar-nos de que s'ha instal·lat correctament.
3. Des de la terminal accedir en el directori on hi han els fitxers del projecte
4. Un cop en el PATH on hi han els fitxers del projecte executar la següent comanda:
  - `python -m http.server` (en Windows) (figura 93)
  - `python -m SimpleHTTPServer` (en Mac i Linux) (figura 94)
5. Per defecte, s'executarà el servidor local en el port 8000. Es pot accedir aquest servidor local accedit-hi amb un navegador web amb l'url localhost:8000
  - (a) Es possible que el port 8000 ja s'estigui executant per un altre servei. En aquest cas, es pot escollir un port en concret amb la comanda "python -m SimpleHTTPServer [port]"

Per veure els passos a seguir amb més detall es pot accedir a partir de la referència número [2]

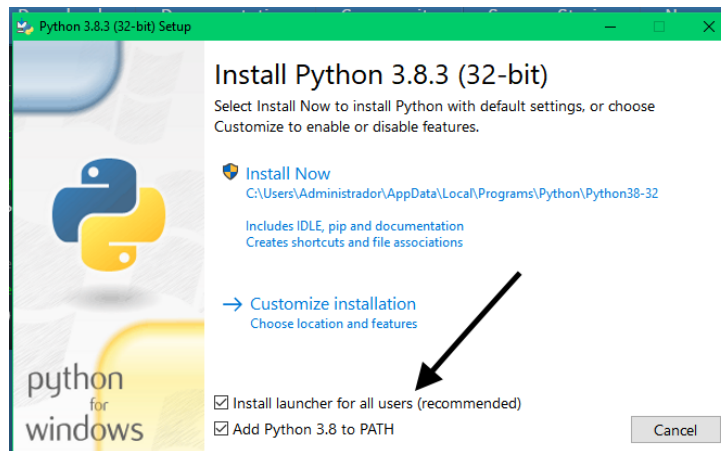


Figura 92: Exemple d'instal·lació de Python

```
Directorio de C:\website
07/06/2020 11:01 <DIR> .
07/06/2020 11:01 <DIR> ..
07/06/2020 11:00 <DIR> csv
07/06/2020 11:00 <DIR> imgs
07/06/2020 00:30      23.509 interaction.js
07/06/2020 00:46      24.713 riders-graph.js
06/06/2020 21:29       6.278 riders-info.js
07/06/2020 00:51       3.054 stages.js
07/06/2020 00:36      10.078 style.css
06/06/2020 21:28       4.002 visualitza.html
          6 archivos          71.634 bytes
          4 dirs 171.896.438.784 bytes libres

C:\website>python -m http.server
Servicing HTTP on :: port 8000 (http://[::]:8000/) ...
```

Figura 93: Exemple d'executar el servidor local amb Windows

```
luisleon$ python -m SimpleHTTPServer
Servicing HTTP on 0.0.0.0 port 8000 ...
```

Figura 94: Exemple d'executar el servidor local amb un Mac OS X

## 10.2 Lectura dels fitxers d'entrada

Un cop el servidor local s'està executant, l'últim pas és accedir al fitxer visualitza.html del projecte a partir de la ruta localhost:8000 que s'ha introduït en la URL del navegador. Un cop es carregui el fitxer visualitza.html es llegiran totes les dades de la cursa que estiguin guardades en la carpeta .csv del directori actual i es mostrarà la gràfica de la pàgina web. És per això molt important tenir guardades les dades dels fitxers d'entrada amb el format correcte.

## 10.3 Resultats dels objectius assolits

A continuació s'adjunten una sèrie de captures de pantalla on es mostrarà l'assoliment dels diferents requeriments que es van plantejar des de el principi del projecte.

### 10.3.1 Visualització de la pàgina

En la figura 95 que es mostra a continuació, es mostra quina és la primera visualització a l'iniciar la pàgina web si totes les dades s'ha llegit correctament.

A l'esquerra hi ha totes les etapes que es poden seleccionar, i com es pot veure en la figura l'etapa seleccionada queda ombrejada de color marró, a la dreta si pot veure informació de ciclistes i equips, juntament amb un botó que deixa escollir quins elements vols mostrar. En la part d'en mig de la pàgina veiem en tot moment origen, destí i distància que hi ha en aquesta etapa, juntament amb els diferents tipus de guanyadors. En la part d'en mig es forma tot el gràfic, representant les dades de l'etapa seleccionada. Es representa amb un estil de colors que va a joc amb el color verd (ja que el color verd és un dels colors que més predomina en tota la pàgina). En aquest mateix gràfic es visualitzen les trajectòries, els diferents polígons, algunes etiquetes entre els grups i exactament 4 línies verticals que representen els punts quilomètrics a mostrar. Per últim, a la part de sota es mostra l'imatge del desnivell d'aquesta etapa actual, juntament amb un element ombrejat a sobre.

En la part d'adalt de la web on hi veiem l'origen i el destí, al costat podem observar uns icones que representen la sortida i l'arribada respectivament, aquestes icones s'han afegit amb la finalitat de comprendre amb més facilitat quina és la direcció que fan els ciclistes durant l'etapa. Respecte al 1r, 2n i 3r finalistes de l'etapa juntament amb líder de la classificació general (mallot groc) s'ha dissenyat expressament perquè es mostrin alineats a l'esquerra, si ha sigut així es gràcies als experts que ens van dir que aquest alineació era la més convenient.

Respecte al perfil de l'etapa que es mostra en la part inferior de la pàgina s'ha d'esmentar que els



quilòmetres que es mostren no han de coincidir amb els quilòmetres del gràfic. Això és perquè els quilòmetres del gràfic són d'esquerra a dreta els quilòmetres que falten per acabar l'etapa. En canvi la representació del perfil de l'etapa està fet de manera que d'esquerra a dreta mostra el quilòmetres que s'han recorregut. Es per això que aquest número de quilòmetres no ha de coincidir el del gràfic amb el del perfil de l'etapa. Una altra de les raons és perquè la imatge del perfil de l'etapa s'ha obtingut de la web Pro Cycling i aquestes són les dades que hem obtingut[5].

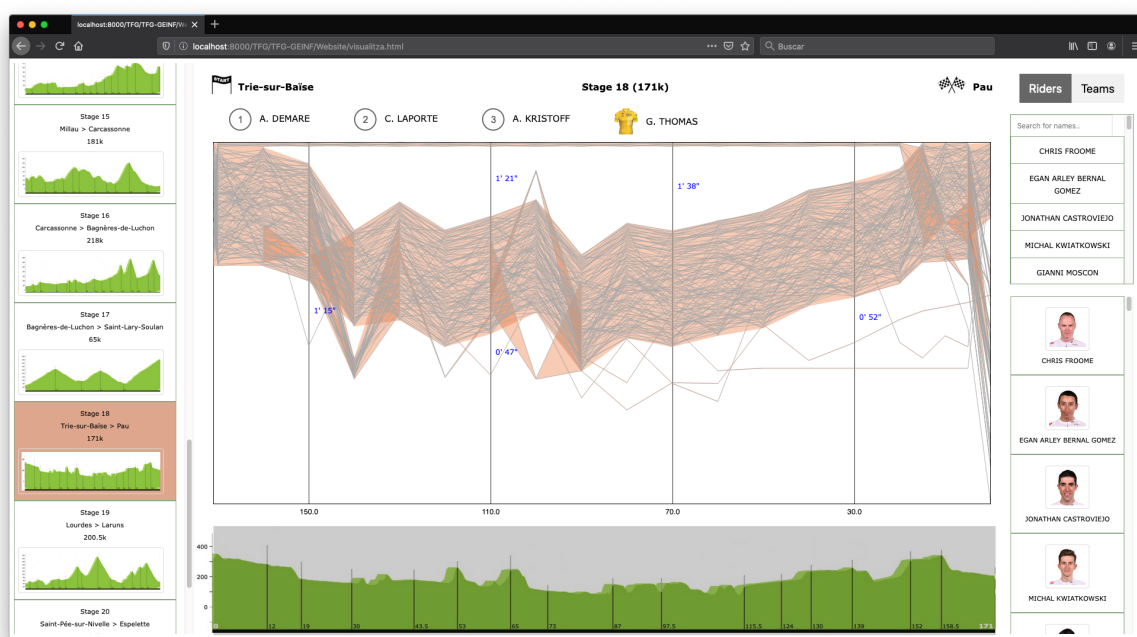


Figura 95: Visualització de la pàgina web

### 10.3.2 Visualització dels grups

En aquest apartat es comentarà breument les diferents visualitzacions de grups que pot haver-hi.

#### Grups que es divideixen

En la figura 96 es veu els tipus de grups que es divideixen d'un punt quilomètric a una altre. En la imatge es pot observar que en el punt quilomètric número 110 hi ha un grup format per la gran majoria de ciclistes, que en el tram següent es dividirà en 3 grups. El primer grup el formaran pocs ciclistes, si fèssim zoom segurament veuríem que deuen haver-hi 2, el segon grup el formen gran part dels ciclistes, format en el centre, i per últim el tercer grup es situa abaix, probablement també només per dos ciclistes.

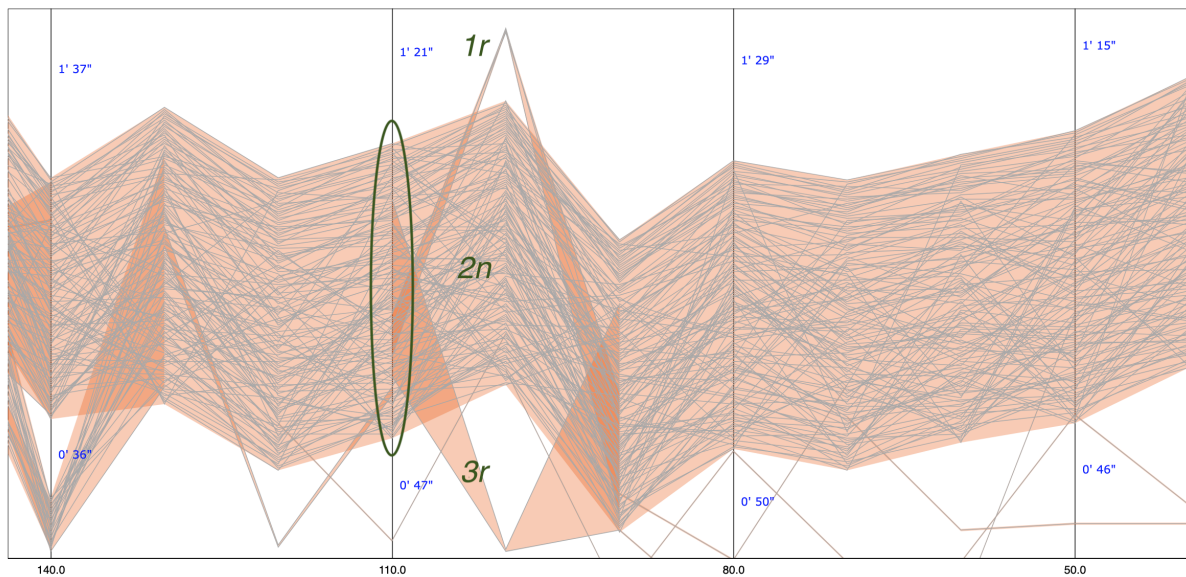


Figura 96: Visualització dels grups que es divideixen

### Grups que es fusionen

En la figura 97 veiem com en el punt quilomètric número 100, hi han tres grups (1r, 2n i 3r), aquests 3 grups en el següent punt quilomètric podem veure que es fusionen per formar un sol grup. Si la línia quilomètrica del punt de control on es fusionen no es visualitza, és perquè la pàgina web està dissenyada per mostrar les línies verticals que millor convinguin.

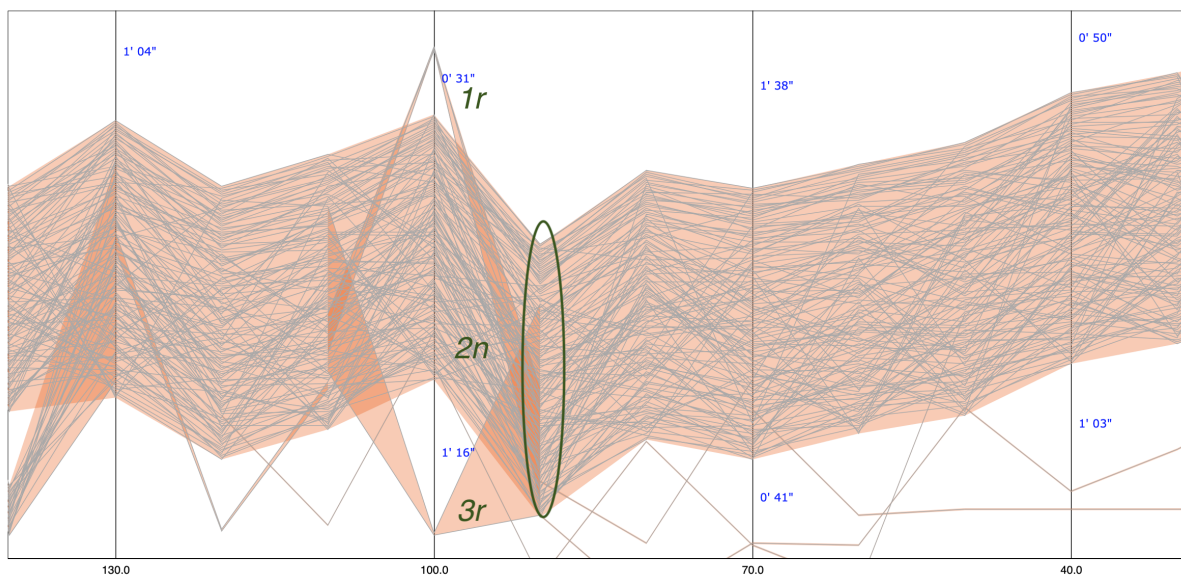


Figura 97: Visualització dels grups que es fusionen

### Grups que es continuen

En la figura 98 podem observar quina és l'evolució dels grups que continuen, en la imatge s'ha encerclat amb un mateix color aquells grups que continuen en diferents punts quilomètrics, és pot arribar a dir que els grups no canvien, però en canvi dintre d'aquests grups l'ordre d'arribada, es a dir les posicions, de cada ciclista si es veu afectada.

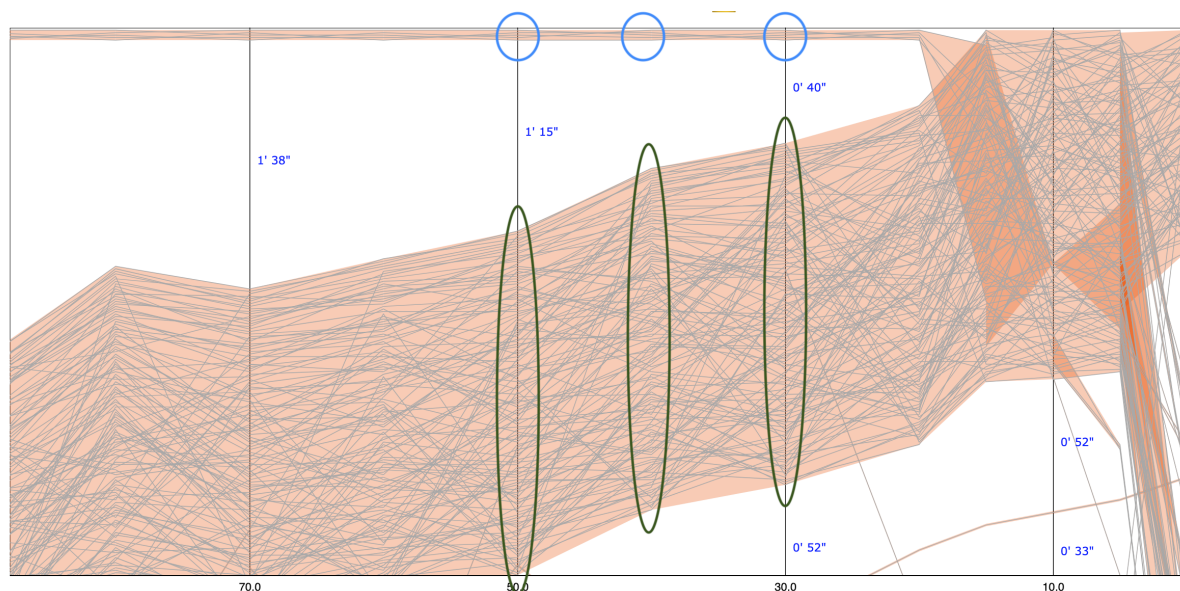


Figura 98: Visualització dels grups que es continuen

### 10.3.3 Seleccionar un ciclista o trajectòria

En la figura 99 s'observa un nou canvi en la pàgina. Hi han dos possibles processos per arribar aquesta imatge, una és fer clic a sobre d'una de les imatges dels diferents ciclistes que apareixen en la llista. El comportament que ha implica això és pintar de color vermell quina és la trajectòria que està associada al ciclista seleccionat. En aquesta mateixa trajectòria es mostren uns punts, on si es fa mouse hover amb el mouse es mostra la informació temporal del ciclista en aquell punt quilomètric. L'altre procés possible és haver seleccionat en concret aquella trajectòria en concret. La visualització i la funcionalitat a la pàgina seria la mateixa.

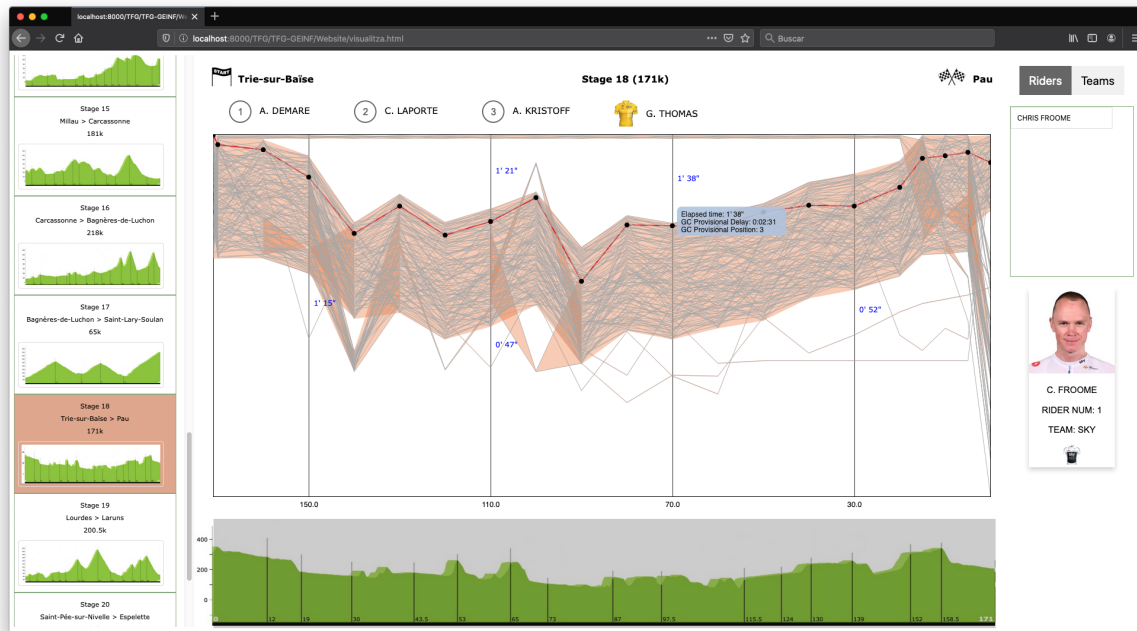


Figura 99: Visualització de seleccionar un ciclista

### 10.3.4 Seleccionar un equip

Per arribar a mostrar el que es veu en la figura 100, el primer que s'ha hagut que fer és seleccionar el botó de Teams de dalt a la dreta per mostrar tots els equips que hi ha la cursa, seguidament escollir-ne un i fer clic sobre de l'imatge. Llavors automàticament es mostren quines són les trajectòries seleccionades que representen cada un dels integrants de l'equip, amb els respectius punts.

Encara que està molt bé tenir una visualització clara de quina ha sigut l'evolució d'un equip en concret, també ens interessa saber a quin ciclista representa cada una de les trajectòries, és per això que en el seleccionar un equip, el tooltip no només mostra la informació temporal, sinó que a més a més també informa de qui és el ciclista, la informació que aporta és: el nom, número de dorsal i inicials de l'equip al que pertany. En la imatge que es mostra a continuació es pot veure l'exemple.

Una altra informació que aporta aquesta funcionalitat és la de veure quins són els membres de l'equip que s'ha seleccionat. En la captura següent es mostra el símbol de l'equip, juntament amb el nom. I més a sota una llista amb els noms del membres de l'equip, juntament amb el seu número de dorsal.

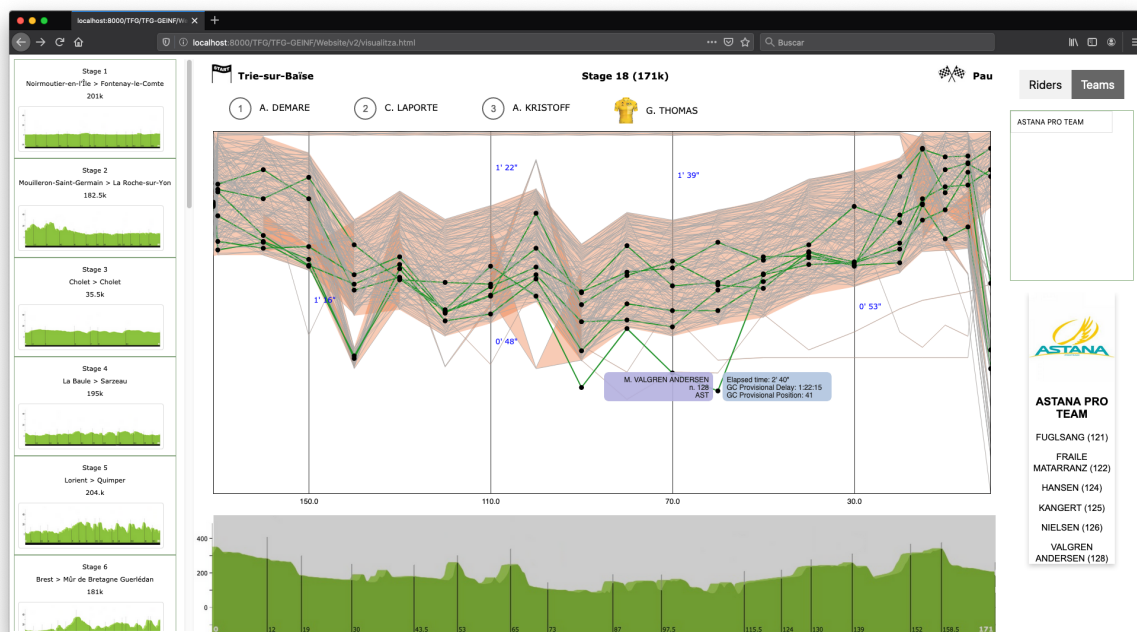


Figura 100: Visualització de seleccionar un equip

### 10.3.5 Aplicar zoom a la gràfica

Un dels altres requisits importants del projecte era donar la possibilitat de fer zoom en qualsevol moment. En la figura 101 que es mostra a continuació s'ha realitzat un cop s'ha fet zoom i posteriorment s'ha escollit un equip (fer-ho al revés, primer seleccionar equip i després fer zoom, té el mateix resultat).

En aquest exemple en concret s'ha escollit l'equip Wanty. Un cop es selecciona, automàticament es visualitzen les trajectòries dels participants d'aquest equip. Si ens posem a analitzar la imatge seguint veiem que, en el punt quilomètric 50 tots els membres de l'equip formen un sol grup a excepció d'un. És en aquest cas quan la funcionalitat de zoom és molt útil per veure amb detall aquest tram. Un cop es fa zoom, si es vol saber més informació sobre la trajectòria que està separada de la resta del grup, només cal fer mouse over per mostrar més informació en aquell punt, concretament podem saber que la trajectòria representa el ciclista A. PASQUALON, amb dorsal 216 juntament amb la seva informació temporal, aquesta informació la podem saber gràcies als tooltips que apareixen en la gràfica.

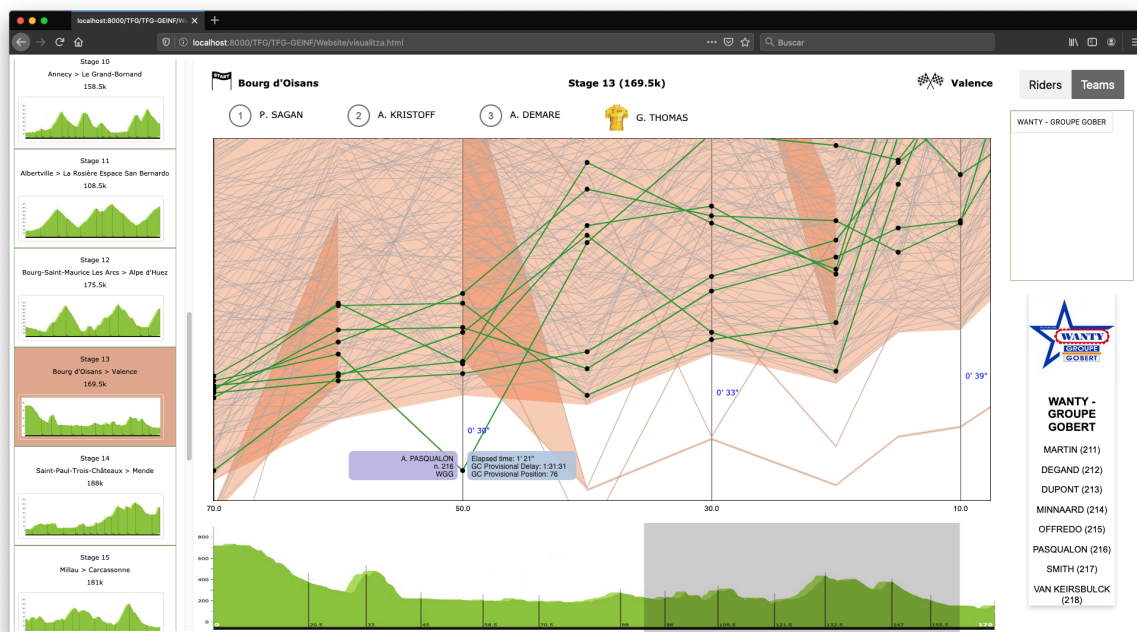


Figura 101: Exemple de zoom amb un equip seleccionat

### 10.3.6 Comparar dos ciclistes

Un dels altres requeriments que es necessitaven assolir era tenir la capacitat de comparar dos ciclistes que estan participant una mateixa etapa. En aquest cas s'han escollit un parell de trajectòries que en la gràfica s'observa que forma una punxa i es crea un polígon de dos integrants. Pot ser interessant voler saber quins són aquests dos membres i perquè van junts, un cop es seleccionen ens fixem que aquest parell de ciclistes són del mateix equip. Pot arribar a tenir sentit ja que durant la cursa potser van decidir anar una estona junts. També s'ha de diferenciar a qui li pertany cada trajectòria, és per això que es pinten cada un amb els colors respectius.

Un petit comentari a fer és que quan es va implementar aquesta funcionalitat, ens vam donar compte que seleccionar un sol ciclista, i representar-lo amb un color en concret, comportava la mateixa feina que seleccionar-ne més d'un. És per això, que vam haver de parlar amb en Pere-Pau, (expert en la visualització de les dades) per preguntar-li si seguíem amb l'objectiu inicial, de comparar 2 ciclistes, o ampliàvem una mica aquesta funcionalitat i així oferir a l'usuari seleccionar (o comparar) més d'un ciclista. Després de parlar-ho amb ell és va decidir continuar amb la primera idea, ja que era la millor opció per visualitzar les dades.

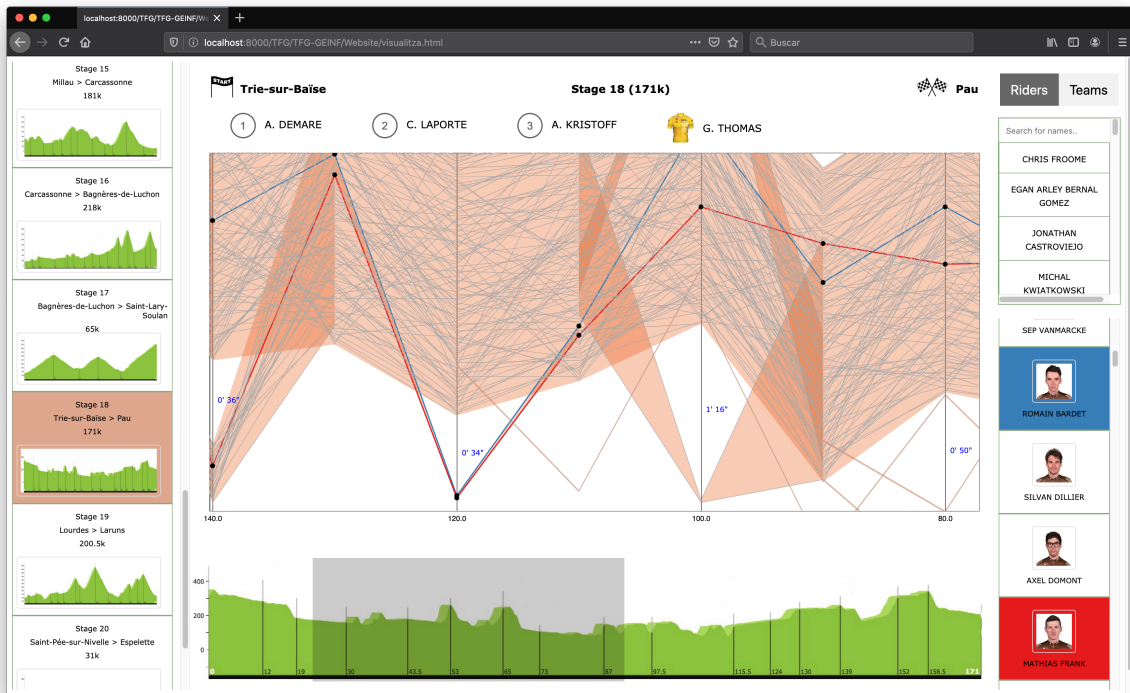


Figura 102: Visualització de comparar dos ciclistes

## 11 Conclusions

L'objectiu principal del projecte era visualitzar l'evolució d'una cursa, en concret era crear una aplicació capaç de mostrar l'evolució individual i grupal de curses de ciclistes amb un recorregut prefixat. Com ja s'ha mencionat en aquest document, no han de ser necessàriament de ciclistes, però les dades que nosaltres teníem eren aquestes i és per això que ho hem implementat d'aquesta manera. En aquest aspecte considerem que s'han assolit els requeriments suggerits al principi, ja que s'ha aconseguit mostrar l'evolució dels ciclistes en diferents etapes d'una cursa, utilitzant diferents representacions per cada un dels elements com ja són les trajectòries per representar els ciclistes, polígons en representació a l'evolució del grup, línies verticals per identificar els quilòmetres restants per acabar l'etapa, diferents estils de colors per representar elements seleccionats i oferir una interacció amb l'usuari amb la que es pot fer un anàlisi de la cursa, en aquest darrer cas la funcionalitat més important és el poder fer zoom en la gràfica i veure amb més detall els diferents elements que apareixen. Totes aquestes característiques poden facilitar la feina a l'objectiu original, que és trobar patrons de moviment en el transcurs d'una cursa.

El resultat final que hem obtingut per la representació dels grups ha sigut bastant efectiva, ja que considerem que representa bastant bé l'evolució dels ciclistes. L'altra opció que ja hem mencionat era utilitzar diagrames Sankey, aquest tipus de diagrama el vam descartar ja que consideràvem que la representació dels grups no coincidiria visualment amb les trajectòries. És veritat que amb el diagrama de Sankey hauríem vist amb més claredat el flux que es va formant entre els diferents trams quilomètrics, però juntament amb les trajectòries no s'arribaria a tenir una visualització tan clara com l'actual. Per això, aquesta representació en el seu moment es va descartar.

Pel que fa els nou coneixements obtinguts, durant el transcurs d'aquest projecte, estic bastant satisfet, no solament en l'àmbit de programació per haver après noves llibreries o de visualització de les dades, que també, sinó per haver après a gestionar millor la feina a fer. Per mi ha sigut molt important haver tingut en tot moment una organització de cada una de les tasques a realitzar, d'aquesta manera podia veure amb perspectiva quins eren les més urgents i quines no. Un altra aspecte que vull valorar és haver estat treballant amb altres persones, en Jose, Pere-Pau i la meva tutora Marta. Fer un projecte amb altres persones implica tenir una comunicació constant per tractar els diferents aspectes a treballar i tenir una aprovació conjunta. No és suficient amb tenir els coneixements necessaris per implementar codi, sinó també és important saber comunicar-se amb la resta de l'equip per expressar els teus motius de disseny. És per això que agraeixo haver estat particip d'aquest grup.



## 12 Treball futur

En aquest projecte ja coneixem els requeriments inicials i hem vist com s'han implementat. Però aquesta aplicació web té possibles ampliacions de cares a millorar la interpretació de les dades.

La primera de les millores que es podria implementar té a veure amb la comparació de dos ciclistes (o trajectòries). Per ara, es poden seleccionar dos ciclistes o dues trajectòries, però no es pot seleccionar una trajectòria i després a un ciclista, ni tampoc seleccionar un ciclista i després una trajectòria. És per això que poder fer una fusió de les dues funcionalitats milloraria l'aplicació. D'aquesta manera es podria comparar una trajectòria en concret amb un ciclista en concret (o viceversa). A part d'això, un cop es tenen els elements seleccionats es pinten les trajectòries i els ciclistes amb un mateix color per identificar-los, però es segueixen mostrant a la barra lateral la resta de ciclistes de la cursa, una possible implementació seria mostrar únicament aquests ciclistes seleccionats.

Una altra de les altres ampliacions que es podria fer, i la de les més interessants, seria poder utilitzar aquesta aplicació en temps real. Els algorismes que s'han implementat per generar cada un dels diferents elements de la pàgina són ho suficientment òptims perquè es puguin visualitzar les dades a mesura que es vagin obtenint. Per implementar aquesta millora s'hauria de refactoritzar tot el codi per adaptar-ho aquestes noves necessitats.

Una altra funcionalitat que es podria afegir seria la de penjar aquesta aplicació online. D'aquesta manera qualsevol usuari podria accedir i visualitzar aquestes dades des de el seu dispositiu. Aquesta ampliació arreglaria la dependència de tenir les dades i un servidor local en l'ordinador. Amb la idea de penjar-ho online, també es podria afegir la funcionalitat de permetre als usuaris penjar els seus propis fitxers de dades. D'aquesta manera serien els usuaris qui gestionarien aquests fitxers, permetent que la resta d'usuaris puguin visualitzar les noves dades compartides. Per implementar aquesta nova idea el primer que s'hauria de fer és una migració de totes les dades cap a un altra tipus de servidor, ja que amb un servidor local aquesta ampliació no es possible. A més a més s'hauria de modificar i part de la visualització, on deixés pujar fitxers propis dels usuaris.

Com s'ha comentat en l'apartat 8.4 el fitxer d'entrada té un estil definit, però en les dades reals que nosaltres tenim hi ha molta més informació que no està implementada a la pàgina. Una molt bona idea seria agafar la resta de dades dels fitxers .csv i intentar afegir-les a l'aplicació. Per fer això està clar que s'hauria de tornar a dissenyar la pàgina per saber on col·locar cada nou element. Un de les dades que tenim i que no s'utilitzen són les dades de la velocitat. Seria molt interessant agafar aquestes dades i fer una estadística de les diferents velocitats que van apareixent en l'etapa i plasmar-les en la web. Això

seria de gran ajudar a voler analitzar millor les dades de l'evolució de la cursa.

Respecte a la codificació de la pàgina, estaria molt bé fer una refactorització del codi, però sobretot de la plantilla d'estils CSS. Seria molt recomanable fer una migració de l'estil CSS de la pàgina a les llibreries Bootstrap, al menys els elements que siguin possibles, ja que estan més generalitzats i ajudaria a altres dissenyadors o programadors a entendre millor la codificació del programa.

Una altra idea que es va comentar amb els membres del projecte va ser en implementar en el futur aquesta aplicació per dispositius mòbil. En l'aplicació web que s'ha implementat, la interacció que es fa amb l'usuari es fa a partir d'events que l'usuari pugui generar amb el mouse o trackpad i el teclat, entre aquests events els que són de clic a sobre de les trajectòries són molt precisos ja que poden afectar a nivell de píxel. En canvi si l'aplicació s'executés en un dispositiu mòbil, l'única interacció possible que hi hauria seria la dels dits, i a diferència dels events anteriors aquests no poden ser tan precisos. Per crear els diferents elements de la web i per gestionar la seva interacció s'ha utilitzat llibreries de JavaScript, com D3 o jQuery. I aquestes són totalment compatibles amb dispositius mòbils, però encara que sigui així es segueix tenint el problema de la precisió. Segurament, si es volgués portar aquesta aplicació a dispositius mòbils s'hauria de crear un altre tipus de disseny per representar les dades.

## 13 Bibliografia

### Referències

- [1] Group evolution patterns in running races <https://www.dropbox.com/s/xg5kcdt9ud7hb5r/AnalisiTrajectoriesFinal.pdf?dl=0>
- [2] How do you set up a local testing server? [https://developer.mozilla.org/en-US/docs/Learn/Common\\_questions/set\\_up\\_a\\_local\\_testing\\_server](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/set_up_a_local_testing_server)
- [3] Método rápido para servir archivos a través de HTTP (Servidor HTTP) con SimpleHTTPServer <http://www.3engine.net/wp/2016/03/metodo-rapido-para-servir-archivos-a-traves-de-http-servidor-http-con-simplehttpserver/>
- [4] What is Scrum? <https://www.scrum.org/resources/what-is-scrum>
- [5] Pro Cycling Stats <https://www.procyclingstats.com>
- [6] Visualizing Personal Progress in Participatory Sports Cycling Events [https://www.dropbox.com/sh/n97u58gækwjb23y/AADB6\\_NSlRHgcU2ui92io8n2a?dl=0&preview=Vis\\_progress\\_cycling.pdf](https://www.dropbox.com/sh/n97u58gækwjb23y/AADB6_NSlRHgcU2ui92io8n2a?dl=0&preview=Vis_progress_cycling.pdf)
- [7] Cycling journeys using interactive graphics [https://www.dropbox.com/sh/n97u58gækwjb23y/AADB6\\_NSlRHgcU2ui92io8n2a?dl=0&preview=14-Cycling\\_journeys.pdf](https://www.dropbox.com/sh/n97u58gækwjb23y/AADB6_NSlRHgcU2ui92io8n2a?dl=0&preview=14-Cycling_journeys.pdf)
- [8] State of the Art of Sports Data Visualitzation [https://www.dropbox.com/sh/n97u58gækwjb23y/AADB6\\_NSlRHgcU2ui92io8n2a?dl=0&preview=eurovis18-star-sports-data-visualization.pdf](https://www.dropbox.com/sh/n97u58gækwjb23y/AADB6_NSlRHgcU2ui92io8n2a?dl=0&preview=eurovis18-star-sports-data-visualization.pdf)
- [9] Where should you store your data? <https://www.labfolder.com/cloud-vs-local-server/>
- [10] W3Schools <https://www.w3schools.com/about/default.asp>
- [11] Data-Driven Documents <https://d3js.org>
- [12] d3-transitions <https://github.com/d3/d3-transition>
- [13] Color Brewer <https://colorbrewer2.org/>
- [14] Web security [https://developer.mozilla.org/en-US/docs/Learn/Server-side/First\\_steps/Website\\_security](https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Website_security)
- [15] The D3.js Graph Gallery <https://www.d3-graph-gallery.com/index.html>

- [16] Changes in D3 versions <https://github.com/d3/d3/blob/master/CHANGES.md>
- [17] Pro Cycling Stats <https://www.procyclingstats.com/race/tour-de-france/2018/gc/stages/all-stage-profiles>
- [18] d3-brush <https://github.com/d3/d3-brush>
- [19] d3-zoom <https://github.com/d3/d3-zoom>
- [20] Web Application Viewer - generate diagram from application <https://github.com/bakunin95/wavi>

## 14 Annexos

### 14.1 Generar grups de ciclistes

```
1 function getGroupRidersStage(elapsedDataRider){
2
3     var grupsCiclistesEtapa = [];
4     for(var i = 0; i < columnNameElapsedKm.length; i++){
5         var grupsCiclistesKM = [];
6         ///ordenar els ciclistes per km "i"
7         var posicionsCiclista = [];
8         for(var j = 0; j < elapsedDataRider.length; j++){
9             if(!isNaN(elapsedDataRider[j][i].y)){ //comprova que sigui una punt v lid
10                posicionsCiclista.push(elapsedDataRider[j][i])
11            }
12        }
13        posicionsCiclista.sort(function(a,b) { //ordena les posicions dels ciclistes
14            if( a.x == b.x) return a.y-b.y;
15            return a.x-b.x;
16        });
17        for(var k = 0; k < posicionsCiclista.length; k++){
18            var grupN = []
19            if(esPotFerGrupCiclista(k, posicionsCiclista)){
20                var puntIni = k;
21                grupN.push(posicionsCiclista[k]);
22                while(posicionsCiclista[k].y === (posicionsCiclista[k+1].y -1)){
23                    k++;
24                    grupN.push(posicionsCiclista[k]);
25                    if(k === posicionsCiclista.length - 1){
26                        break;
27                    }
28                }
29                grupsCiclistesKM.push(grupN);
30            }
31            else{ // es un ciclista que va sol i no te grup
32                grupN.push(posicionsCiclista[k]);
33                grupsCiclistesKM.push(grupN);
34            }
35        }
36        grupsCiclistesEtapa.push(grupsCiclistesKM);
37    }
38    return grupsCiclistesEtapa;
39 }
```

Listing 2: Zoomed Function Javascript

## 14.2 Generació dels polígons

```

1 function getPolygonsGroups(columnsNameElapsedKm,grupsCiclistesEtapa, elapsedDataRider){
2
3     var polygonsGrups = [];
4     for(var i = 0; i < columnsNameElapsedKm.length - 1; i++){ //per cada km
5         var grupsCiclistaKM = grupsCiclistesEtapa[i];
6         for(var j = 0; j < grupsCiclistaKM.length; j++){ //per cada grup de ciclistes
7             var grupCiclista = grupsCiclistaKM[j];
8             var puntTractades = [];
9             for(var k = 0; k < grupCiclista.length; k++){ //per cada ciclista
10                //inicializem les punt del poligon
11                var puntPrimerCiclista = elapsedDataRider[grupCiclista[k].id][i];
12                var puntUltimCiclista = elapsedDataRider[grupCiclista[k].id][i];
13                var puntPrimerCiclistaSegGrup = elapsedDataRider[grupCiclista[k].id][i + 1];
14                var puntUltimCiclistaSegGrup = elapsedDataRider[grupCiclista[k].id][i + 1];
15                if(!puntTractades.includes(elapsedDataRider[grupCiclista[k].id][i])){
16                    var seguentpuntCiclista = elapsedDataRider[grupCiclista[k].id][i + 1];
17                    var grupSeguent = trobarGrup(seguentpuntCiclista, grupsCiclistesEtapa[i +
18                1]);
19                if(grupSeguent !== -1){ //no hi ha info del ciclista
20                    for(var l = k; l < grupCiclista.length; l++){
21                        var puntSubGrup = elapsedDataRider[grupCiclista[l].id][i]
22                        var puntSubGrupSeg = elapsedDataRider[grupCiclista[l].id][i + 1]
23                        var grupSeguentCandidat = trobarGrup(puntSubGrupSeg,
24                    grupsCiclistesEtapa[i + 1]);
25
26                        if((grupSeguentCandidat !== -1) && (sonGrupsIguals(grupSeguent,
27                    grupSeguentCandidat))){
28                            puntTractades.push(puntSubGrup)
29                            //Actualitza ultim ciclista del costa esq.
30                            puntUltimCiclista = puntSubGrup;
31                            //Actualitza punt costat dret
32                            if(puntSubGrupSeg.y > puntUltimCiclistaSegGrup.y){
33                                puntUltimCiclistaSegGrup = puntSubGrupSeg;
34                            }
35                            else if(puntSubGrupSeg.y < puntPrimerCiclistaSegGrup.y){
36                                puntPrimerCiclistaSegGrup = puntSubGrupSeg;
37                            }
38                        }
39                    }
40                }
41            }
42        }
43    }
44 }

```

```
36     }
37     //Si es un ciclista que va sol
38     if(puntPrimerCiclistaSegGrup === puntUltimCiclistaSegGrup){
39
40         puntPrimerCiclista = {id: puntPrimerCiclista.id, x: puntPrimerCiclista
41         .x, y: puntPrimerCiclista.y - 0.5};
42         puntPrimerCiclistaSegGrup = {id: puntPrimerCiclistaSegGrup.id, x:
43         puntPrimerCiclistaSegGrup.x, y: puntPrimerCiclistaSegGrup.y - 0.5};
44         puntUltimCiclista = {id: puntUltimCiclista.id, x: puntUltimCiclista.x,
45         y: puntUltimCiclista.y + 0.5};
46         puntUltimCiclistaSegGrup = {id: puntUltimCiclistaSegGrup.id, x:
47         puntUltimCiclistaSegGrup.x, y: puntUltimCiclistaSegGrup.y + 0.5};
48     }
49     //Afegir el punts com un pol gon nou
50     polygonsGrups.push({points: [puntPrimerCiclista,
51     puntPrimerCiclistaSegGrup, puntUltimCiclistaSegGrup, puntUltimCiclista]});
52 }
53 }
54 }
55 }
56 }
57 }
58 }
59 }
60 }
61 }
62 }
63 }
64 }
65 }
66 }
67 }
68 }
69 }
70 }
71 }
72 }
73 }
74 }
75 }
76 }
77 }
78 }
79 }
80 }
81 }
82 }
83 }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }
101 }
102 }
103 }
104 }
105 }
106 }
107 }
108 }
109 }
110 }
111 }
112 }
113 }
114 }
115 }
116 }
117 }
118 }
119 }
120 }
121 }
122 }
123 }
124 }
125 }
126 }
127 }
128 }
129 }
130 }
131 }
132 }
133 }
134 }
135 }
136 }
137 }
138 }
139 }
140 }
141 }
142 }
143 }
144 }
145 }
146 }
147 }
148 }
149 }
150 }
151 }
152 }
153 }
154 }
155 }
156 }
157 }
158 }
159 }
160 }
161 }
162 }
163 }
164 }
165 }
166 }
167 }
168 }
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }
190 }
191 }
192 }
193 }
194 }
195 }
196 }
197 }
198 }
199 }
200 }
```

Listing 3: Funció generar polígons

### 14.3 Comportament zoom

```
1 function zoomed() {
2
3     // create new scale objects based on event
4     var new_xScale = d3.event.transform.rescaleX(xScale);
5     var new_yScale = d3.event.transform.rescaleY(yScale);
6     // guarda l'escala nova com l'actual
7     xScaleLabel = new_xScale;
8     yScaleLabel = new_yScale;
9     // buscar l'escala actual, per saber quines etiquetes entre grups s'ha de mostrar
10    var zoomTransformString = d3.event.transform.toString();
11    var splitZoomString = zoomTransformString.split(" ");
12    var zoomScale = parseFloat(splitZoomString[1].slice(splitZoomString[1].indexOf("(")
13    + 1, splitZoomString[1].indexOf(")"))
14    // obtenir les l nies quilom triques que ha de mostrar
```

```
14   var arr_kmMostrar = updateAxisXKm(new_xScale.domain(), columnsKm);
15   // re-scale axes
16   svg.select(".y.axis")
17     .call(yAxis.scale(new_yScale))
18     .selectAll(".tick")
19     .style("display", "none");
20
21   svg.select(".x.axis")
22     .call(xAxis.scale(new_xScale))
23     .selectAll(".tick")
24     .style("display", function(){
25       if(arr_kmMostrar.includes(parseFloat(this.textContent))) return "";
26       else return "none";
27     });
28   // re-draw line
29   plotLine = d3.line()
30     .x(function (d) {
31       return new_xScale(d.x);
32     })
33     .y(function (d) {
34       return new_yScale(d.y);
35     });
36
37   svg.selectAll('path.line').attr("d", plotLine);
38   //re-draw polygons
39   polygonsScale = function(d){
40     return d.points.map(function(d) { return [new_xScale(d.x),new_yScale(d.y)].join(","); }).join(" ");
41   }
42   svg.selectAll("polygon").attr("points", polygonsScale)
43   //re-draw labels
44   svg.selectAll("text.timeGroup")
45     .attr("x", function(d) {
46       return new_xScale(d.x) + 10;
47     })
48     .attr("y", function(d) {
49       return new_yScale(d.y);
50     })
51     .style("visibility", function(d, i) {
52       return (d.timeSeconds * zoomScale) > corredorMaxElapsed(elapsedDataRider) *
53         varEtiquetesTemps / height && arrKmMostrar.includes(d.x) ? "visible" : "hidden";
54     });
55   //re-draw dots
56   svg.selectAll("circle")
```



```

56     .attr("cx", function(d) {
57         return new_xScale(d.x);
58     })
59     .attr("cy", function(d) {
60         return new_yScale(d.y);
61     })
62     //re-draw perfil de l'etapa
63     var percentatgeIni = 97.5
64     var distanciaCursa = columnsKm[0]
65     var newAttrX = (Math.abs(percentatgeIni - (new_xScale.domain()[0] * 100) /
66     distanciaCursa)).toString() + "%";
67     var newWidth = 100 - Math.abs(percentatgeIni - (new_xScale.domain()[0] * 100) /
68     distanciaCursa);
69
70     var newAttrwidht = (Math.abs(newWidth - (new_xScale.domain()[1] * 100) /
71     distanciaCursa)).toString() + "%";
72
73     $("#brush").attr("x", newAttrX)
74     $("#brush").attr("width", newAttrwidht)
75 }

```

Listing 4: Funció Zoomed

## 14.4 Trobar punts quilomètrics a mostrar

```

1 function updateAxisXKm(arrayDomain, columnsKm){
2
3     var km_maxim_vist = arrayDomain[0];
4     var km_minim_vist = arrayDomain[1];
5     var rang_visio = km_maxim_vist - km_minim_vist;
6     var inc = rang_visio / (showNAxisChart + 1);
7     var inc_ampliat = Math.ceil(inc/separacioEntreKm) * separacioEntreKm;
8     var rang_ampliat = inc_ampliat * (showNAxisChart + 1);
9     var inc_rang = rang_ampliat - rang_visio;
10    var km_min = km_minim_vist - inc_rang / 2;
11    arrKmMostrar = [];
12    if(rang_visio > (separacioEntreKm * (showNAxisChart + 1))){
13        for (var i = 1; i <= showNAxisChart ; i++){
14            var km = Math.round((km_min + i * inc_ampliat) / separacioEntreKm) *
15            separacioEntreKm;
16            arrKmMostrar.push(km);
17        }
18    }
19 }

```

```
18     else{
19         var inici;
20         for(var i = 0; i < columnsKm.length; i++){ //busquem el primer km vist
21             if(km_maxim_vist >= parseFloat(columnsKm[i])){
22                 inici = i;
23                 break;
24             }
25         }
26         var final = columnsKm.length - 1;
27         for(var i = inici + 1; i < columnsKm.length; i++){ //busquem el darrer km vist
28             if(km_minim_vist <= parseFloat(columnsKm[i])){
29                 final = i;
30             }
31         }
32         if(final - inici < showNAxisChart){ //si no n'hi ha masses, els guardo tots
33             for(var j = inici; j < columnsKm.length; j++){
34                 arrKmMostrar.push(parseFloat(columnsKm[j]))
35             }
36         }
37         else{ //n'hi ha masses, miro quins hauria de mostrar i els busco
38             var kms = [];
39             for(var i = 0; i < showNAxisChart; i++){
40                 kms.unshift(km_min + i * inc_ampliat)
41             }
42             var j=0;
43             for(var i = inici; i < final-1 && j < showNAxisChart; i++){
44                 if(columnsKm[i] >= kms[j] && columnsKm[i+1] < kms[j]){
45                     arrKmMostrar.push(parseFloat(columnsKm[i]))
46                     j++;
47                 }
48             }
49         }
50     }
51     arrKmMostrar = arrKmMostrar.filter(function(element){
52         return element <= arrayDomain[0] && element >= arrayDomain[1];
53     });
54     return arrKmMostrar;
55 }
```

Listing 5: Funció trobar punts quilomètrics per mostrar

## 15 Manual d'usuari i/o instal·lació

El resultat final d'aquest projecte ha sigut una aplicació web per visualitzar l'evolució dels ciclistes en les diferents etapes de la cursa del Tour de França. Aquesta aplicació es pot executar en qualsevol navegador, sempre i quan s'estigui executant el servidor local i òbviament es tinguin les dades dels fitxer d'entrada. En aquest darrer apartat s'explicarà com es pot utilitzar aquesta pàgina web per analitzar els ciclistes. En les captures següents es mostra quina es l'etapa que seleccionada.

### 15.1 Seleccionar una etapa de la cursa

Un cop s'ha carregat la pàgina es visualitzarà un graf amb l'evolució dels ciclistes en l'etapa número 13. Hem escollit la número 13 ja que com s'ha mencionat només disposàvem de les dades de 3 etapes, la 11, 13 i 18. Per escollir qualsevol de les altres dues etapes només cal fer scroll amb el mouse i fer clic esquerra amb el ratolí en l'etapa que desitgem. Un cop fem clic, l'etapa escollida quedarà seleccionada.

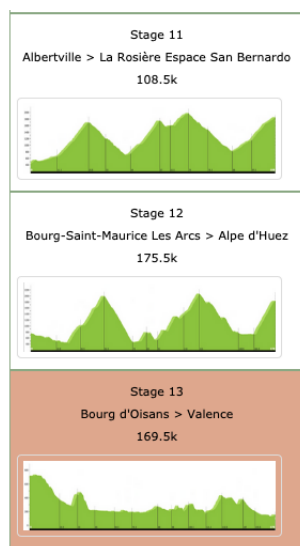


Figura 103: Seleccionar etapa (1)

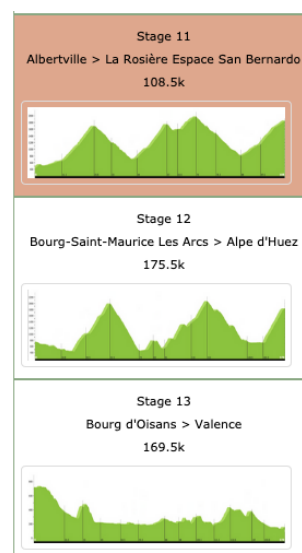


Figura 104: Seleccionar etapa (2)

### 15.2 Fer zoom en la gràfica

La funcionalitat de fer zoom es pot utilitzar en qualsevol moment en que s'està executant l'aplicació. Hi han diferents opcions per fer zoom-in o zoom-out:

- Scroll amb la rodeta del mouse: es pot fer scroll amb la rodeta del mouse, en qualsevol posició del graf. On estigui posicionat el cursor del ratolí es farà zoom en aquella direcció.
- Zoom in: per fer zoom in també es pot clicant dos cops el botó esquerra del ratolí, el curso ha d'estar en alguna zona del graf.

- Trackpad: en el cas d'estar executant l'aplicació en un portàtil, també es pot fer zoom utilitzant el trackpad de l'ordinador.

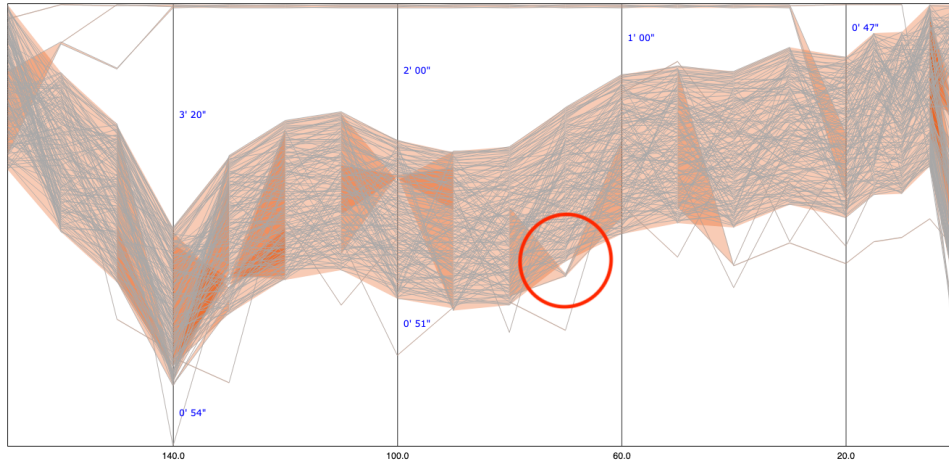


Figura 105: Gràfica sense fer zoom

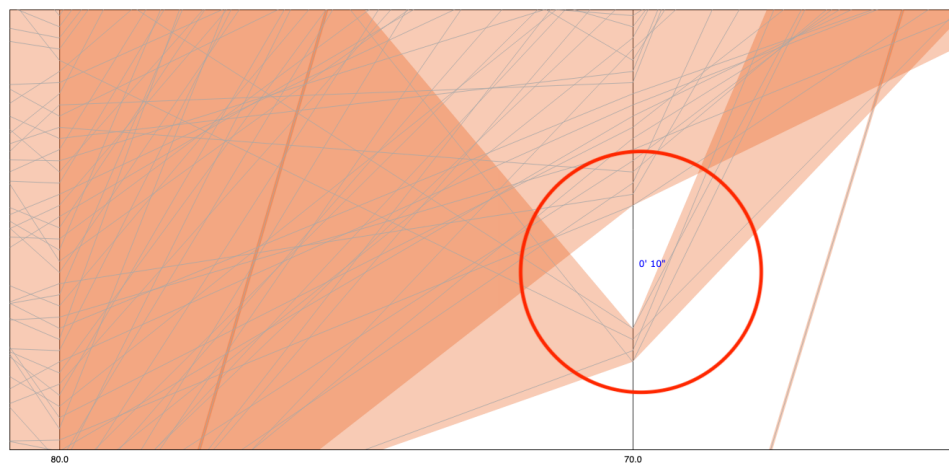


Figura 106: Gràfica fent zoom

### 15.3 Seleccionar una trajectòria

Per seleccionar una única trajectòria és suficient amb fer clic esquerra en una de les trajectòries que es faci hover. Automàticament el color de la trajectòria quedarà fixa i en la barra lateral dreta es detallarà més informació sobre el ciclista al que li pertany la trajectòria seleccionada.

### 15.4 Seleccionar dues trajectòries

Per seleccionar dues trajectòries i veure a quins ciclistes pertany cada una de les trajectòries, s'han de seguir els següents passos:

1. Mantenir pressionat la tecla Alt
2. Fer clic esquerra en una de les trajectòries que es fa hover
3. Fer clic esquerra en una altra trajectòria de la gràfica
4. Deixar de mantenir pressionat la teclat Alt
5. Un cop ja s'hagin acabat d'analitzar les dades, s'han de desmarcar, explicat en l'apartat 15.8.

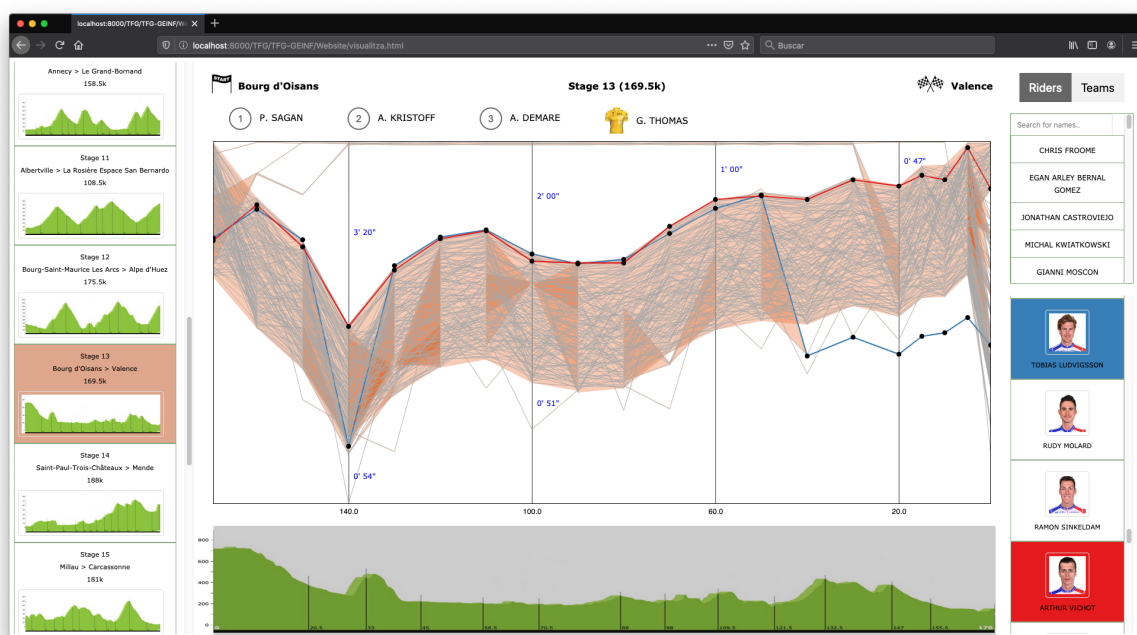


Figura 107: Seleccionar dues trajectòries

## 15.5 Seleccionar un ciclista

Per seleccionar un ciclista és tan senzill com primer de tot assegurar-se de que es té seleccionat el botó de Riders, a continuació només s'ha d'escollir un dels ciclistes que apareix en la barra lateral dreta i fer clic esquerra en l'imatge o en el nom. Per facilitar la cerca es pot filtrar per nom. En la figura 99 en podem veure un exemple.

## 15.6 Seleccionar un equip

Per seleccionar un equip s'ha de seguir la metodologia de l'apartat anterior, però assegurant-nos de tenir seleccionat el botó Teams. En la figura 100 en podem veure un exemple.

## 15.7 Seleccionar dos ciclistes

A part de seleccionar dues trajectòries, també hi ha la funcionalitat de seleccionar dos ciclistes en concret.

Per fer això s'han de seguir els següents passos:

1. Mantenir pressionat la tecla cmd (en Mac) o ctrl (en Windows)
2. Fer clic esquerra en una de les imatges d'un ciclista
3. Fer clic esquerra en una altra imatge d'un ciclista
4. Deixar de mantenir pressionat la tecla cmd o ctrl
5. Un cop ja s'hagin acabat d'analitzar les dades, s'han de desmarcar, explicat en l'apartat 15.8.

El resultat visual que hi haurà en la pàgina, és el mateix que el de seleccionar dues trajectòries, explicat en els apartats anteriors.

## 15.8 Desmarcar les trajectòries i elements seleccionats

Per desmarcar les trajectòries i els ciclistes seleccionats simplement s'ha de fer clic esquerra en qualsevol zona de la gràfica que no sigui algun element dels que es mostren.



ESCOLA POLITÈCNICA SUPERIOR

ENGINYERIA INFORMÀTICA

TFG

## Visualització de l'evolució de curses

Luis Alejandro Leon Corcuera