

## Facultad de Informática

### Grado de Ingeniería Informática

▪ Trabajo Fin de Grado ▪

Ingeniería de Software

Visualización de anotaciones web para la calificación de exámenes

---

Eduardo Pérez Capellán

Septiembre 2018



# Agradecimientos

En primer lugar, me gustaría agradecer a mi director de proyecto Óscar por proponerme realizar este TFG. Agradecerle la confianza depositada en mí, por todo lo que he conseguido aprender gracias a él y por lo involucrado que ha estado durante todo el proyecto sabiendo guiarme correctamente a lo largo de todo el año.

Así mismo, me gustaría agradecer a Haritz por su infinita paciencia conmigo y estar ahí en todo momento para ayudarme y aconsejarme con mis múltiples problemas a lo largo del TFG. La verdad que su ayuda ha resultado muy valiosa y vital para poder finalizar a tiempo.

También quiero dar las gracias a mis compañeros y en especial a aquellos con los que más trato he tenido, por los momentos que hemos compartido dentro y fuera de las aulas.

Tampoco olvidarme de Josemi por orientarme y haberme recomendado a Óscar para realizar este proyecto.

Me gustaría agradecer también a los profesores que me han sufrido como alumno durante estos últimos 4 años por todo lo que he aprendido de ellos.

También agradecer a mi familia y amigos que tanto me han apoyado y confiado en mí.

Por último, agradecer el tiempo dedicado y el interés en el proyecto a los lectores de esta memoria.



# Resumen

---

Este documento corresponde a la memoria del Trabajo de Fin de Grado “Visualización de anotaciones web para la calificación de exámenes” desarrollada por Eduardo Pérez Capellán para la titulación de Grado en Ingeniería Informática en la especialidad de Ingeniería del Software, en la Facultad de Informática de la Universidad UPV/EHU de Donostia-San Sebastián.

En este proyecto se ha desarrollado una extensión web que permite la visualización de anotaciones web en el contexto de la corrección de exámenes basada en rúbricas. Esta extensión permite al profesor analizar los exámenes y la asignatura.

El proyecto ha sido realizado bajo la dirección del profesor de la UPV/EHU Oscar Diaz García y la ayuda de Haritz Medina Camacho.

El presente documento, describe el proceso seguido para desarrollar la extensión, así como los principales objetivos y requisitos del proyecto, las tecnologías utilizadas, el diseño, detalles de la implementación y pruebas realizadas.

La extensión desarrollada se encuentra disponible para su descarga en el siguiente enlace: <https://chrome.google.com/webstore/detail/markandgoviz/lgpfmejklbhokgeohjjbhdinapcoljbo>.



# Índice

---

1. Introducción .....	13
1.1. Evaluación continua .....	14
1.2. Problemática .....	15
1.3. Anotaciones web .....	16
1.3.1. Hypothesis .....	17
1.3.2. Standard W3C para las anotaciones Web .....	18
1.4. Mark&Go .....	18
1.4.1. Definir rúbricas.....	19
1.4.2 Corregir exámenes.....	20
1.4.2.1. Encontrar evidencias .....	20
1.4.2.2. Corregir examen .....	21
1.5. Objetivo .....	23
1.5.1 Visualización de datos .....	23
2. Planificación.....	25
2.1. Alcance .....	26
2.1.1 Exclusión .....	26
2.2. EDT .....	26
2.3. Estimación de dedicación .....	27
2.4. Diagrama de Gantt .....	28
2.5. Recursos humanos .....	28
2.6. Caracterización del sistema de información y del sistema de comunicaciones en el proyecto .....	28
2.6.1. Sistema de Información .....	28
2.6.2. Formatos.....	29
6.6.3. Copias de seguridad .....	29
6.6.4. Comunicaciones .....	29
2.7. Herramientas.....	30
2.8. Riesgos .....	31
2.9. Evaluación económica .....	33
3. Análisis de evidencias de evaluación recogidas mediante anotaciones web .....	35
3.1 Evidencias.....	36
3.2. Necesidades de análisis sobre evidencias .....	36
3.2.1. Requisitos de análisis de datos.....	36

3.2.2 Elección de gráficos.....	36
3.2.2.1. Diagrama alluvial.....	37
3.2.2.2. Diagrama de araña.....	39
3.2.2.3. Diagrama de dispersión .....	42
3.2.5. Tabla preguntas .....	44
4. Diseño: Arquitectura de la solución.....	45
4.1. Requisitos .....	46
4.1.1. Requisitos funcionales .....	46
4.1.2. Requisitos no funcionales .....	46
4.1.3. Requisitos legislativos .....	47
4.2. Casos de uso.....	47
4.3. Flujo de eventos .....	48
4.3.1. Visualizar diagrama de dispersión de evaluación continua.....	48
4.3.2. Visualizar diagrama alluvial .....	48
4.3.3. Permutar pregunta .....	49
4.3.4. Permutar examen .....	49
4.3.5. Eliminar pregunta.....	49
4.3.6. Eliminar examen .....	50
4.3.7. Visualizar diagrama de dispersión de un examen .....	50
4.3.8. Visualizar diagrama de araña .....	51
4.3.9. Seleccionar examen .....	51
4.4. Modelo de datos .....	52
5. Implementación.....	53
5.1. Aumentación web .....	54
5.2. Arquitectura de la extensión .....	54
5.3. API, lenguaje y librerías .....	56
5.3.1. API de Hypothesis .....	56
5.3.1.1. Obtener anotaciones web .....	56
5.3.1.2. Obtener grupos.....	58
5.3.2. Lenguaje: JavaScript.....	59
5.3.3. Librería de gráficos: D3.js.....	60
5.3.4. Librería para manipulación de datos: Lodash .....	60
5.4. Desarrollo .....	61
5.4.1 Diagrama alluvial.....	61
5.4.2. Diagrama de araña.....	64
5.4.3. Diagrama de dispersión de la evaluación continua .....	66

5.4.4. Diagrama de dispersión de un examen .....	68
5.5. Diagramas de secuencia .....	70
5.5.1. Visualizar diagrama de dispersión de un examen .....	70
5.5.2. Visualizar diagrama de dispersión de la evaluación continua .....	71
5.5.3. Visualizar diagrama de araña .....	72
5.5.4. Visualizar diagrama alluvial .....	73
5.6. Dificultades durante el desarrollo .....	74
5.6.1. Problema 1.....	74
5.6.1.2. Solución 1 .....	74
5.6.2. Problema 2.....	74
5.6.2.2. Solución 2 .....	74
5.6.3. Problema 3.....	74
5.6.3.1. Solución 3 .....	74
6. Pruebas.....	77
6.1. Pruebas.....	78
6.1.1. Prueba 1.....	78
6.1.2. Prueba 2.....	79
6.1.3. Prueba 3.....	79
6.1.4. Prueba 4.....	80
6.1.5. Prueba 5.....	80
6.1.6. Prueba 6.....	81
6.1.7. Prueba 7.....	81
6.1.8. Prueba 8.....	82
6.1.9. Prueba 9.....	82
6.1.10. Prueba 10.....	83
6.1.11. Prueba 11.....	84
6.1.12. Prueba 12.....	84
6.1.13. Prueba 13.....	85
6.1.14. Prueba 14.....	86
6.1.15. Prueba 15.....	86
6.1.16. Prueba 16.....	86
6.1.17. Prueba 17.....	86
6.1.18. Prueba 18.....	87
6.1.19. Prueba 19.....	87
6.1.20. Prueba 20.....	88
7. Seguimiento del proyecto.....	89
8. Conclusiones.....	92

8.1. Conclusión del proyecto .....	93
8.2. Conclusiones del proyecto .....	93
8.3. Mejoras y líneas futuras .....	93
9. Bibliografía.....	95
Anexo A. Desarrollo.....	97
A.1. Diagrama alluvial .....	98
A.2. Gráfico de araña.....	106
A.3. Diagrama de dispersión sobre la evaluación continua .....	111
A.4. Diagrama de dispersión sobre un examen .....	114
Anexo B. Actas reuniones.....	117
Anexo C. Manual de usuario.....	129

# Lista de Figuras y Tablas

---

## FIGURAS

Ilustración 1. Ciclo de la evaluación continua .....	14
Ilustración 2. Ejemplo de criterios de rúbrica .....	16
Ilustración 3. Anotación web .....	17
Ilustración 4. Extensión web de Hypothesis .....	18
Ilustración 5. Establecer rango de valores .....	19
Ilustración 6. Identificar evidencias .....	20
Ilustración 7. Corrección de un examen .....	21
Ilustración 8. Ejemplo de criterios de rúbrica .....	22
Ilustración 9. Ejemplo de estructura JSON de una anotación web .....	22
Ilustración 10. Ejemplo de anotación web de los criterios de rúbrica .....	23
Ilustración 11. EDT del proyecto .....	26
Ilustración 12. Diagrama de Gantt .....	28
Ilustración 13. Ejemplo de diagrama alluvial .....	37
Ilustración 14. Diagrama alluvial preguntas (1/2) .....	38
Ilustración 15. Diagrama alluvial exámenes (2/2) .....	39
Ilustración 16. Ejemplo de diagrama de araña .....	40
Ilustración 17. Diagrama de araña .....	41
Ilustración 18. Diagrama de araña de un examen .....	41
Ilustración 19. Ejemplo de diagrama de dispersión .....	42
Ilustración 20. Diagrama de dispersión de la evaluación continua .....	42
Ilustración 21. Diagrama de dispersión de un examen .....	43
Ilustración 22. Diagrama de casos de uso .....	47
Ilustración 23. Modelo de datos .....	52
Ilustración 24. Aumentación web .....	54
Ilustración 25. Arquitectura de una extensión web .....	54
Ilustración 26. Archivo manifiesto de la extensión .....	56
Ilustración 27. Estructura de las anotaciones .....	57
Ilustración 28. Ejemplo de la estructura de una anotación .....	58
Ilustración 29. Estructura del objeto grupo .....	59
Ilustración 30. Ejemplo de la estructura de un grupo .....	59
Ilustración 31. Galería de ejemplos de D3.js .....	60
Ilustración 32. Diagrama alluvial: estructura final .....	61
Ilustración 33. Ejemplo de la estructura final del diagrama alluvial .....	62
Ilustración 34. Diagrama de transformaciones del diagrama alluvial .....	63
Ilustración 35. Diagrama de araña: estructura final .....	64
Ilustración 36. Ejemplo de estructura final del diagrama de araña .....	65
Ilustración 37. Diagrama de transformaciones del gráfico de araña .....	65
Ilustración 38. Diagrama de dispersión de evaluación continua: Estructura final .....	66
Ilustración 39. Ejemplo de estructura final de diagrama .....	67
Ilustración 40. Diagrama de transformaciones del diagrama de dispersión de la evaluación continua .....	67
Ilustración 41. Diagrama de dispersión de un examen: Estructura final .....	68
Ilustración 42. Ejemplo de la estructura final de un diagrama .....	68

Ilustración 43. Diagrama de transformaciones del diagrama de dispersión de un examen .....	69
Ilustración 44. Diagrama de secuencia: Visualizar diagrama de dispersión de un examen .....	70
Ilustración 45. Diagrama de secuencia: Visualizar diagrama de dispersión de la evaluación continua.....	71
Ilustración 46. Diagrama de secuencia: Visualizar diagrama de araña .....	72
Ilustración 47. Diagrama de secuencia: Visualizar diagrama alluvial .....	73
Ilustración 48. Diagrama de araña planificada.....	75
Ilustración 49. Versión final del diagrama de araña.....	75
Ilustración 50. Prueba 1.....	78
Ilustración 51. Prueba 2.....	79
Ilustración 52. Prueba 3.....	79
Ilustración 53. Prueba 4.....	80
Ilustración 54. Prueba 5.....	80
Ilustración 55. Prueba 7.....	81
Ilustración 56. Prueba 8.....	82
Ilustración 57. Prueba 9.....	83
Ilustración 58. Prueba 10.....	83
Ilustración 59. Prueba 11.....	84
Ilustración 60. Prueba 12.....	85
Ilustración 61. Prueba 13.....	85
Ilustración 62. Prueba 17.....	87
Ilustración 63. Prueba 19.....	88
Ilustración 64. Gráfico comparativa dedicación estimada y real .....	90
Ilustración 65. Versión final diagrama de Gantt .....	91
Ilustración 66. Diagrama alluvial: estructura final .....	98
Ilustración 67. Botón de instalación .....	131
Ilustración 68. Establecer rango de valores .....	132
Ilustración 69. Identificar evidencias .....	133
Ilustración 70. Corrección de un examen .....	134
Ilustración 71. Diagrama de dispersión en la página de perfil .....	135
Ilustración 72. Botón para visualizar gráfico alluvial .....	135
Ilustración 73. Diagrama alluvial.....	136
Ilustración 74. Intercambiar preguntas .....	136
Ilustración 75. Intercambiar examen.....	137
Ilustración 76. Botón para visualizar diagrama de dispersión.....	137
Ilustración 77. Visualizar gráfico de araña .....	138
Ilustración 78. Seleccionar examen .....	138

## TABLAS

Tabla 1. Dedicación estimada .....	27
Tabla 2. Riesgos del proyecto .....	32
Tabla 3. Preguntas .....	36
Tabla 4. Tabla relación preguntas-gráficas .....	44
Tabla 5. Parámetros de la llamada a la API para obtener anotaciones.....	57
Tabla 6. Definición del objeto anotación .....	58
Tabla 7. Definición del objeto grupo.....	59
Tabla 8. Comparativa entre la dedicación estimada y real .....	90

# 1. Introducción

---

En este capítulo se presentarán los antecedentes del proyecto. Por un lado, se detalla el impacto de la evaluación continua en la docencia y sus consecuencias. Por otro lado, también se explican conceptos como la corrección en base a rúbricas y las anotaciones web. Por último, se presenta la extensión Mark&Go y el objetivo del proyecto.

## 1.1. Evaluación continua

Desde hace unos años, con la implantación del plan Bolonia en el sistema educativo español, la Universidad del País Vasco (UPV/EHU) y, por lo tanto, también la Facultad de Informática (FISS) viene aplicando la evaluación continua [1] como sistema para evaluar la progresión de los estudiantes en la adquisición de conocimientos, habilidades y destrezas.

El método de evaluación continua es el método de evaluación más aplicado en la actualidad que consiste en realizar periódicamente numerosas pruebas, ejercicios, trabajos, exposiciones, etc. por parte de un estudiante en un intervalo de tiempo determinado con el que ir acumulando puntos para la nota final.

Esta metodología de evaluación tiene como finalidad la identificación de errores, problemas y deficiencias para su corrección. Por lo que estas pruebas periódicas son métodos perfectos para evidenciar cómo se está desarrollando el alumno en el proceso de su educación y consecuentemente diseñar actuaciones encaminadas a corregir errores, repasar o reforzar puntos débiles.

A continuación, se reproduce textualmente la descripción del ciclo de la evaluación continua obtenida de la Wikipedia.

*“Para que realmente exista una evaluación continua en la sala de clases es necesario que los profesores planifiquen sus unidades de aprendizaje, pero que al mismo tiempo la modifiquen clase a clase tomando en cuenta los aprendizajes obtenidos por los estudiantes. Para esto los docentes deben seguir los siguientes pasos de un ciclo de evaluación:*



Ilustración 1. Ciclo de la evaluación continua

1. *Definición de objetivo de aprendizaje: establecer un objetivo de aprendizaje tomando en cuenta los conocimientos previos de los estudiantes evidenciados en una evaluación diagnóstica previa.*
2. *Construcción de indicadores y criterios: estos indicadores y criterios deben guiarse por el objetivo de aprendizaje establecido anteriormente y deben ser observables. El nivel de logro de los indicadores establecerá el cumplimiento o no del objetivo de aprendizaje propuesto.*
3. *Construcción del instrumento y de la actividad de evaluación: evaluar, elegir y modificar tareas y textos en función de una meta de aprendizaje específica. El instrumento debe evaluar la actividad planificada y debe cumplir con los indicadores y criterios propuestos anteriormente.*
4. *Implementación de la clase: realizar la clase aplicando la actividad de evaluación diseñada para ella. De esta forma se evalúan los aprendizajes de la clase implementada y se generan evidencias observables del nivel de logro de los indicadores de evaluación propuestos.*
5. *Aplicación del instrumento y/o recolección de evidencias de aprendizaje: recoger las evidencias de aprendizaje de los estudiantes y aplicar el instrumento diseñado, el cual nos permitirá sistematizar el nivel de logro según indicador de los estudiantes.*
6. *Sistematización de las evidencias: sistematizar todas las evidencias recogidas en tablas y analizar los resultados.*
7. *Toma de decisiones: a partir de los resultados analizados y del nivel de logro de los indicadores alcanzado, tomar decisiones en relación a la planificación y adecuar los objetivos de aprendizaje según las necesidades observadas en las evidencias. “*

Este proyecto se centra en la construcción de indicadores y criterios (punto 2) que se hace mediante el criterio de rúbrica y también en la sistematización de las evidencias (punto 6).

## 1.2. Problemática

La inclusión de la evaluación continua también ha generado un incremento sustancial en la carga de trabajo del profesorado. Esto se debe a que la evaluación continua ha conllevado un incremento en el número de evaluaciones y del tiempo de su preparación y corrección, cuando antes solo había un examen final.

Este proyecto aborda el apartado de corrección. En este caso, se centra en la corrección de exámenes (código) con soporte digital, es decir, el alumno resuelve el ejercicio, lo guarda en formato texto o PDF y lo sube a la plataforma educativa (Egela). Para ello, se utiliza una corrección basada en rúbricas.

La rúbrica [2] es una herramienta de evaluación que permite conocer las competencias que ha adquirido el alumno en la tarea realizada. En ellas se especifican un conjunto de criterios y su grado de consecución, para conocer el mayor o menor dominio que el alumno tiene de los conocimientos que se pretendía que aprendiera. Normalmente una rúbrica se compone por los siguientes elementos:

- Competencias: Conceptos que el profesor quiere valorar en la tarea en concreto.
- Criterios: Cada uno de las competencias a valorar en una rúbrica se definen y especifican en diferentes criterios y estos se asocian para cada nivel de ejecución.
- Escala (grado) de calificaciones.

Para aclarar estos elementos, se aporta un ejemplo.

Escala de calificaciones

↓

Evaluation rubric	0	5	10	15
XML definition	Unable to define the xml without syntax errors	Minor errors in syntax and incomplete	Able to define the xml without errors but incomplete	Correctly defined XML and required data is inserted
XML namespaces	Non-defined namespaces	Defined but wrong attributes	Some error during testing xml	Schema location is defined and xml followed xsd
XSD definition	Xsd contains errors and is incomplete	Xsd is defined with some syntax errors	XSD is defined without errors but is incomplete	XSD is defined with the required specification
XSLT visualization	Xslt is not working	Xslt works but lack of more than 3 elements	Works but is not completed or xsd is not included	Xslt works perfectly with xsd tags included
Xquery	More than 4 queries are not working	Less than 4 queries are not working	Less than 2 queries are not working	All the queries work perfectly

Competencias → ← Criterios

*Ilustración 2. Ejemplo de criterios de rúbrica*

Este proceso de corrección es laborioso y en la medida en la que haya que corregir varias veces a lo largo de la asignatura se vuelve oneroso para el profesor. Para abordar este tema, el grupo Onekin está estudiando la utilización de técnicas de anotación web para la corrección en base a rúbricas.

### 1.3. Anotaciones web

Las anotaciones web [3] son observaciones, explicaciones, notas u otros tipos de comentarios externos que pueden adjuntarse a un sitio web o a una parte de un documento. Al ser externas, posibilita anotar cualquier página web de manera independiente, sin necesidad de editar esa página.

Actualmente, en la mayoría de navegadores no vienen integradas de manera nativa por lo que existen ciertas extensiones web para dichos navegadores que se encargan de darnos este servicio. Entre los más conocidos se encuentran: Annotary [4], Evernote Web Clipper [5], Diigo [6], A.nnotate [7] y Hypothes.is [8]. Este proyecto va a girar en torno a esta última extensión.

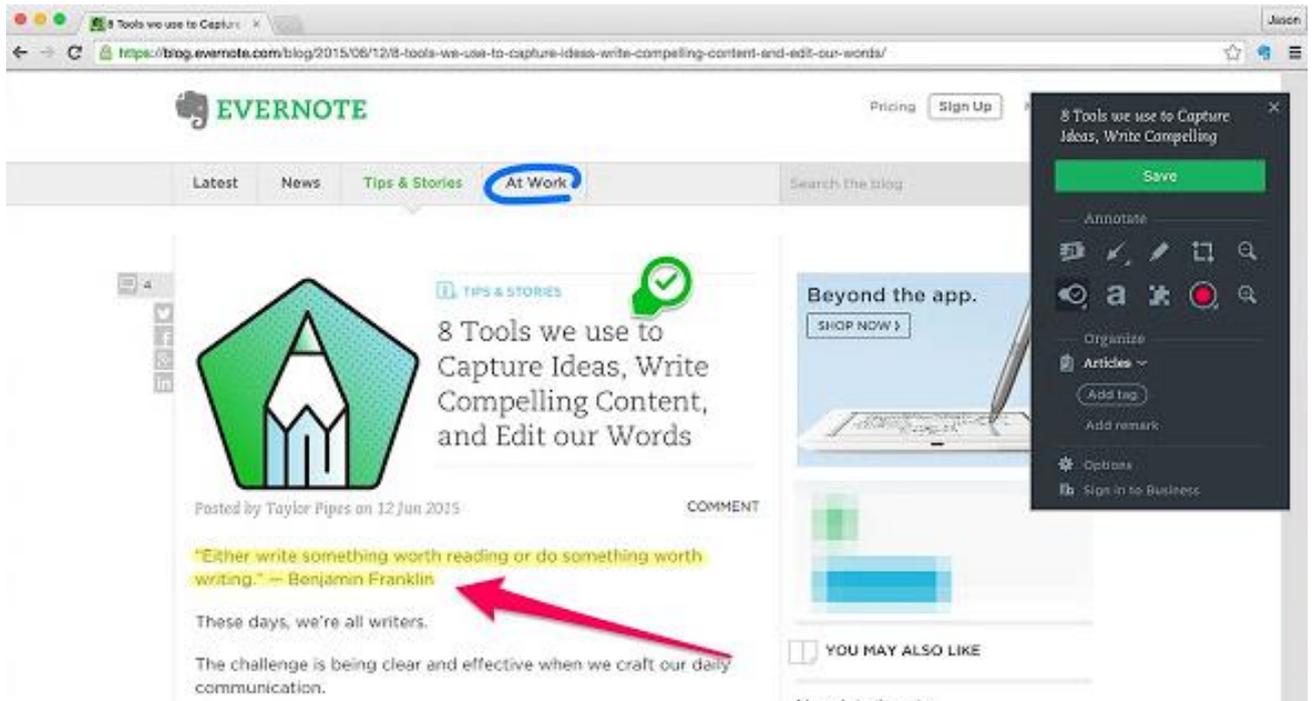


Ilustración 3. Anotación web

### 1.3.1. Hypothesis

La principal idea de *The Hypothesis Project* es implementar una capa de conversación en toda la web que funciona en todas partes, sin necesidad de implementación por parte de ninguna aplicación externa a la que tengamos que dirigirnos. Y es que el objetivo de Hypothes.is no es otro que el de crear una plataforma en la que sus usuarios, una vez se hayan registrado, puedan realizar anotaciones en las webs que visitan todo ello usando el estándar de anotaciones web W3C.

Hypothesis dispone de una extensión para Google Chrome que permite seleccionar y resaltar partes de texto de páginas webs, comentar, añadir etiquetas y compartir en un grupo privado o públicamente las anotaciones generadas sobre ellas.

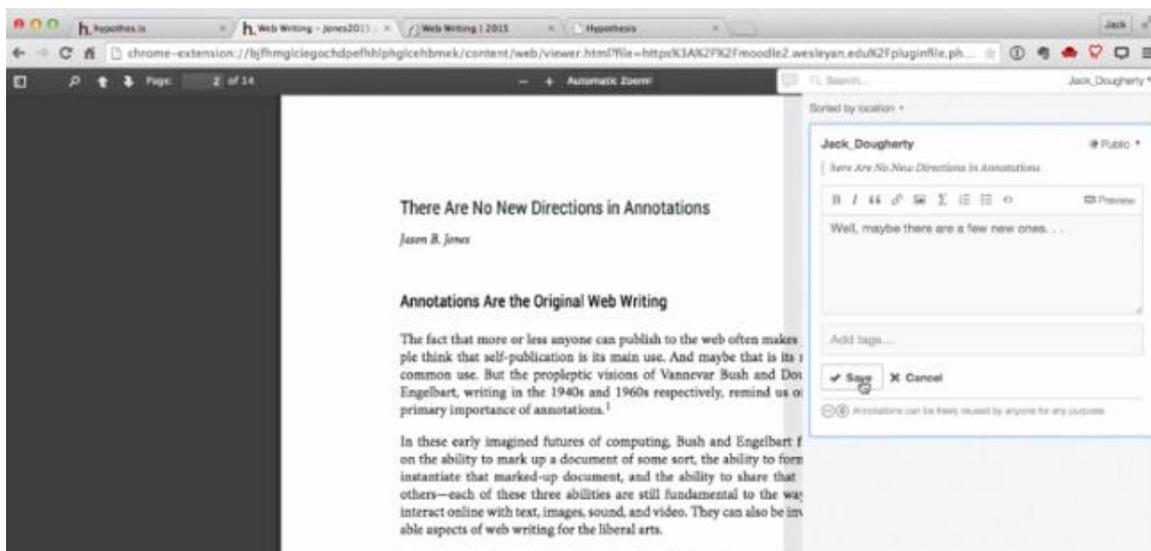


Ilustración 4. Extensión web de Hypothesis

### 1.3.2. Standard W3C para las anotaciones Web

World Wide Web Consortium [9], conocido también por las siglas W3C, es una comunidad internacional que desarrolla estándares web. Dirigido por el creador de la web, Tim Berners-Lee, tiene como finalidad la creación de una Web universal, accesible, fácil de usar y en la que todo el mundo pueda confiar.

El World Wide Web Consortium anunció en febrero del 2017 la publicación de tres nuevos estándares en cuanto a las anotaciones web:

- Web Annotation Data Model: Ofrece detalles sobre la estructura de las anotaciones (en formato JSON).
- Web Annotation Vocabulary: Proporciona el soporte semántico.
- Web Annotation Protocol: Proporciona un sencillo API RESTful HTTP para el intercambio de anotaciones entre clientes y servidores.

Gracias a los nuevos cambios anunciados por la W3C, estos les permitirán a los usuarios tener un control mayor sobre estas. Si este nuevo sistema tiene éxito, los navegadores permitirán a los usuarios ver y participar en conversaciones en cualquier página sin necesidad de intervención por parte de los creadores de dicha página.

### 1.4. Mark&Go

Este proyecto utiliza Mark&Go [10]. Mark&Go es una extensión web para Google Chrome desarrollado por el grupo Onekin que permite la corrección de los exámenes en base a rúbricas mediante anotaciones web haciendo uso de la API de Hypothesis. En el anexo C se puede encontrar el manual de usuario donde se explica de manera más detallada el funcionamiento de Mark&Go pero a continuación se hará un breve resumen de la herramienta. Para empezar a usar Mark&Go primero hay que definir las rúbricas.

#### 1.4.1. Definir rúbricas

La hoja de cálculo (Google Spreadsheet) es donde se definen las rúbricas y las notas se visualizan en formato tabular. En esta plantilla [11] se puede ver como se define una hoja de cálculo para poder empezar a corregir con Mark&Go. La hoja de cálculo debe cumplir con el siguiente formato:

- La primera columna representa al alumno
- El resto de columnas son las competencias a evaluar (Solo hasta la columna "Total")
- Para cada estudiante se debe proporcionar un enlace de la carpeta donde reside el examen. Por ejemplo,

<https://drive.google.com/drive/folders/17f034xn3otZw96JcdTKIE8gINL3j25d>

Para establecer los valores o criterios posibles, haciendo clic en el botón derecho en la segunda fila, se puede establecer cuál es el rango de valores para cada competencia.

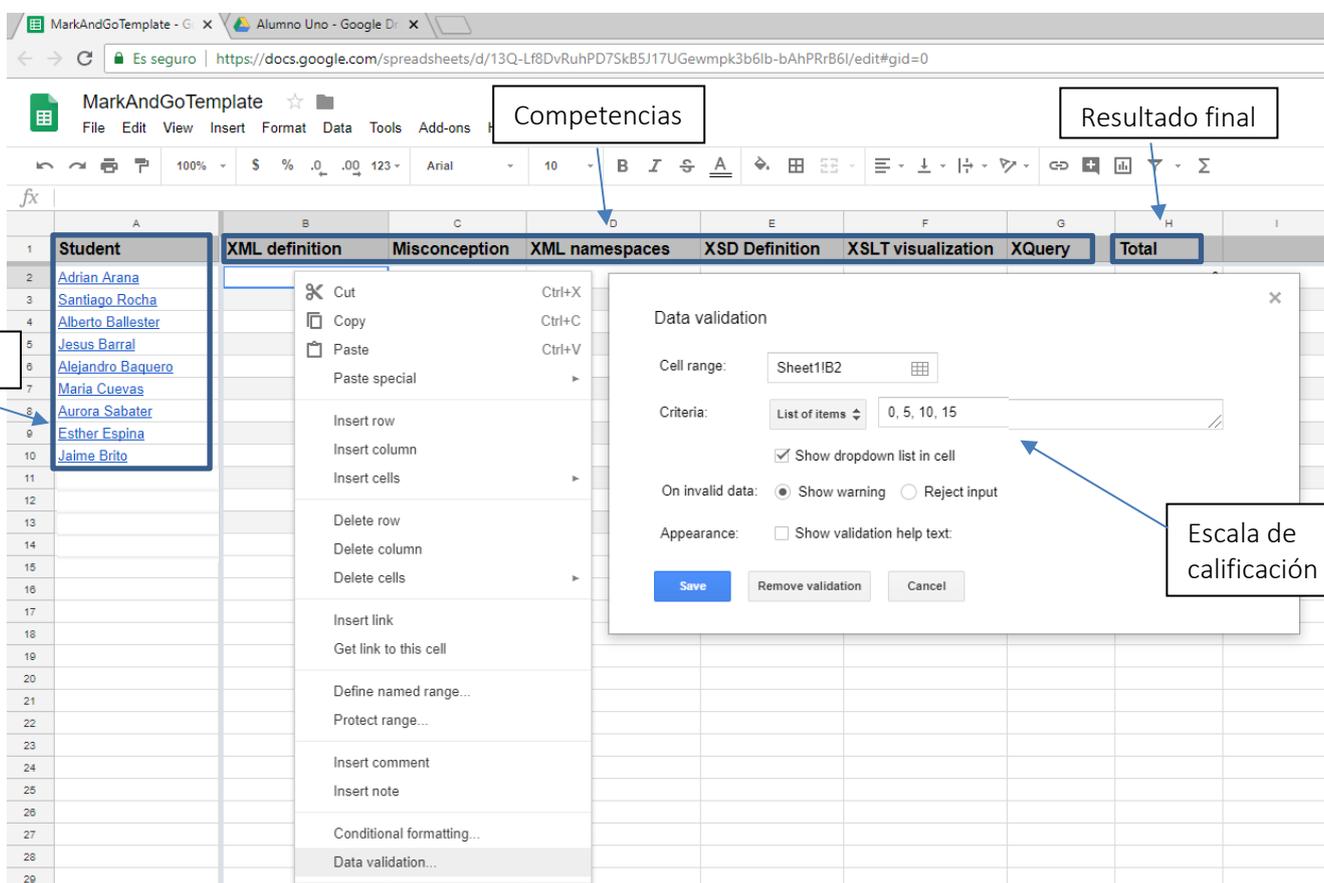


Ilustración 5. Establecer rango de valores

Después de finalizar la preparación de la hoja de cálculo, hay que hacer clic en el icono de la extensión en la parte superior derecha del navegador y la herramienta se configurará automáticamente para comenzar a corregir al mismo tiempo que genera un grupo de Hypothesis donde se guardaran las anotaciones que se hagan del examen. Además, en ese mismo momento

la extensión también genera unas anotaciones que corresponde cada una a cada una de las calificaciones posibles de cada pregunta que se guardan en dicho grupo. Mas adelante se explicará con más detalle.

#### 1.4.2 Corregir exámenes

Primero hay que abrir el examen del alumno en la web y después clicar en el icono de la extensión. El modo de evidencia (evidencing mode) se activará en la barra lateral. Haciendo clic en el interruptor se puede cambiar al modo de corrección (marking mode).

##### 1.4.2.1. Encontrar evidencias

Comenzando en modo de evidencia, hay que subrayar las evidencias para establecer más adelante la nota de los exámenes de los alumnos. Para anotar, hay que subrayar el texto que se usará como evidencia y la barra lateral se abrirá automáticamente para elegir cuál es la competencia que se está evaluando.

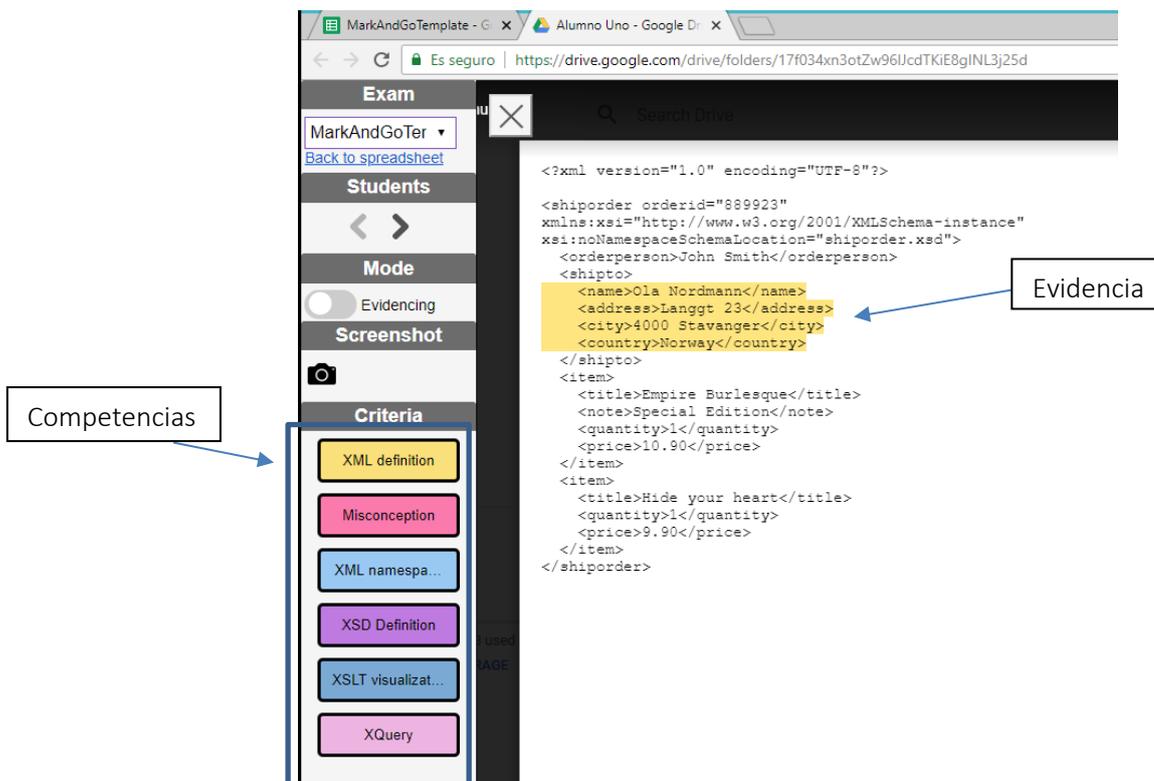


Ilustración 6. Identificar evidencias

#### 1.4.2.2. Corregir examen

Para corregir hay que pasar al modo de corrección. Desde aquí se puede calificar el examen seleccionando los botones de la barra lateral haciendo que la nota sea automáticamente trasladada a la hoja de cálculo al clicar en ella y a su vez se genera una anotación web que se guarda en Hypothesis.

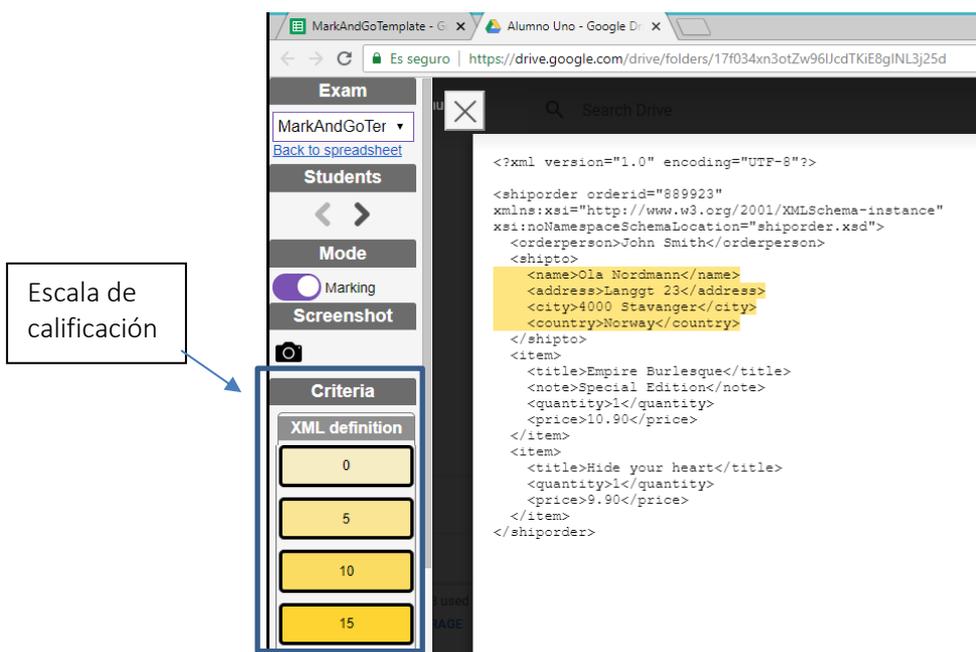


Ilustración 7. Corrección de un examen

Una anotación es un objeto formado con la siguiente información:

- La URI de la página donde está almacenado el examen.
- El grupo de Hypothes.is donde se va a guardar la anotación. Un grupo contiene las anotaciones de un examen en concreto.
- La fecha en la que se realizó la anotación.
- El contenido subrayado, es decir, la evidencia.
- La pregunta o competencia que se está evaluando.
- La calificación asignada.

Para una rúbrica de 5 competencias, se generan al menos 5 anotaciones por alumno ya que una anotación corresponde a una competencia, pero puede darse el caso de que para una competencia sea necesaria más de una anotación ya que pueden existir varias evidencias a la hora de puntuarla. En este caso, estas anotaciones tendrían las mismas etiquetas. Supóngase la siguiente rúbrica.

Evaluation rubric	0	5	10	15
<b>XML definition</b>	Unable to define the xml without syntax errors	Minor errors in syntax and incomplete	Able to define the xml without errors but incomplete	Correctly defined XML and required data is inserted
<b>XML namespaces</b>	Non-defined namespaces	Defined but wrong attributes	Some error during testing xml	Schema location is defined and xml followed xsd
<b>XSD definition</b>	Xsd contains errors and is incomplete	Xsd is defined with some syntax errors	XSD is defined without errors but is incomplete	XSD is defined with the required specification
<b>XSLT visualization</b>	Xslt is not working	Xslt works but lack of more than 3 elements	Works but is not completed or xsd is not included	Xslt works perfectly with xsd tags included
<b>Xquery</b>	More than 4 queries are not working	Less than 4 queries are not working	Less than 2 queries are not working	All the queries work perfectly

*Ilustración 8. Ejemplo de criterios de rúbrica*

A continuación, un ejemplo de la estructura JSON de las anotaciones que se guardan en Hypothesis. En este caso, corresponde al del ejemplo de las ilustraciones 6 y 7.

```
{
  "group": "JY658rmX",
  "target": "<name>Øla Nordmann</name><address>Langgt 23</address><city>
>4000 Stavanger</city><country>Norway</country>",
  "tags": [
    "exam:isCriteriaOf:XML definition",
    "exam:mark:15"
  ],
  "text": "",
  "created": "2018-05-25T17:31:17.379162+00:00",
  "uri": "https://drive.google.com/drive/folders
/159xx2fEyzhE2t1U5yXhzC6zl0UwofClF",
  "user": "acct:Oscar@hypothes.is",
  "id": "bkkSFGBBEeiDkgs2dv47dQ"
}
```

*Ilustración 9. Ejemplo de estructura JSON de una anotación web*

A pesar de que la mayoría de las anotaciones que se guardan en un grupo de examen de Hypothesis son las relativas a correcciones de exámenes de alumnos, como el ejemplo de la ilustración 9, también existen otras pues nada más crear un grupo de Hypothesis se crea una anotación por cada calificación posible de cada pregunta que se han establecido en la hoja de cálculo. Estas anotaciones únicamente guardan dos etiquetas siendo la primera la nota y la segunda la pregunta (orden inverso para diferenciarlos). Esta información se utiliza para que la herramienta conozca las preguntas y la escala de calificaciones a la hora de corregir, y para saber el valor máximo de cada pregunta y el valor del examen necesario en alguna de las visualizaciones. Para el resto de operaciones hay que filtrar las anotaciones para deshacerse de estas. Tomando como ejemplo la rúbrica de la ilustración 8 y la hoja de cálculo de la ilustración 5, la siguiente sería una de las anotaciones que se generarían automáticamente.

```

{
  "group": "JY658rmx",
  "target": "",
  "tags": [
    "exam:mark:0",
    "exam:isCriteriaOf:XML namespaces"
  ],
  "text": "",
  "created": "2018-05-25T16:54:42.991876+00:00",
  "uri": "https://hypothes.is/groups/rx4XNKMw/exam-4",
  "user": "acct:Oscar@hypothes.is",
  "id": "VVGthrDpEeifIF9xmEOubA"
}

```

*Ilustración 10. Ejemplo de anotación web de los criterios de rúbrica generados automáticamente al principio*

De igual manera que el ejemplo, se crearían otras tres anotaciones para la pregunta “XML namespaces” con la etiqueta con el prefijo “exam:mark” con las notas 5, 10 y 15. También se repetiría el mismo proceso con las 4 preguntas restantes. Las anotaciones restantes del grupo corresponden a las correcciones del examen como el ejemplo de la ilustración 9.

## 1.5. Objetivo

El objetivo del proyecto es la visualización de anotaciones web en el contexto de la corrección basada en rúbricas. Para ello, se ha realizado una extensión web que consulta las anotaciones almacenadas y dinámicamente genera una visualización. El diseño de esta visualización viene guiado por unas necesidades de información que el profesor quiere contestar. Este proyecto trabaja sobre las anotaciones realizadas con Mark&Go. Esta herramienta permite realizar anotaciones, pero no analizarlas, por ello, el objetivo es analizar anotaciones web en el contexto de la corrección de exámenes en base a rúbricas.

### 1.5.1 Visualización de datos

Uno de los sectores que más está evolucionando en el software científico es la visualización de datos [12]. En el análisis de datos, la visualización se ha convertido en un aspecto clave pues ayuda a detectar errores o confirmar hipótesis.

La visualización de datos no es más que la representación de los datos en un formato gráfico que permite comprender y transmitir mejor y más rápidamente la información; así se puede identificar patrones, comprender relaciones y detectar dependencias para conseguir conclusiones con mayor ventaja que, tal vez, no se estaría teniendo en cuenta en una presentación como en una hoja de cálculo.



## 2. Planificación

---

En este capítulo se presentan los aspectos correspondientes a la gestión del proyecto. En primer lugar, se expone el alcance del proyecto junto con las exclusiones. En segundo lugar, se presenta un diagrama EDT que muestra los paquetes de trabajo en los que se ha descompuesto el proyecto. En tercer lugar, se presenta el diagrama de Gantt, para exponer el tiempo de dedicación previsto a cada paquete de trabajo. En cuarto lugar, se describe el sistema de información y las herramientas. En quinto lugar, se analizan los riesgos que pueden aparecer durante el desarrollo del proyecto junto a un plan de contingencia para evitar alterar dicho desarrollo. Por último, una evaluación económica del proyecto.

## 2.1. Alcance

El objetivo del proyecto es la visualización de anotaciones web en el contexto de la corrección basada en rúbricas. Para ello, se va a realizar una extensión web para Google Chrome que consulta las anotaciones almacenadas y dinámicamente genera una visualización. El diseño de esta visualización viene guiado por unas necesidades de información que el profesor quiere contestar.

### 2.1.1 Exclusión

Debido al coste adicional que supone su desarrollo, se excluyen del proyecto las siguientes funcionalidades o características de la extensión:

- Integración de la extensión en Mark&Go.
- Traducción de la extensión a otros idiomas.
- Asegurar la correcta visualización de la extensión web en otros navegadores distintos a Google Chrome.

## 2.2. EDT

La estructura de descomposición del trabajo (EDT) es una herramienta que consiste en la descomposición jerárquica del proyecto para cumplir con los objetivos del mismo. El EDT del proyecto a desarrollar se divide en 7 secciones en las que se agrupan los paquetes de trabajo:

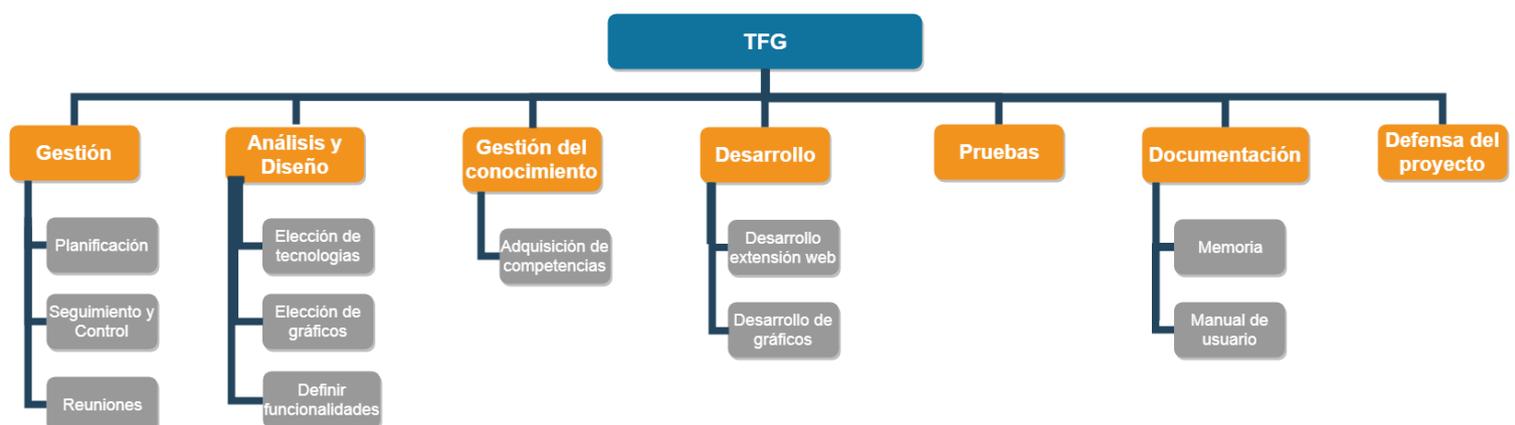


Ilustración 11. EDT del proyecto

1. **Gestión:** Esta sección incluye los paquetes de trabajo relacionados con la gestión del proyecto, entre los que se incluyen la planificación del proyecto, el seguimiento y control, y las reuniones.
2. **Análisis y diseño:** Este sección contiene las tareas correspondientes a la elección de tecnologías, las funcionalidades de la extensión web y la elección de gráficos.
3. **Gestión del conocimiento:** El paquete de trabajo de Adquisiciones de Competencias contiene todas las tareas necesarias para la adquisición de competencias necesarias para el desarrollo de la extensión web.
4. **Desarrollo:** El desarrollo agrupará todas las tareas que se llevarán a cabo para el correcto desarrollo del sitio web.
5. **Pruebas:** Esta sección contiene los paquetes de trabajo relacionados con las pruebas para comprobar el correcto funcionamiento de la extensión.
6. **Documentación:** Esta sección contiene los paquetes de trabajo relacionados con la documentación del proyecto. En este caso, la memoria del proyecto y el manual de usuario.
7. **Defensa del proyecto:** Engloba los paquetes de trabajo tanto de la propia defensa como de la preparación de la exposición.

### 2.3. Estimación de dedicación

En este apartado se estimará cuántas horas durará el proyecto en función de la duración estimada de cada tarea.

Tareas	Estimación en horas
<b>Gestión</b>	<b>36</b>
Planificación	15
Seguimiento y control	15
Reuniones	6
<b>Análisis y diseño</b>	<b>30</b>
Elección de tecnologías	10
Elección de gráficos	10
Definir funcionalidades	10
<b>Gestión del conocimiento</b>	<b>30</b>
<b>Desarrollo</b>	<b>100</b>
<b>Documentación</b>	<b>77</b>
Memoria	75
Manual	2
<b>Pruebas</b>	<b>10</b>
<b>Defensa del proyecto</b>	<b>25</b>
<b>TOTAL</b>	<b>308</b>

Tabla 1. Dedicación estimada

## 2.4. Diagrama de Gantt

En este apartado se presenta el diagrama de Gantt inicial. En él, se puede apreciar la planificación organizada a través de las semanas y obtener una visión global de la organización del proyecto. El diagrama se ha dividido en dos imágenes para facilitar su visionado.

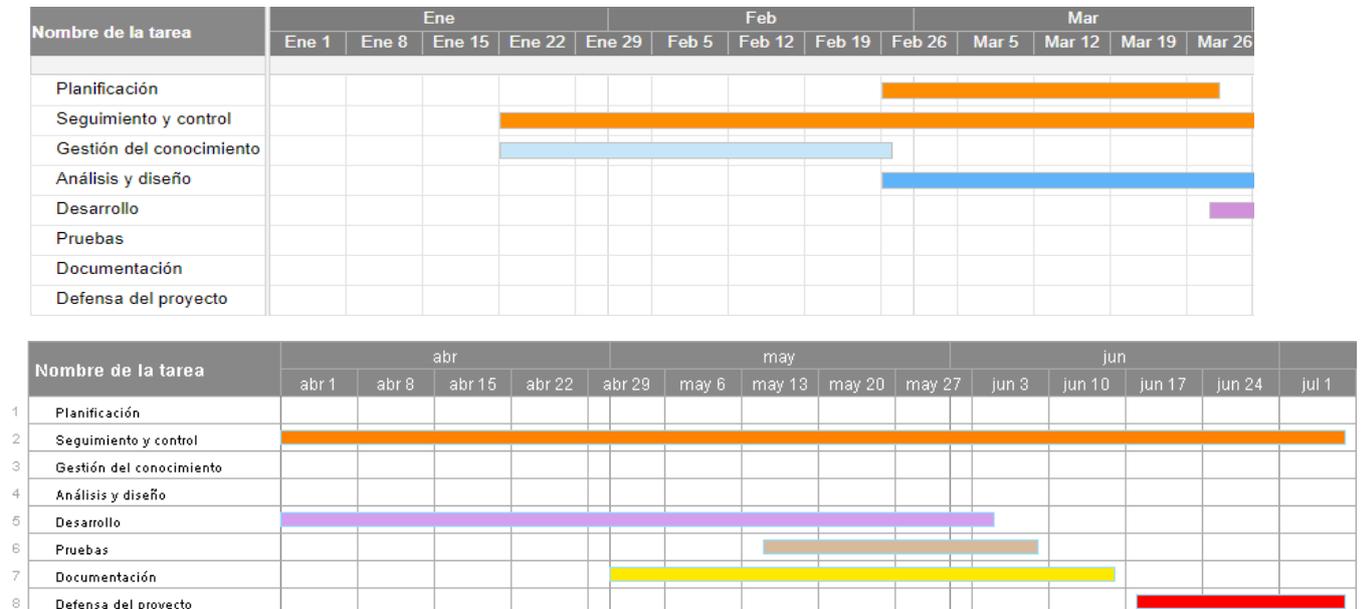


Ilustración 12. Diagrama de Gantt

## 2.5. Recursos humanos

En este apartado se presentan los miembros del equipo y sus responsabilidades.

- Director: Oscar Díaz
- Ayudante del director: Haritz Medina Camacho
- Encargado de realizar el proyecto: Eduardo Pérez Capellán

## 2.6. Caracterización del sistema de información y del sistema de comunicaciones en el proyecto

### 2.6.1. Sistema de Información

El sistema de información del proyecto estará formado por 3 subsistemas:

- S1: Residente en el equipo del alumno
- S2: Repositorio en GitHub
- S3: Carpeta en Google Drive

Se creará un subsistema residente en el equipo personal del alumno donde se mantendrá una copia de seguridad “offline” del proyecto separada de la nube y otro residente en la misma nube. Además, el código de la extensión se ira guardando en un repositorio de GitHub.

Los subsistemas estarán compuestos por una carpeta raíz denominada TFG, la cual contendrá en su interior un documento compartido con el director y tres carpetas principales: Actas, Proyecto y Gestión. En Actas se guardan todo lo relacionado a las actas y las reuniones en sí. La carpeta de Gestión contendrá la planificación, una hoja de cálculo que contiene las dedicaciones y un documento a modo de diario donde se van añadiendo por cada día las tareas realizadas. La carpeta Proyecto a su vez, estará dividida en otras tres carpetas: Extensión, Documentación y Copias de Seguridad. De esta manera, la información estará mejor organizada y mucho más localizable. En cuanto al documento restante, todos los miembros del equipo (El director y el ayudante) tendrán acceso al mismo y servirá para ir añadiendo dudas, avances, enlaces y las actas.

#### 2.6.2. Formatos

- Los documentos de texto editables se realizan con Microsoft Office 2016, por lo tanto, tendrán un formato “.docx”.
- Los documentos entregables tienen el formato “.pdf”.
- Los documentos tienen un formato común cuyas características generales están en las siguientes plantillas:
  - [Memoria TFG](#)

#### 6.6.3. Copias de seguridad

Las copias de seguridad se realizan en el propio ordenador y en Google Drive, por lo que, si el ordenador dejase de funcionar o Google Drive decidiera dejar de dar acceso, se podría recuperar el proyecto de una manera u otra.

#### 6.6.4. Comunicaciones

- Las reuniones se han previsto realizarlas en las fechas y horarios acordados por el tutor del proyecto y el alumno. A medida de lo posible cada dos semanas.
- Las comunicaciones con el tutor y el ayudante se realizarán vía correo electrónico.

## 2.7. Herramientas

Para el desarrollo del proyecto se usarán las siguientes herramientas.

### 2.7.1. WebStorm

JetBrains Webstorm es un IDE profesional desarrollado por JetBrains que es compatible con una amplia gama de tecnologías modernas relacionadas con el lenguaje de programación JavaScript, HTML y CCS. WebStorm se va a utilizar para el desarrollo de la extensión web.

### 2.7.2. Codepen

Codepen es una herramienta online de prototipado rápido y prueba de fragmentos de código. Se encuentra dividida en secciones de CSS, HTML, JavaScript y la salida por lo que hace que sea muy sencillo visualizar los resultados del código de manera inmediata. El principal uso será hacer pruebas antes de pasar el código a WebStorm.

### 2.7.3. Postman

Postman está compuesto por diferentes herramientas y utilidades gratuitas que permiten hacer peticiones (GET, POST, etc.) a APIs y generar colecciones de peticiones que nos permitan probarlas de una manera rápida y sencilla. Esta herramienta se utilizará para trabajar con la API de Hypothesis.

### 2.7.4. Microsoft Office Word

Microsoft Word es un programa informático orientado al procesamiento de textos creado por la empresa Microsoft. La versión del programa en este proyecto es el Microsoft Office Word 2016 para Windows y se utilizará para realizar la memoria y documentación.

### 2.7.5. Microsoft Excel

Microsoft Excel es una aplicación de hojas de cálculo que forma parte de la suite de oficina Microsoft Office. La versión del programa en este proyecto es el Microsoft Office Excel 2016 para Windows y se utilizará para contabilizar la dedicación del proyecto.

### 2.7.6. GitHub

GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git. El objetivo es usar GitHub para ir compartiendo y guardando continuamente el código de la extensión.

#### 2.7.7. Google Docs

Ofrece la posibilidad de crear documentos en línea editar los que ya teníamos o compartirlos en la red con otros usuarios. Se ha creado un documento compartido con el director y el ayudante del proyecto para ir actualizando el progreso, dudas, compartir enlaces e información, y las actas de reuniones.

#### 6.7.8. Google Drive

Google Drive es un servicio de alojamiento de archivos multiplataforma en la nube. La principal función de esta será guardar las copias de seguridad.

#### 2.7.9. SmartSheet

SmartSheet es una herramienta online para la gestión de proyectos. Se ha seleccionado esta herramienta porque ya se había utilizado en anteriores proyectos y por la variedad de opciones que ofrece en cuanto a creación de cronogramas como añadir tareas, periodos de realización, etc.

#### 2.7.10. Draw.io

Draw.io es una aplicación de Google Drive que le permite dibujar diagramas de todo tipo. Se utilizará para crear el EDT.

#### 2.7.11. StarUML

StarUM es un software de modelado basado en los estándares UML (Unified Modeling Language) y que proporciona una gran variedad de tipos de diagrama, entre los que se encuentran los diagramas de secuencia y de casos de uso.

### 2.8. Riesgos

Durante la planificación del proyecto, se han identificado diferentes riesgos que podrían afectar al desarrollo del proyecto, además de la propia incertidumbre. Para cada uno de ellos se presenta un posible plan de mitigación a aplicar en caso de que éste se diera, con el fin de reducir su impacto. A continuación, se listan los riesgos identificados:

1. Problemas personales: Problemas de salud pueden hacer que el desarrollo se retrase debido a la pausa necesaria para la recuperación del alumno. También el tener que realizar un viaje por motivos personales pueden causar una pausa.
2. Cambios en las funcionalidades: Es posible que durante el desarrollo de la extensión surjan mejoras y nuevas ideas cuya importancia y desarrollo conlleven un retraso sobre la planificación inicial.
3. Realizar una mala planificación. Se deben tener en cuenta la posibilidad de que ocurran hechos inicialmente no previstos que afecten negativamente al proyecto. Una

planificación incorrecta puede suponer no cumplir bien los plazos y producir retrasos en las entregas.

4. Incumplimiento de los plazos previstos: Si no se cumpliesen los hitos marcados sería un gran riesgo debido a que los hitos están muy cerca de las fechas límite.
5. Avería en la máquina de trabajo del alumno: Podría afectar negativamente al desarrollo del proyecto y a la propia planificación en caso de no disponer de alternativas y de estar realizando una tarea de la que dependen muchas otras aún sin empezar y que requieren de su finalización.
6. Asignaturas de cuarto curso: Suspender alguna asignatura reduciría el tiempo disponible para dedicar al proyecto.
7. Falta de conocimiento sobre las tecnologías utilizadas: Varias tecnologías, nuevas para el alumno, pueden retrasar el progreso debido a la necesidad de estudiar y corregir errores en su utilización y puesta en marcha.

	Impacto				
Probabilidad	Muy bajo	bajo	Medio	Alto	Muy alto
Muy baja					5
baja			6	1,2,3,4,6	
Media				7	
Alta					
Muy alta					

*Tabla 2. Riesgos del proyecto*

### 2.8.1. Plan de mitigación

1. Problemas personales: En caso de no poder continuar con el desarrollo del proyecto debido a problemas personales, el desarrollo se retrasará hasta que los problemas se hayan superado y el alumno esté en condiciones de retomar el trabajo.
2. Cambios en las funcionalidades de la aplicación: Hay que fijar el alcance y requerimientos del proyecto lo antes y mejor posible con el fin de evitar grandes modificaciones en el alcance del proyecto.
3. Fallos en la planificación: Realizar un reajuste rápido de la planificación de forma que pueda corregirse el fallo el mismo día que fue detectado.
4. Incumplimiento de los plazos previstos: Apresurarse en la finalización de dichas tareas, puesto que las fechas de entrega son muy próximas a las marcadas como de finalización.
5. Fallo en la máquina de trabajo: Una copia de seguridad tanto en Drive como en GitHub solucionaría el acceso a la información.
6. Asignaturas de cuarto curso: Se replanificará el trabajo y una vez terminados los exámenes la dedicación al proyecto será total.

## 2.9. Evaluación económica

A la hora de realizar un proyecto, es importante ver el impacto económico que este supondrá y estimar lo que va a costar su desarrollo. Al tratarse de un proyecto desarrollado para el grupo de investigación Onekin de la UPV/EHU se tratará de una aplicación gratuita por la cual no se percibirá ningún aporte económico.

En el apartado de hardware no se espera ningún gasto ya que se hará uso únicamente del ordenador personal.

En cuanto al software tampoco hay gastos ya que:

- **WebStorm:** Es un producto de pago, pero gracias a la licencia de estudiante se puede obtener gratuitamente.
- **Codepen:** Cuenta con dos versiones, una gratuita y otra de pago con algunas opciones extra. Se hará uso de la versión gratuita pues cumple con las necesidades.
- **Microsoft Office:** Software de pago que ya estaba adquirido.
- **GitHub:** Es completamente gratis e ilimitado para proyectos públicos.
- **Google Drive y Google Docs:** Ambas son gratuitas.
- **Draw.io:** Es una aplicación completamente gratuita.
- **SmartSheet:** Es una herramienta de pago, pero ha sido posible elaborar el cronograma gracias a la prueba gratuita de 30 días que ofrece.
- **StarUML:** Se hará uso de la versión gratuita.
- **Postman:** Ofrece diversos planes entre los que se encuentra la gratuita que es suficiente para este proyecto.

El único coste serán 5 dólares que cuesta subir una extensión al Chrome Web Store.



# 3. Análisis de evidencias de evaluación recogidas mediante anotaciones web

---

En este capítulo se mostrarán, por un lado, las necesidades de información que necesita contestar el profesor y, por otro lado, la elección del tipo de visualización.

### 3.1 Evidencias

Entendemos por evidencia, el subrayado realizado por un profesor de cara a la valoración de una determinada competencia. Estas evidencias se pueden disponer de forma tabular: las filas corresponden a los alumnos mientras que las columnas representan las competencias. Una celda (competencia, alumno) recogería las evidencias (párrafos del examen junto con la calificación).

### 3.2. Necesidades de análisis sobre evidencias

#### 3.2.1. Requisitos de análisis de datos

Una parte esencial del proyecto es buscar esas necesidades de información que necesita resolver el profesor. A continuación, se ha realizado una lista con una serie de posibles preguntas que se podría plantear un docente sobre un examen o la asignatura.

P1	¿Existe alguna relación entre dos preguntas de un examen?
P2	¿Cuál es la pregunta más complicada?
P3	¿Cuál es la pregunta más fácil?
P4	¿Qué competencias no han adquirido los alumnos en la asignatura?
P5	¿Existe alguna relación entre el resultado de una pregunta y el resultado final del examen?
P6	¿El resultado de un examen tiene alguna relación con el resto de exámenes?
P7	¿Cuál es la tendencia de las notas a lo largo de la evaluación continua?
P8	¿Qué tal son los resultados de un alumno en comparación con la media de la clase?
P9	¿Cuáles son los ejercicios más complicados para un alumno en comparación con el resto de la clase?

*Tabla 3. Preguntas*

A raíz de estas preguntas, se hizo un estudio para encontrar las mejores visualizaciones que pudieran ayudar a contestar estas preguntas.

#### 3.2.2 Elección de gráficos

Hay infinidad de maneras de representar los datos. Tan importante es representarlos, como saber de qué manera hacerlo para conseguir una mejor comprensión de la información. Para ello, tras analizar diferentes propuestas que pudieran satisfacer las necesidades del profesor, se decidió que las siguientes visualizaciones eran las mejores opciones.

### 3.2.2.1. Diagrama alluvial

Estos diagramas, también llamados Sankey [13], muestran los flujos y sus cantidades en proporción entre sí. El ancho de las líneas se utiliza para mostrar sus magnitudes, por lo tanto, cuanto mayor sea la línea, mayor será la cantidad de flujo. Las líneas de flujo (enlaces) pueden combinarse o dividirse a través de sus trayectorias en cada etapa de un proceso (nodos). El color se puede utilizar para dividir el diagrama en diferentes categorías. El ejemplo que se muestra a continuación [14], trata sobre los pasajeros que iban en el Titanic. En ella se pueden ver las relaciones entre la clase de los pasajeros, el sexo, la edad y, finalmente, si sobrevivieron al hundimiento o no. Los colores de los enlaces o flujos durante todo el proceso indican si sobrevivieron o no. Por ejemplo, todas las mujeres que iban en primera clase sobrevivieron.

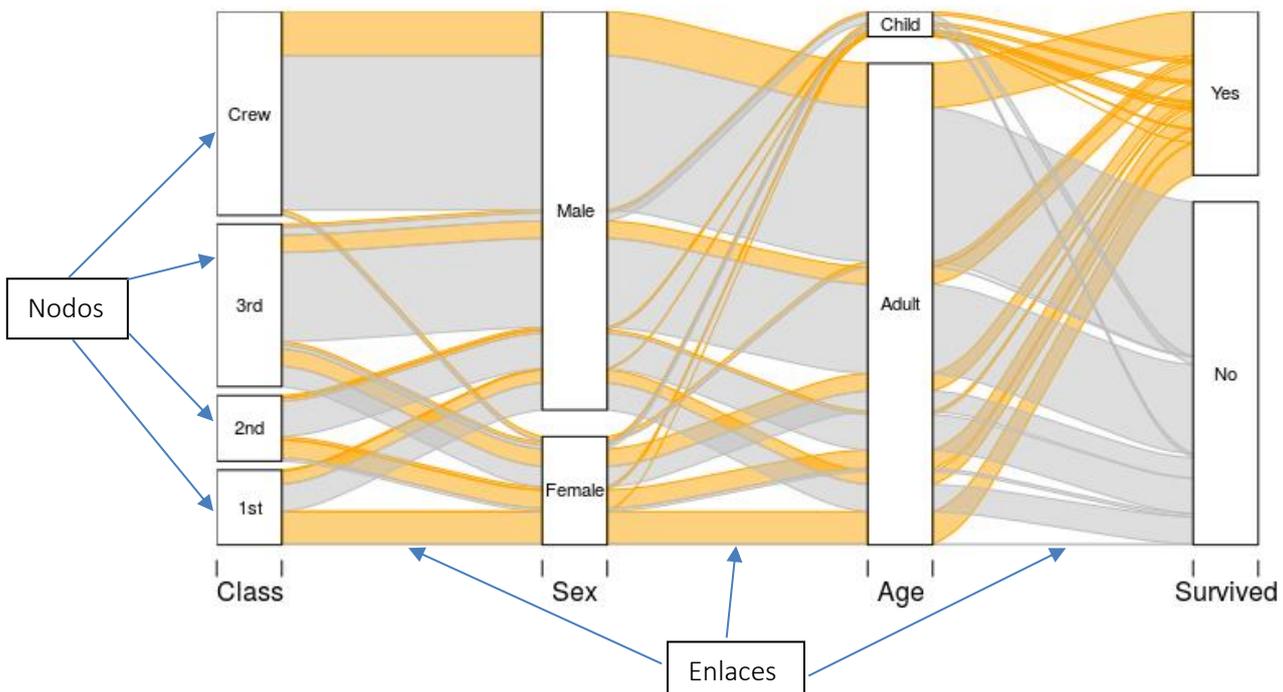


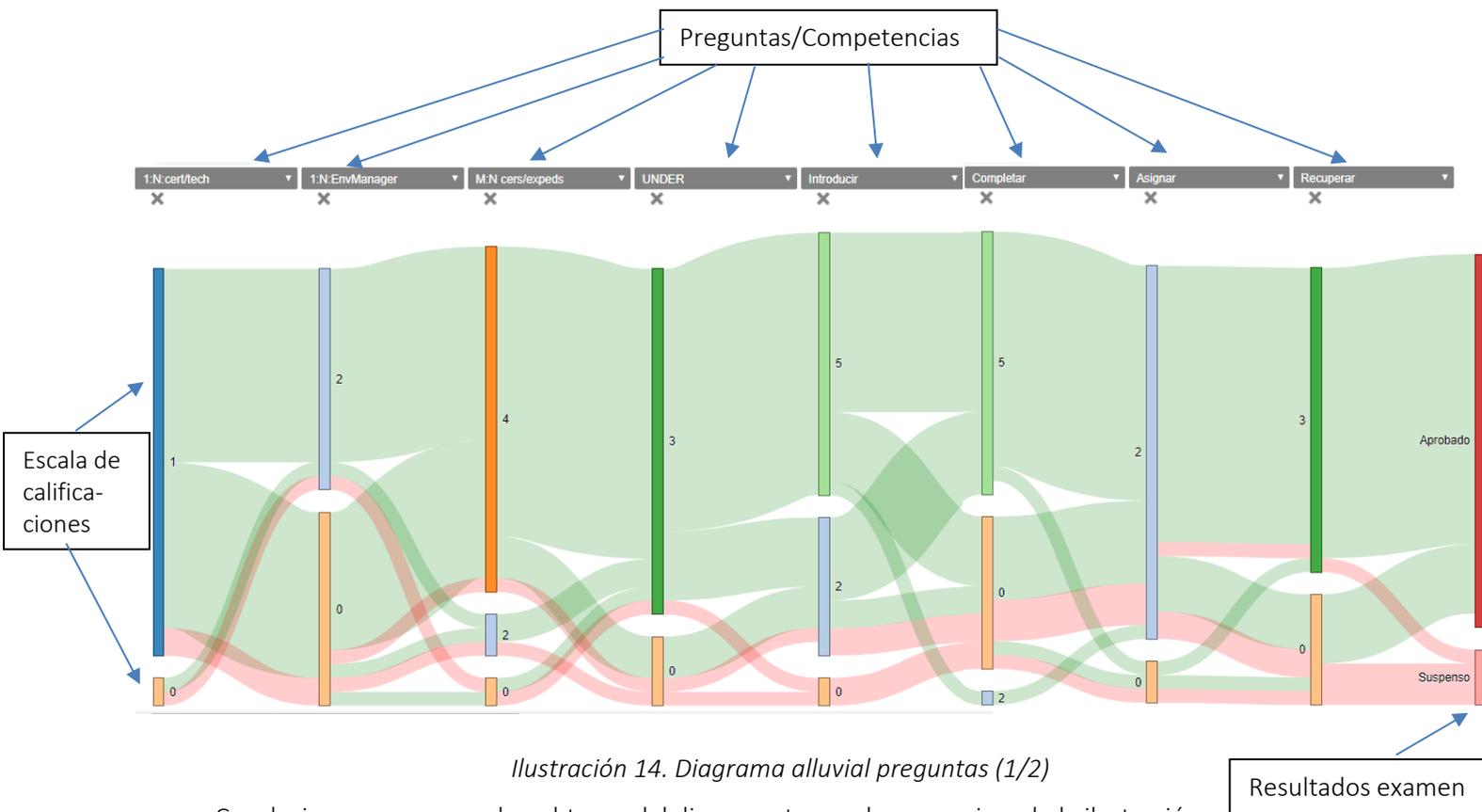
Ilustración 13. Ejemplo de diagrama alluvial

### Diagrama alluvial en el proyecto

El diagrama alluvial de este proyecto se divide en dos partes:

- En la primera parte, las líneas (enlaces) representan la relación entre los resultados de las preguntas (nodos), la anchura el número de alumnos y el color informa si finalmente aprobaron. Con ello se puede ver cuántos alumnos que han sacado una determinada nota en una pregunta, que notas han obtenidos en la siguiente pregunta y si aprobaron finalmente o no el examen. Puede resultar muy útil para ver relaciones entre preguntas y cuáles fueron las más complicadas y fáciles.

- En la segunda parte, el diagrama permite ver las relaciones entre los resultados del resto de exámenes de la evaluación continua.



Conclusiones que se pueden obtener del diagrama tomando como ejemplo la ilustración.

- Preguntas muy fáciles (> 90). Por ejemplo: “1:N cert/tech”.
- Preguntas muy difíciles (< 10). Por ejemplo: “1:N cert/tech”.
- Dependencias “de flujos” entre preguntas, es decir, si saben bien la pregunta A, saben bien la B, o al revés. En el ejemplo, se producen flujos “bifidos” entre las dos primeras preguntas, mientras que las preguntas 3 y 4 son “lineales”.
- Para los alumnos que aprueban, preguntas que no contestan correctamente, es decir, aprueban la asignatura sin saber estos conceptos. En el ejemplo, “1:N EnvManager” e “Introducir” ilustran esta situación.

En el mismo diagrama también se puede ver la relación con el resto de exámenes (“18.2 XQuery” y “prueba”) como se puede ver en la siguiente ilustración.

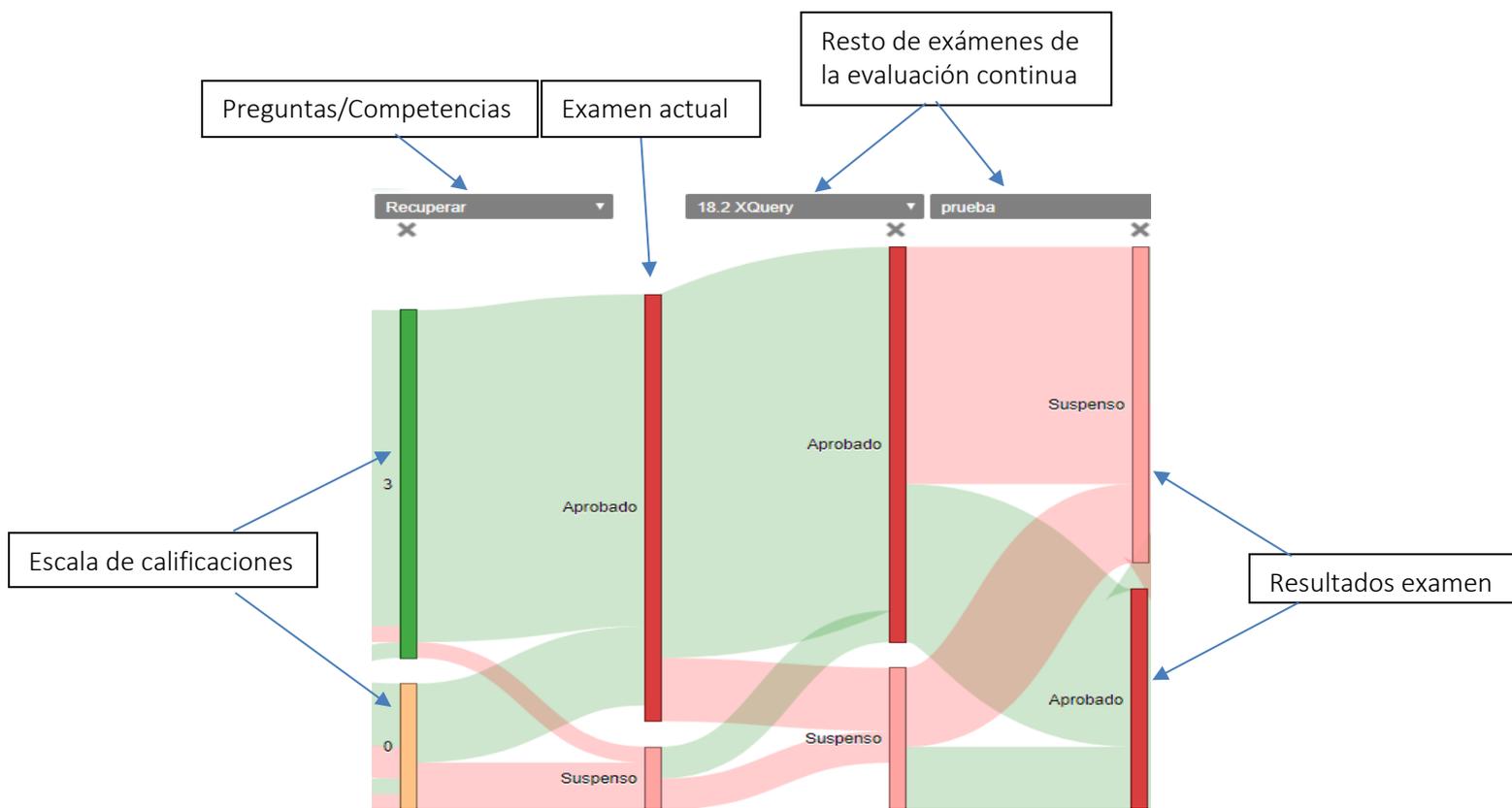


Ilustración 15. Digrama alluvial exámenes (2/2)

De la anterior ilustración se puede obtener, por ejemplo, las siguientes conclusiones:

- La gran mayoría de los que han aprobado el examen actual, aprobaron también el examen “18.2 XQuery”, mientras que la mayoría de los que aprobaron “18.2 XQuery”, suspendieron el último examen.
- En el examen actual se han presentado menos alumnos que en “18.2 XQuery”.

### 3.2.2.2. Diagrama de araña

El gráfico de araña [15] es una representación gráfica en él que se proyectan de manera visual y atractiva diversos datos, atributos y cualidades. Este tipo de gráfico se utiliza, principalmente, como herramienta útil de comparación de datos para ver qué variables tienen valores similares o si hay valores atípicos.

Por ejemplo, en el siguiente caso [16] podemos ver el porcentaje de usuarios tanto de tablets como de smartphones que hacen uso de una serie de servicios. El 60% de los usuarios usan su smartphone para mirar el email frente al 48% que lo hacen desde su tablet.

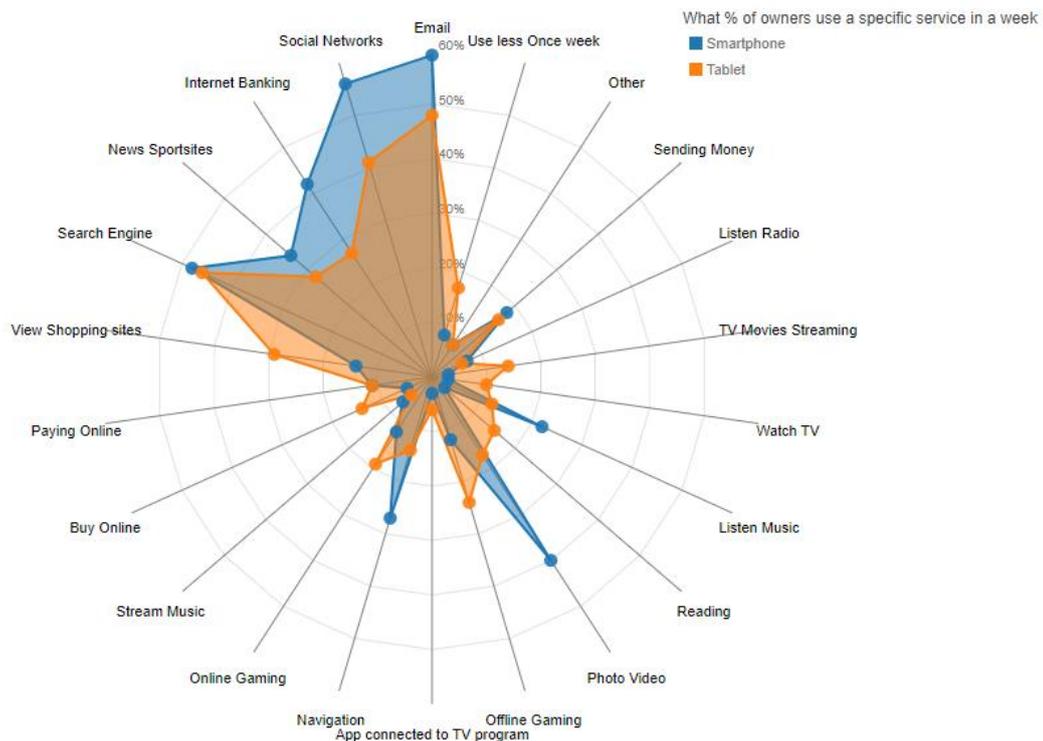


Ilustración 16. Ejemplo de diagrama de araña

### Diagrama de araña en el proyecto

Para este proyecto, estos gráficos son efectivos para comparar los resultados de un alumno con la media de la clase, siendo ideales para mostrar el rendimiento. Tomando como ejemplo la siguiente ilustración, el polígono azul es el relativo al alumno mientras que el rojo refleja la media de la clase siendo los ejes las preguntas de los exámenes. En primera instancia, el diagrama que se visualiza está formado con los datos (preguntas) de todos los exámenes de evaluación continua de la asignatura. Visualmente es fácil obtener conclusiones mediante este gráfico como en el siguiente ejemplo:

- El alumno en 9 de las 13 preguntas ha sacado la máxima nota mientras que ha sacado una vez la mínima nota, en este caso, en la pregunta 'XSQL 2'.
- El alumno ha estado encima de la media en 9 ocasiones.
- El alumno ha estado en 2 preguntas por debajo del 50%: 'XSQL 1' y 'XSQL 2'.
- La media de la clase solo en la pregunta 'XSQL 2' ha estado por debajo del 50%.

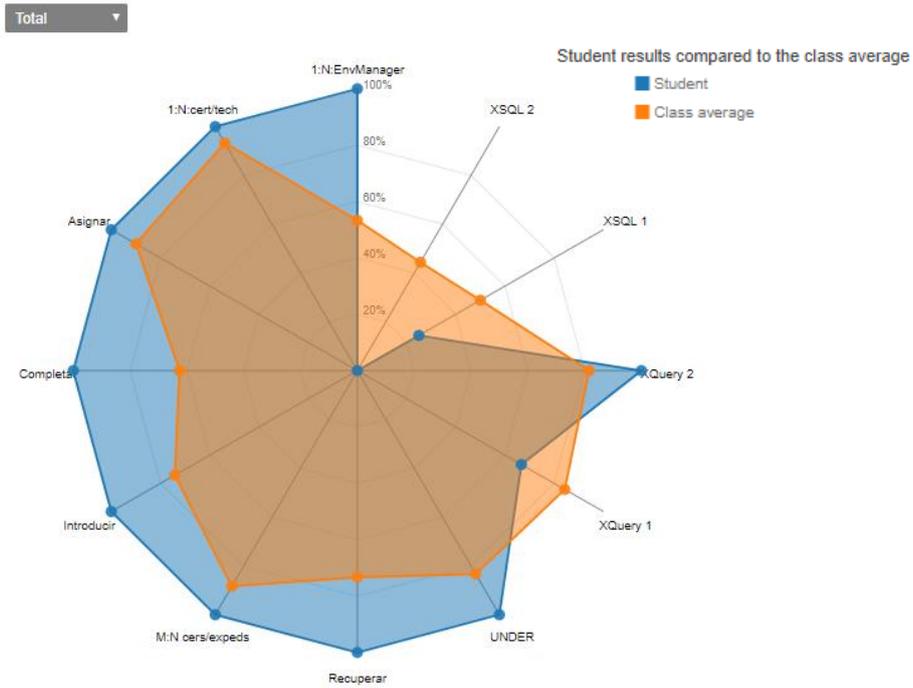


Ilustración 17. Diagrama de araña

Seleccionando en la lista desplegable, situado en la parte izquierda superior, se puede seleccionar una vista más concreta de un único examen. En este caso, se muestra también tanto el resultado final del alumno en dicho examen como la media de la clase.

En este ejemplo, se puede apreciar que en el examen '18.3 OR' el alumno actual obtuvo una nota muy destacada en dicha prueba al obtener la máxima puntuación en todas las preguntas muy por encima de la media de la clase.

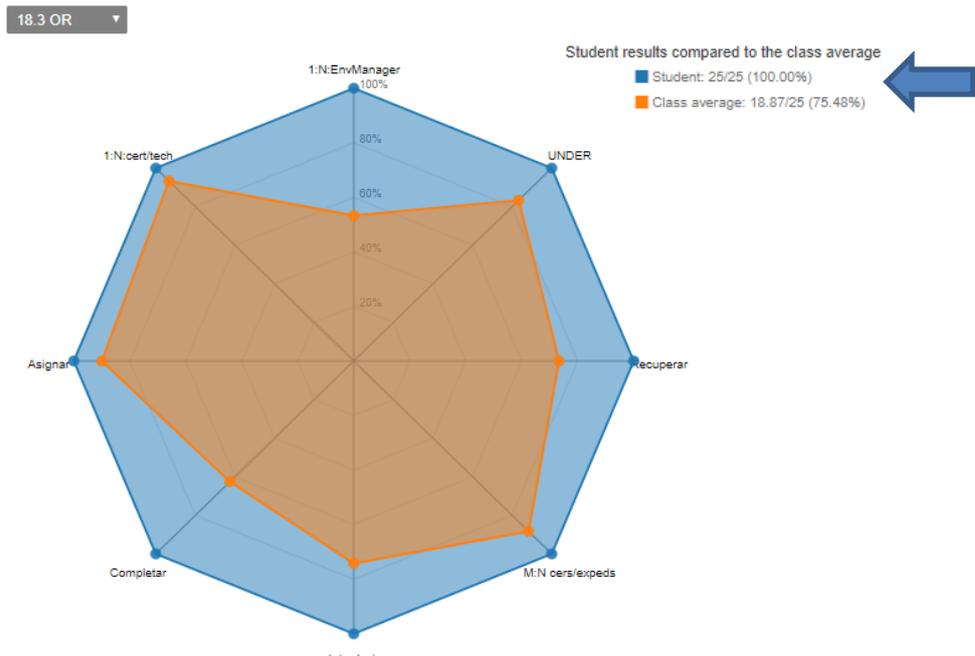


Ilustración 18. Diagrama de araña de un examen

### 3.2.2.3. Diagrama de dispersión

Un diagrama de dispersión [17] es un tipo de gráfico donde los valores correspondientes de un conjunto de datos se ubican como puntos de un plano cartesiano. El objetivo es comparar el grado de relación entre dos variables (X, Y) y hacer predicciones basadas en los datos. En el ejemplo [18] se puede ver la media de ingresos de una tienda de helados según la temperatura que haga.

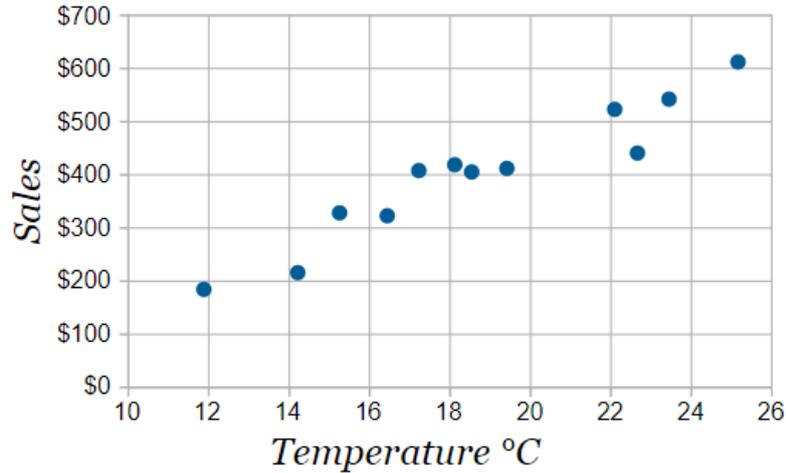


Ilustración 19. Ejemplo de diagrama de dispersión

#### 3.2.2.3.1 Diagrama de dispersión sobre la evaluación continua

En este proyecto se obtienen, para cada examen de la evaluación continua, las distintas notas que se han obtenido y cuantas veces, es decir, la frecuencia de cada una de ellas. El objetivo de este gráfico es poder apreciar la evolución de las notas de los alumnos a lo largo de la evaluación continua. El eje X indica cada uno de los exámenes y el eje Y la nota, mientras que la anchura de los puntos la frecuencia.

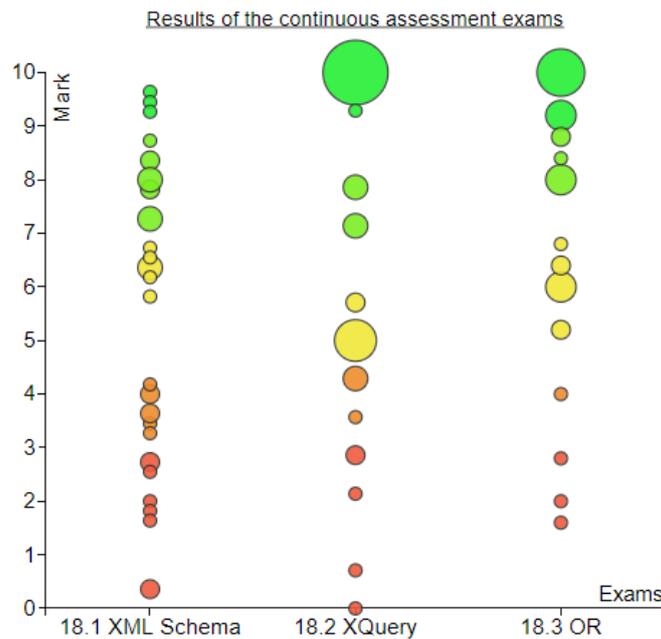


Ilustración 20. Diagrama de dispersión de la evaluación continua

Tomando como ejemplo la anterior ilustración, se puede obtener la siguiente información:

- El primer examen de la evaluación, XML Schema, se podría intuir que fue el más complicado ya que fue el examen con mayor número de suspensos. Los resultados están polarizados.
- En el segundo examen, XQuery, fue en el que más veces se obtuvo la máxima puntuación.
- El último examen, OR, fue el que mayor número de aprobados obtuvo.
- La conclusión que se puede tomar es que a medida que ha avanzado la evaluación continua, el número de aprobados y, en consecuencia, las notas medias han ido aumentando.

### 3.2.2.3.2. Diagrama de dispersión sobre un examen

Con este diagrama, el profesor puede ver para cada pregunta del examen, la frecuencia de las distintas notas obtenidas por los alumnos. Gracias a esta visualización, el profesor puede apreciar cuales fueron las preguntas más complicadas y asequibles para los alumnos.

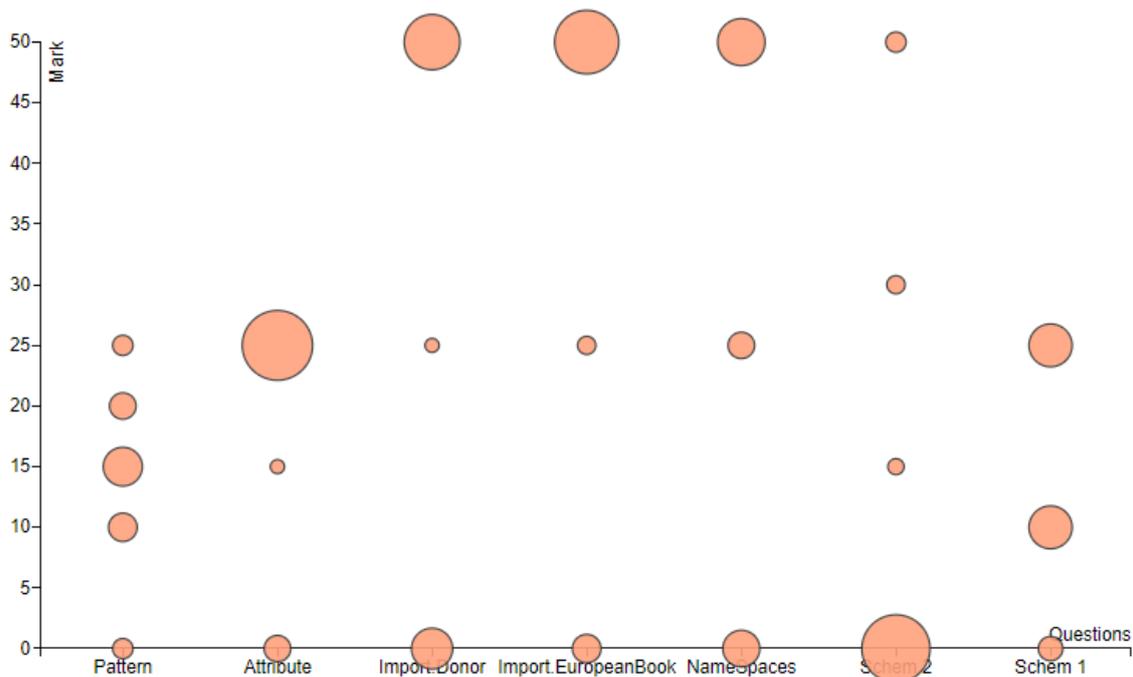


Ilustración 21. Diagrama de dispersión de un examen

Conclusiones que podría obtener el profesor partiendo de la última visualización.

- La pregunta “Schem 2” ha sido la que peor resultados ha obtenido del examen con 27 alumnos sobre 36 obteniendo 0, por lo que se puede intuir que ha sido la más complicada o que los alumnos no estaban lo suficientemente preparados para ella.
- Las preguntas “Attribute” y “Import.EuropeanBook” han sido las que mayor puntuación se han obtenido con 28 y 25 alumnos respectivamente sobre 36.

- Las preguntas “Pattern”, “Schem 1” y “Schem 2”, son las únicas que la nota con mayor frecuencia no ha sido la máxima, siendo bastante distribuida entre las distintas notas posibles.

### 3.2.5. Tabla preguntas

En la siguiente tabla se puede apreciar que preguntas de las que se han planteado en el apartado 2.2.1 responden cada uno de los diagramas vistos.

		Alluvial	Araña	Dispersión evaluación	Dispersión examen
P1	¿Existe alguna relación entre dos preguntas de un examen?	X			
P2	¿Cuál es la pregunta más complicada?	X	X		X
P3	¿Cuál es la pregunta más fácil?	X	X		X
P4	¿Qué competencias no han adquirido los alumnos en la asignatura?	X	X		X
P5	¿Existe alguna relación entre el resultado de una pregunta y el resultado final del examen?	X			
P6	¿El resultado de un examen tiene alguna relación con el resto de exámenes?	X			
P7	¿Cuál es la tendencia de las notas a lo largo de la evaluación continua?			X	
P8	¿Qué tal son los resultados de un alumno en comparación con la media de la clase?		X		
P9	¿Cuáles son los ejercicios más complicados para un alumno en comparación con el resto de la clase?		X		

*Tabla 4. Tabla relación preguntas-gráficas*

# 4. Diseño: Arquitectura de la solución

---

En este capítulo se encuentra todo lo relacionado con el diseño del proyecto. En él se presentan los requisitos de la extensión junto el diagrama de casos de uso, el flujo de eventos correspondiente a cada uno de los casos de uso y, por último, el modelo de datos.

## 4.1. Requisitos

### 4.1.1. Requisitos funcionales

- Cuando un usuario accede a su cuenta de hypothesis, si esta tiene al menos un grupo relativo a un examen, en la página de perfil aparece un gráfico de dispersión mostrando la evolución de los alumnos a lo largo de la evaluación continua.
- Cuando un usuario accede a un grupo de un examen, se le da la opción de visualizar dos tipos de gráficos distintos entre los que se encuentran:
  - Gráfico alluvial: Muestra las relaciones entre las preguntas del examen y el resto de exámenes de la asignatura. Debe de contener los siguientes requisitos:
    - Un usuario puede eliminar una pregunta o un examen del diagrama alluvial.
    - El usuario puede intercambiar la posición de dos preguntas o exámenes.
  - Gráfico de dispersión: Muestra por cada pregunta cuantos alumnos han obtenido cada nota.
- Cuando un usuario accede a la carpeta de un alumno en Google Drive, aparece un gráfico de araña comparando los resultados del alumno a lo largo de la evaluación continua respecto a la clase.
  - Un usuario puede seleccionar una vista de un examen en concreto.

### 4.1.2. Requisitos no funcionales

Los requerimientos no funcionales representan características generales y restricciones de la extensión web desarrollada.

- Interoperabilidad: La extensión web está desarrollada para el navegador Google Chrome. Otros navegadores como Opera o Firefox también son compatibles con las extensiones de Chrome.
- Es compatible con cualquier sistema operativo que admita Google Chrome: Windows, Mac y Linux.
- Los tiempos de carga son mínimos gracias a la eficiencia de las librerías usadas.
- El rendimiento puede verse ligeramente afectado según las limitaciones de la maquina con la que se trabaje.
- Existe un límite de datos con la que pueda trabajar la extensión, esta es de 5000 anotaciones.
- Solo funciona para trabajar con una asignatura.
- En el caso de diagrama alluvial, en caso de haber muchas preguntas, para una mayor legibilidad, se incluye una barra de desplazamiento.
- La extensión únicamente está disponible en inglés.
- Usabilidad: La aplicación es intuitiva al uso y fácil de utilizar a simple vista.
- En el futuro debe de ser fácil de integrar en sistemas similares que hagan uso de anotaciones web.
- La extensión cuenta con un manual de usuario.

### 4.1.3. Requisitos legislativos

Los requerimientos legislativos son las características que debe cumplir la extensión web para que opere dentro de la ley. Actualmente no está cumpliendo todos los requisitos ya que la universidad y las leyes de protección de datos dictan que los exámenes y las notas no pueden estar alojadas en un servidor externo o que no sea propiedad de la universidad. En este caso, esta información esta guardada tanto en los servidores Google Drive como la de Hypothesis. Estas consideraciones no se han tenido en cuenta pues ya viene condicionada por la decisión del proyecto. Aun así, todos los datos son privados y están encriptados cumpliendo en todo momento con el GDPR (Reglamento General de Protección de Datos Europeo).

### 4.2. Casos de uso

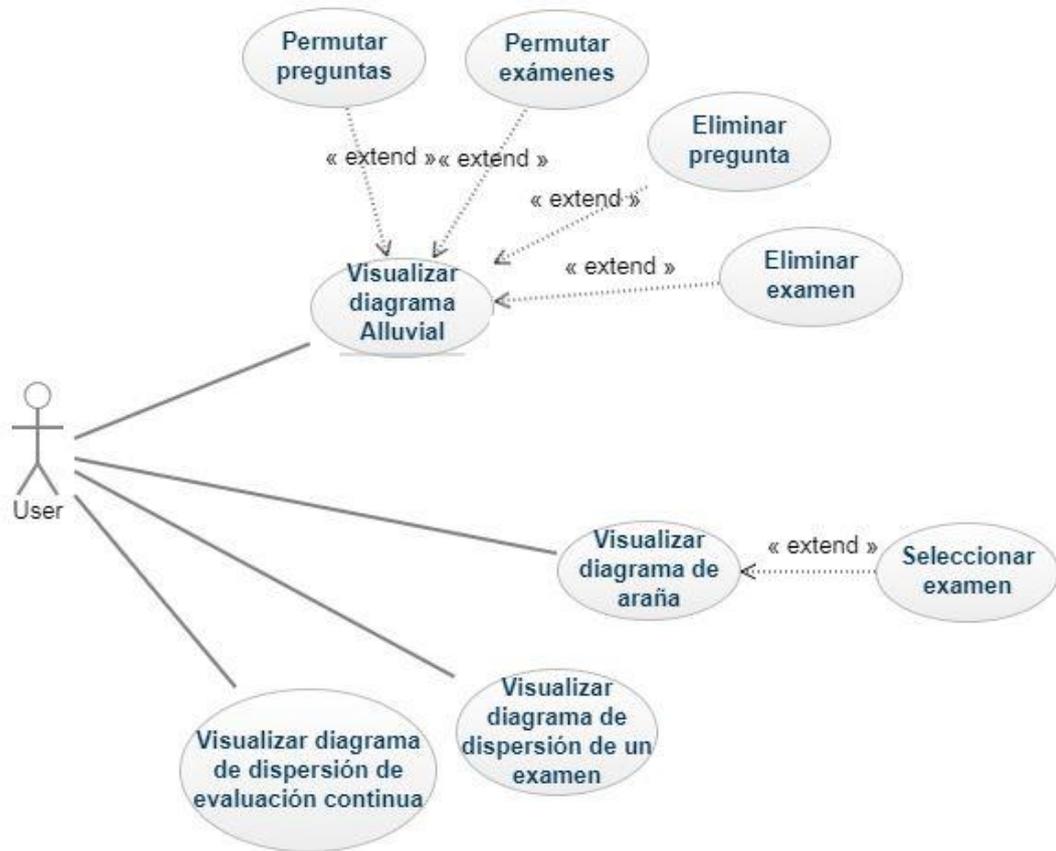


Ilustración 22. Diagrama de casos de uso

### 4.3. Flujo de eventos

En esta sección se presentan los diagramas de flujo de eventos correspondientes a los casos de uso mostrados en el diagrama del apartado anterior.

#### 4.3.1. Visualizar diagrama de dispersión de evaluación continua

**Descripción:** Permite al usuario visualizar un diagrama de dispersión con la evolución de las notas a lo largo de la evaluación continua.

**Actores:** Usuario

**Precondiciones:** Haber iniciado sesión en la web de Hypothesis.

**Flujo de eventos principal:**

- El usuario accede a su página de perfil de Hypothesis.
- El sistema obtiene la información de todos los grupos del usuario que sean relativos a un examen y dibuja el diagrama de dispersión.

**Flujo de eventos alternativos:**

- Si no hubiera ningún grupo relativo a un examen, no se dibuja ningún gráfico.

#### 4.3.2. Visualizar diagrama alluvial

**Descripción:** Permite al usuario visualizar el diagrama alluvial.

**Actores:** Usuario

**Precondiciones:** Haber iniciado sesión en la web de Hypothesis.

**Flujo de eventos principal:**

- El usuario accede a un grupo de [Hypothesis](#).
- El sistema comprueba que el grupo al que se ha accedido es relativo a un examen y añade dos botones.
- El usuario selecciona el botón que se indica como "Alluvial diagram".
- El sistema obtiene las anotaciones del grupo actual y dibuja el diagrama.

**Flujo de eventos alternativos:**

- Si el grupo accedido por el usuario no fuera relativo a un examen, el sistema no añade los botones.

#### 4.3.3. Permutar pregunta

**Descripción:** Permite al usuario intercambiar la posición de dos preguntas en el diagrama alluvial.

**Actores:** Usuario

**Precondiciones:** Haber iniciado sesión en la web de Hypothesis y encontrarse dentro de un grupo de un examen.

**Flujo de eventos principal:**

- El usuario selecciona la lista desplegable de una de las columnas de las preguntas del diagrama y selecciona una pregunta.
- El sistema dibuja el diagrama alluvial permutando la columna de la pregunta actual con la seleccionada.

#### 4.3.4. Permutar examen

**Descripción:** Permite al usuario intercambiar la posición de dos exámenes en el diagrama alluvial.

**Actores:** Usuario

**Precondiciones:** Haber iniciado sesión en la web de Hypothesis y encontrarse dentro de un grupo de examen.

**Flujo de eventos principal:**

- El usuario selecciona la lista desplegable de una de las columnas de exámenes del diagrama y selecciona un examen.
- El sistema dibuja el diagrama alluvial permutando la columna actual con la seleccionada.

#### 4.3.5. Eliminar pregunta

**Descripción:** Permite al usuario eliminar una pregunta del diagrama alluvial.

**Actores:** Usuario

**Precondiciones:** Haber iniciado sesión en la web de Hypothesis y encontrarse dentro de un grupo de examen.

**Flujo de eventos principal:**

- El usuario selecciona el botón eliminar de una de las columnas de las preguntas del diagrama.
- El sistema dibuja el diagrama alluvial sin la columna de la pregunta seleccionada.

**Flujo de eventos alternativos:**

-Si solo hubiera una columna de preguntas, el sistema muestra una alerta avisando de que no se puede eliminar la pregunta.

**4.3.6. Eliminar examen**

**Descripción:** Permite al usuario eliminar un examen del diagrama alluvial.

**Actores:** Usuario

**Precondiciones:** Haber iniciado sesión en la web de Hypothesis y encontrarse dentro de un grupo de examen.

**Flujo de eventos principal:**

- El usuario selecciona el botón eliminar de una de las columnas de los exámenes del diagrama.
- El sistema dibuja el diagrama alluvial sin la columna del examen seleccionada.

**4.3.7. Visualizar diagrama de dispersión de un examen**

**Descripción:** Permite al usuario visualizar el diagrama de dispersión con las notas de los alumnos en cada una de las preguntas de un examen.

**Actores:** Usuario

**Precondiciones:** Haber iniciado sesión en la web de Hypothesis y encontrarse dentro de un grupo relativo a un examen.

**Flujo de eventos principal:**

- El usuario accede a un grupo de Hypothesis.
- El sistema comprueba que el grupo al que se ha accedido es relativo a un examen y añade dos botones.
- El usuario selecciona el botón que se indica como "Scatter plot".
- El sistema dibuja el diagrama.

**Flujo de eventos alternativos:**

-Si el grupo accedido por el usuario no fuera relativo a un examen, el sistema no añade los botones.

#### 4.3.8. Visualizar diagrama de araña

**Descripción:** Permite al usuario visualizar el grafico de araña.

**Actores:** Usuario

**Precondiciones:** Haber iniciado sesión en Google Drive

**Flujo de eventos principal:**

- El usuario accede a la carpeta del alumno en Google Drive
- El sistema comprueba que el grupo al que se ha accedido es relativo a un estudiante y dibuja el diagrama de araña.

**Flujo de eventos alternativos:**

- Si no hubiera ningún grupo relativo a un examen, no se dibuja ningún gráfico.

#### 4.3.9. Seleccionar examen

**Descripción:** Permite al usuario visualizar el gráfico de araña.

**Actores:** Usuario

**Precondiciones:** Haber iniciado sesión en Google Drive y estar dentro de la carpeta de exámenes de un alumno.

**Flujo de eventos principal:**

- El usuario selecciona un examen de la lista desplegable.
- El sistema actualiza el gráfico araña con las preguntas del examen seleccionado.

#### 4.4. Modelo de datos

Las anotaciones web que utiliza la extensión web residen en las bases de datos de Hypothesis. Para visualizar y comprender mejor el modelo de datos, se ha realizado el siguiente diagrama de la estructura de datos.

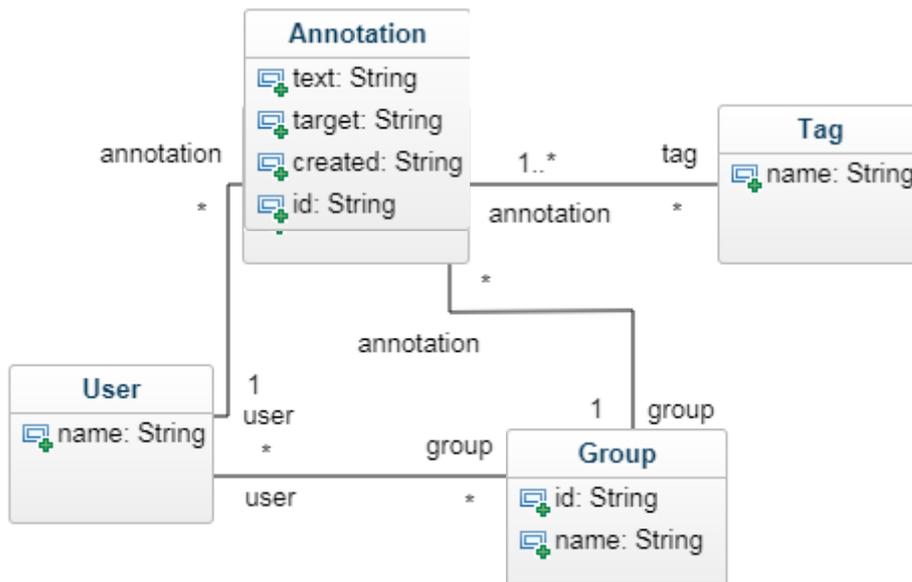


Ilustración 23. Modelo de datos

- Un usuario tiene un nombre que le identifica.
- Un usuario puede tener tantas anotaciones como desee.
- Un usuario puede pertenecer a uno o más grupos, o a ninguno.
- Un grupo tiene una id y un nombre que le identifican.
- Un grupo puede ser privado o público.
- Un grupo puede contener tantas anotaciones como se desee.
- Una anotación tiene una id que le identifica.
- Cada anotación guarda un comentario, el contenido subrayado y cuando se hizo.
- Una anotación pertenece a un grupo y a un usuario.
- Una anotación puede contener etiquetas.
- Una etiqueta puede ser usada por varias anotaciones.

# 5. Implementación

---

En este capítulo se encuentra todo lo relacionado con la implementación del proyecto. En él se presentan la arquitectura de la extensión, las librerías usadas, la API de Hypothesis, las transformaciones realizadas a los datos para obtener cada uno de los gráficos, los diagramas de secuencia y, por último, las dificultades durante el desarrollo.

## 5.1. Aumentación web

El uso de este término se remonta, al menos, al año 1997 y es probable que se haya derivado por analogía al concepto de realidad aumentada. La aumentación web [19] permite que los usuarios accedan a una versión mejorada de una aplicación existente que no fue diseñada originalmente para afrontar las necesidades que cubren esas características agregadas.

La técnica más común es introducir estos cambios mediante scripting, utilizando la API del DOM, una vez que el documento ha sido parseado. Las implementaciones más comunes de este mecanismo son las extensiones de navegador.

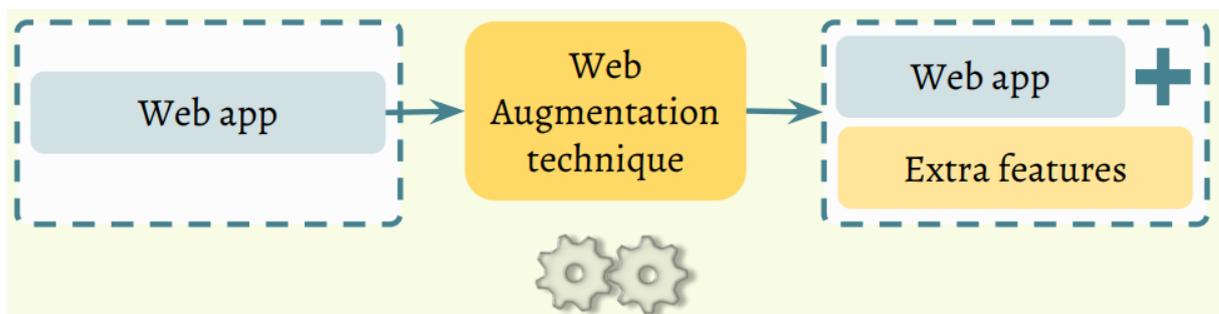


Ilustración 24. Aumentación web

## 5.2. Arquitectura de la extensión

En este apartado se expone la arquitectura de la extensión web del proyecto. Para hacerse una idea, estas extensiones están formado por los elementos que se pueden ver en la siguiente ilustración.

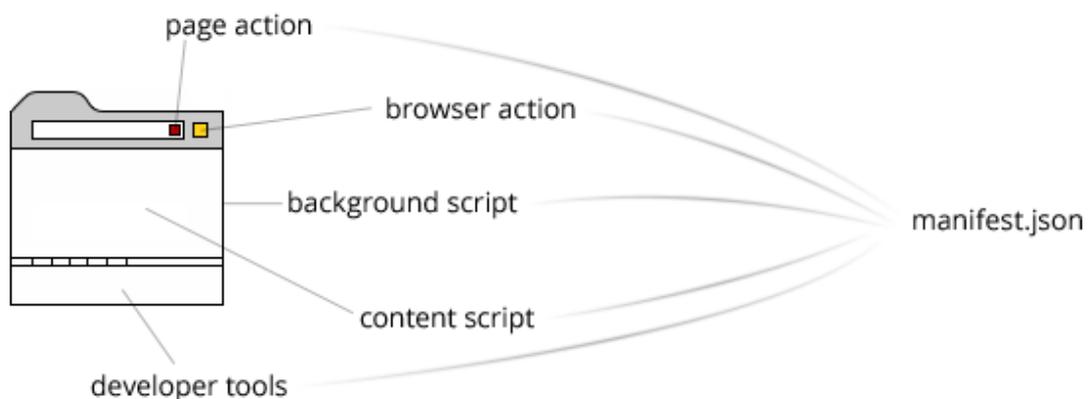


Ilustración 25. Arquitectura de una extensión web

La extensión web contiene un archivo de manifiesto llamado manifest.json. Este archivo en formato JSON contiene toda la información básica de la extensión. A continuación, la descripción de los atributos más importantes del manifiesto:

- **manifest\_version:** Es la versión de manifiesto, se representa por un número entero que especifica la versión del formato de archivo. Es un campo obligatorio.
- **name:** El nombre de la extensión. Este debe estar comprendido por un máximo de 45 caracteres y es un campo obligatorio.
- **version:** Debe estar comprendida de uno a cuatro números enteros separados por puntos que identifican la versión de la extensión. Es un campo obligatorio.
- **description:** Comprende una cadena en texto plano con un máximo de 132 caracteres que describe la funcionalidad de la extensión. Este campo no es obligatorio, pero es recomendable.
- **icons:** Permite incluir uno o más íconos que representen a la extensión. Los íconos generalmente deben estar en formato PNG, porque tiene el mejor soporte para la transparencia.
- **browser\_action:** El browser action añade el icono de la extensión en la barra de herramientas principal que se encuentra en la parte superior del navegador Google Chrome, a la derecha de la barra de direcciones.
- **permissions:** Indica los recursos a los que puede acceder la extensión.
- **Background:** Se trata de un archivo encargado de gestionar alguna tarea o estado en un segundo plano. En nuestro caso se encargará de obtener la API token del usuario para poder realizar las llamadas a la API.
- **content script:** Son archivos JavaScript que se ejecutan en el contexto de las páginas web. Mediante el uso del estándar Document Object Model (DOM), pueden leer los detalles de las páginas web o realizar cambios en ellos. Por tanto, lo razonable es que la aumentación que se va a generar trabaje sobre el content script. Contiene a su vez los siguientes campos:
  - **matches:** Especifica a que páginas se le inyectara el script.
  - **css:** Especifica una lista de archivos css que serán insertados en la página que se solicite en el navegador.
  - **js:** Especifica una lista de archivos JavaScript que serán insertados en la página que se solicite en el navegador. Estos serán los encargados de crear los diagramas e inyectarlos en las páginas web.

El formato del manifest.json del proyecto es el siguiente:

```

{
  "manifest_version": 2,
  "name": "Mark&GoViz",
  "description": "Web annotations visualization in the context of rubric based correction",
  "version": "1.0",
  "icons": {
    "16": "icons/icon-16.png",
    "48": "icons/icon-48.png",
    "128": "icons/icon-128.png"
  },

  "browser_action": {
    "default_icon": "icons/icon.png"
  },

  "background": {
    "scripts": [
      "scripts/background.js"
    ]
  },
  "content_scripts": [
    {
      "matches": ["https://hypothes.is/groups/*"],
      "css": ["styles/style.css"],
      "js": ["scripts/MarkAndGoVizForExamGroup.js", "scripts/ScatterPlotExam.js", "scripts/SankeyDiagram.js"]
    },
    {
      "matches": ["https://drive.google.com/drive/*"],
      "css": ["styles/style.css"],
      "js": ["scripts/d3-radar.js", "scripts/SpiderChart.js"]
    },
    {
      "matches": ["https://hypothes.is/users/*"],
      "css": ["styles/style.css"],
      "js": ["scripts/ScatterPlotContinuousAssessment.js"]
    }
  ],
  "permissions": [
    "activeTab",
    "tabs",
    "<all_urls>"
  ]
}

```

*Ilustración 26. Archivo manifiesto de la extensión*

## 5.3. API, lenguaje y librerías

### 5.3.1. API de Hypothesis

Para obtener todas las anotaciones se hará uso de la API de Hypothesis [20]. Para ello, primero hay que conocer como son las estructuras de las dos llamadas que se hacen uso.

#### 5.3.1.1. Obtener anotaciones web

La estructura de la llamada es la siguiente:

<https://hypothes.is/api/search?>

A continuación, hay que añadirle los parámetros de búsqueda.

<b>limit</b>	El número máximo de anotaciones que se desean recibir. El límite máximo de anotaciones que puede devolver es 200. Por defecto son 20 anotaciones.
<b>offset</b>	El número de anotaciones iniciales que se desean saltar, es decir, que no devuelve.
<b>order</b>	Orden en el que se van a enviar las anotaciones. Puede tomar los valores 'asc' o 'desc' para ordenarlos de más antiguos a más recientes o al revés respectivamente.
<b>group</b>	Limita los resultados a las anotaciones hechas en el grupo especificado.
<b>user</b>	Limita los resultados a las anotaciones hechas por el usuario especificado.
<b>uri</b>	Limite los resultados a anotaciones que coincidan con el URI especificado.
<b>tag</b>	Limita los resultados a anotaciones etiquetadas con el valor especificado.

*Tabla 5. Parámetros de la llamada a la API para obtener anotaciones*

Los 4 primeros son los únicos parámetros que interesan en este caso para realizar la llamada.

- Se establece el límite máximo (200).
- En caso de no haber recibido todas las anotaciones (cuando hay más de 200 anotaciones) entonces se vuelve a realizar otra llamada para solicitar otras 200 anotaciones, pero esta vez añadiendo el parámetro offset con valor 200 para saltar las primeras 200 anotaciones.
- Se indica que el orden sea ascendente (de más antiguas a más recientes).
- El grupo del examen del que se obtienen las anotaciones, se obtiene a raíz de la URL de la página del grupo.

Además de estos parámetros, es necesario la API Token del usuario para poder realizar la llamada.

[https://hypothes.is/api/search?group="+group+"&limit=200&order=asc](https://hypothes.is/api/search?group=)

Tras realizar la llamada a la API de Hypothesis, ésta devuelve una respuesta JSON con las anotaciones que se le han solicitado. El resultado es un objeto JSON formado por dos atributos: "rows" y "total". El primero contiene una lista de objetos que corresponden a cada una de las anotaciones y el segundo es el número de anotaciones totales que hay en grupo especificado. A continuación, la estructura de la respuesta:

```
{
  "rows": [
    {
      "group": "string",
      "tags": [],
      "target": "string",
      "text": "string",
      "uri": "string",
      "id": "string"
    }
  ],
  "total": number
}
```

*Ilustración 27. Estructura de las anotaciones*

A continuación, un ejemplo de la estructura.

```

{
  "rows": [
    {
      "group": "JY658rmX",
      "target": "/author = current()/author and /genre !=
        current()/genre">",
      "tags": [
        "exam:isCriteriaOf:Schem 2",
        "exam:mark:15"
      ],
      "text": "",
      "created": "2018-05-25T17:31:17.379162+00:00",
      "uri": "https://drive.google.com/drive/folders
        /159xx2fEyzhE2t1U5yXhzC6zl0Uwofc1F",
      "user": "acct:Oscar@hypothes.is",
      "id": "bkkSFGBBEeiDkgs2dv47dQ"
    }
  ],
  "total": 1
}

```

*Ilustración 28. Ejemplo de la estructura de una anotación*

Aunque el objeto tenga más atributos, aquí solo vamos a mencionar los más importantes. La definición del objeto anotación es la siguiente:

<b>group</b>	Grupo al que pertenece la anotación.
<b>target</b>	Contenido subrayado.
<b>tags</b>	Etiquetas de la anotación.
<b>text</b>	Comentario.
<b>created</b>	Fecha de creación de la anotación.
<b>uri</b>	URI de la página donde se realizó la anotación.
<b>user</b>	Usuario que creó la anotación.
<b>id</b>	Id de la anotación.

*Tabla 6. Definición del objeto anotación*

Una vez recibida la respuesta, se comprueba si el número de anotaciones que se encuentra en el atributo "rows" coincide con el de "total". En caso de que no sea así, hay que realizar otra llamada a la API, pero esta vez con el parámetro offset para poder saltar las 200 primeras anotaciones. Así sucesivamente hasta obtener todas.

#### 5.3.1.2. Obtener grupos

La estructura de la llamada es la siguiente:

<https://hypothes.is/api/groups?>

A diferencia del anterior, no hace falta ningún parámetro. Eso sí, es necesario la API Token del usuario para poder realizar la llamada.

La llamada devuelve una respuesta JSON con los grupos a los que pertenece el usuario. A continuación, la estructura de la respuesta:

```
{
  "id": string,
  "links": string,
  "name": string,
  "type": string
}
```

Ilustración 29. Estructura del objeto grupo

<b>id</b>	ID del grupo.
<b>links</b>	URL de la página del grupo.
<b>name</b>	Nombre del grupo.
<b>type</b>	Indica si el grupo es privado o no.

Tabla 7. Definición del objeto grupo

Por ejemplo:

```
[
  {
    "name": "Public",
    "links": "https://hypothes.is/groups/__world__/public",
    "id": "__world__",
    "type": "open"
  },
  {
    "name": "18.1 XML Schema",
    "links": "https://hypothes.is/groups/riVj9Xqw/18-1-xml-schema",
    "id": "riVj9Xqw",
    "type": "private"
  },
  {
    "name": "18.3 OR",
    "links": "https://hypothes.is/groups/Pk1XQv8D/18-3-or",
    "id": "Pk1XQv8D",
    "type": "private"
  }
]
```

Ilustración 30. Ejemplo de la estructura de un grupo

### 5.3.2. Lenguaje: JavaScript

El lenguaje de programación utilizado ha sido JavaScript. En este caso se ha hecho adaptándose a ECMAScript 6 (abreviado como ES6) [21], el nuevo estándar de JavaScript. Esta nueva versión incluye muchas características nuevas que facilitan la sintaxis y el trabajo a la hora de programar en este lenguaje.

### 5.3.3. Librería de gráficos: D3.js

En el mercado existen una gran variedad de herramientas para la visualización de datos, algunas se sustentan en lenguajes de programación como R, otras orientadas a la web como Chartist.js, Chart.js o la utilizada en este proyecto, D3.js [22].

D3.js (Data-Driven Documents) es una librería JavaScript, desarrollado por Mike Bostock y publicado en el año 2011, para la manipulación de documentos basados en datos. Gracias al uso de los estándares web, funciona en cualquier navegador y dispositivo. Existe una multitud de tipos de gráficos, desde simples gráficos de barras o líneas, hasta complejos gráficos de gran riqueza como se puede revisar en la galería de ejemplos de la web de D3 para comprender el potencial que tiene.



Ilustración 31. Galería de ejemplos de D3.js

### 5.3.4. Librería para manipulación de datos: Lodash

Lodash [23] es a día de hoy la librería de JavaScript más famosa para trabajar con colecciones de objetos. Esta librería incluye una gran cantidad de utilidades para realizar funciones comunes y simplificar el manejo y edición de objetos, strings, arrays, etc. El objetivo es simplificar el trabajo a realizar consiguiendo así que el código sea mucho más legible y fácil de seguir para terceras personas.

En este proyecto se usa Lodash para permitir manipular estructuras JSON de forma más sencilla, simplificando considerablemente las operaciones más frecuentes que se realizan como ordenaciones, filtrados, búsquedas, etc.

## 5.4. Desarrollo

En este apartado se explican las transformaciones que se han ido realizando a los datos obtenidos de la API de Hypothesis en cada uno de los gráficos. Para ello, se hará uso de varios diagramas para ir explicando dichas transformaciones de manera simplificada. En el anexo A de esta memoria se puede encontrar una explicación más detallada de estas transformaciones.

### 5.4.1 Diagrama alluvial

Una vez obtenidas las anotaciones de la llamada de la API, solo hace falta conseguir la estructura de datos necesaria para dibujar este tipo de diagramas. Para ello, es necesario ir haciendo distintas transformaciones a partir de las anotaciones e ir creando estructuras intermedias. La estructura que se está buscando ha de tener de la siguiente forma:

```
{
  "nodes": [
    {
      "id": string,
      "visualization": string
    }
  ],
  "links": [
    {
      "source": string,
      "target": string,
      "value": number,
      "finalResult": string
    }
  ]
}
```

*Ilustración 32. Diagrama alluvial: estructura final*

- nodes: Escala de calificaciones de cada pregunta según los criterios de rúbrica del examen. Contiene los atributos "id" y "visualization" para identificarlo y que valor mostrar en el diagrama respectivamente.
- links: Lista de enlaces que conectan las calificaciones de dos preguntas. Los enlaces tienen la siguiente estructura:
  - source: Indica la "id" de la calificación (nodo) de la pregunta de origen.
  - target: Indica la "id" de la calificación de la pregunta de destino.
  - finalResult: Indica si aprobaron o no el examen. Puede tomar dos valores posibles: "Pass" (Aprobado) o "Fail" (Suspense).
  - value: Indica el número de alumnos que sacaron como notas las indicadas en "source" y "target", y obtuvieron el resultado final indicado.

A continuación, se muestra un ejemplo de cómo sería la estructura que se busca.

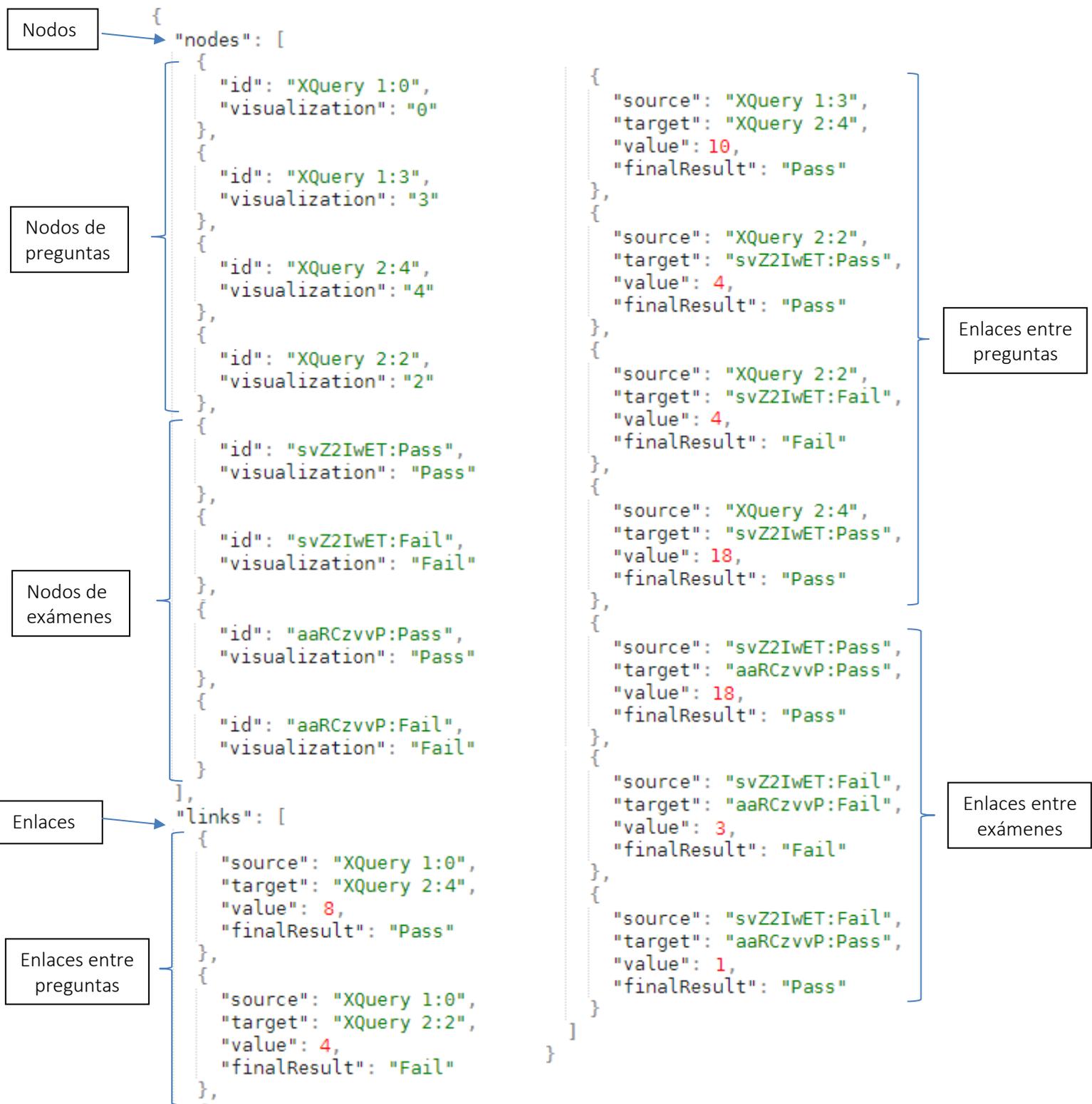
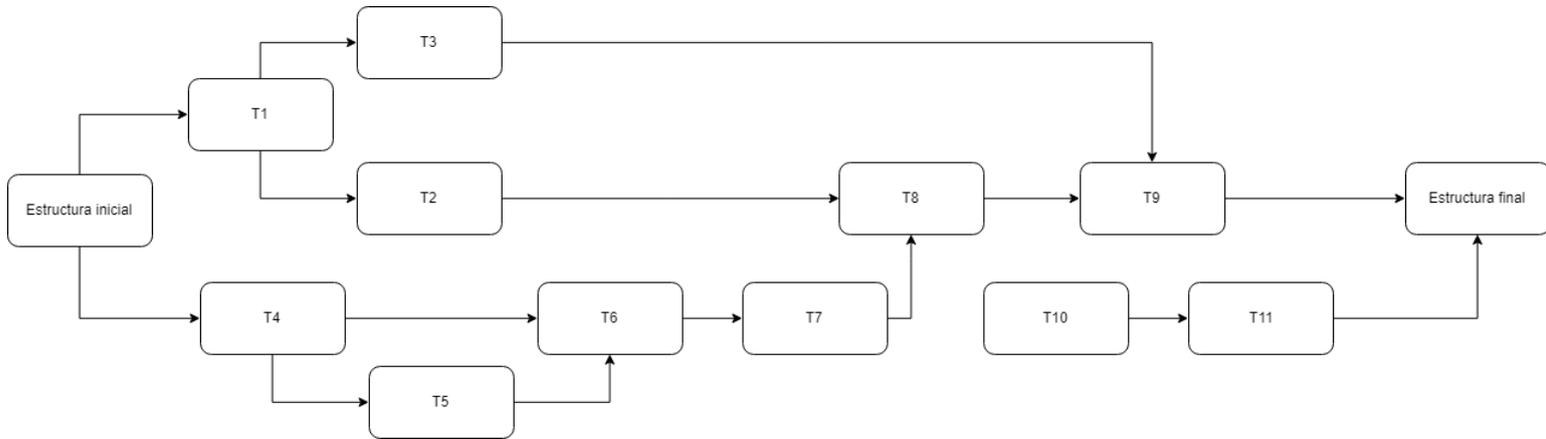


Ilustración 33. Ejemplo de la estructura final del diagrama alluvial

A continuación, se explican las transformaciones.



*Ilustración 34. Diagrama de transformaciones del diagrama alluvial*

- Estructura inicial: Lista de anotaciones del grupo obtenidas de la llamada a la API de Hypothesis.
- T1: Se obtiene una lista con todos los tags (etiquetas) de las anotaciones.
- T2: A raíz de los tags se obtienen las preguntas del examen.
- T3: De los tags también se obtienen los nodos, es decir, la escala de calificación para cada una de las preguntas o competencias que se evalúan en el examen (Rúbrica).
- T4: De cada anotación, se obtiene únicamente la URI para identificar al alumno, la pregunta y la nota.
- T5: Se agrupa por URI sumando todas las notas del alumno en cada uno de los ejercicios para calcular si ha aprobado o suspendido el examen.
- T6: Se actualiza la estructura obtenida en T4. Junto a la URI, la pregunta y la nota se le añade un nuevo campo que indica si el alumno finalmente aprobó o suspendió, información obtenida en T5. Esta información es necesaria a la hora de colorear los enlaces del diagrama.
- T7: Se agrupa por pregunta, es decir, cada pregunta tendrá una lista con las notas de cada alumno en dicha pregunta.
- T8: Se obtienen las distintas notas que han obtenido los alumnos en la primera pregunta. A continuación, se iterará sobre esa lista de notas y, para cada una, se obtiene qué alumnos sacaron dicha nota y qué nota obtuvieron en la segunda pregunta. Se cuentan las distintas notas que se han obtenido en la segunda pregunta y se va creando una lista de objetos enlace. Se repite el proceso con la segunda y la tercera pregunta y así sucesivamente hasta que se llegue a la última.
- T9: Una vez obtenido los nodos (T3) y los enlaces (T8), se unen en una estructura final, pero sin las relaciones con los otros exámenes, es decir, únicamente entre las preguntas del examen. Esto se debe a que la obtención de esas relaciones requiere más tiempo.
- T10: Se obtienen todos los grupos del usuario mediante la API y se comprueba si cada uno es de un examen o no. En caso afirmativo, se obtienen sus anotaciones y se realizan las mismas transformaciones que en T4 y T5 para saber si aprobó o no cada alumno.

- T11: Por cada examen, se obtienen cuantos alumnos que aprobaron ese examen en el siguiente aprobaron o suspendieron y lo mismo con los alumnos que suspendieron. De esta manera se generan los enlaces de los exámenes.
- Estructura final: La estructura obtenida en T9 se actualiza con la obtenida en T11 obteniendo la estructura final.

#### 5.4.2. Diagrama de araña

En este apartado se va a explicar la consecución de la estructura de datos necesaria para dibujar un diagrama de araña. Para ello, es necesario ir haciendo distintas transformaciones a partir de las anotaciones obtenidas e ir creando estructuras intermedias. La estructura que se está buscando ha de tener la siguiente forma:

```
[
  [
    {
      "axis": string,
      "mark": number
      "maximumValue": CriterioRubricaType*,
      "percentage": number,
      "group": string
    }
  ]
]
```

*Ilustración 35. Diagrama de araña: estructura final*

\* *CriterioRubricaType* es un subconjunto del tipo *number* que no va a admitir cualquier valor, sino solo unos muy concretos para poder referirse a las posibles notas para una pregunta en concreto.

Esta estructura está formada por un array de 2 arrays de objetos. Estos objetos corresponden a cada una de las preguntas con el siguiente formato:

- *axis*: Nombre del eje que corresponde a una pregunta de examen.
- *mark*: *mark*: Nota del alumno y media de la clase en la pregunta en el primer y segundo array respectivamente.
- *maximumValue*: Nota máxima que se puede obtener en la pregunta.
- *percentage*: Porcentaje de la puntuación de la pregunta obtenida.
- *group*: ID del examen al que pertenece la pregunta. Corresponde a la ID Grupo de Hypothesis del examen.

El primero de los dos arrays corresponden a las preguntas del alumno mientras que, el segundo, a las medias de los alumnos de la clase. Por ejemplo:

```
[
  [
    {
      "axis": "XQuery 1",
      "mark": 30,
      "maximumValue": 50,
      "percentage": 60,
      "group": "18.2 XQuery"
    },
    {
      "axis": "XQuery 2",
      "mark": 25,
      "maximumValue": 25,
      "percentage": 100,
      "group": "18.2 XQuery"
    }
  ],
  [
    {
      "axis": "XQuery 1",
      "mark": 27,
      "maximumValue": 50,
      "percentage": 54,
      "group": "18.2 XQuery"
    },
    {
      "axis": "XQuery 2",
      "mark": 18.4,
      "maximumValue": 25,
      "percentage": 73.6,
      "group": "18.2 XQuery"
    }
  ]
]
```

Ilustración 36. Ejemplo de estructura final del diagrama de araña

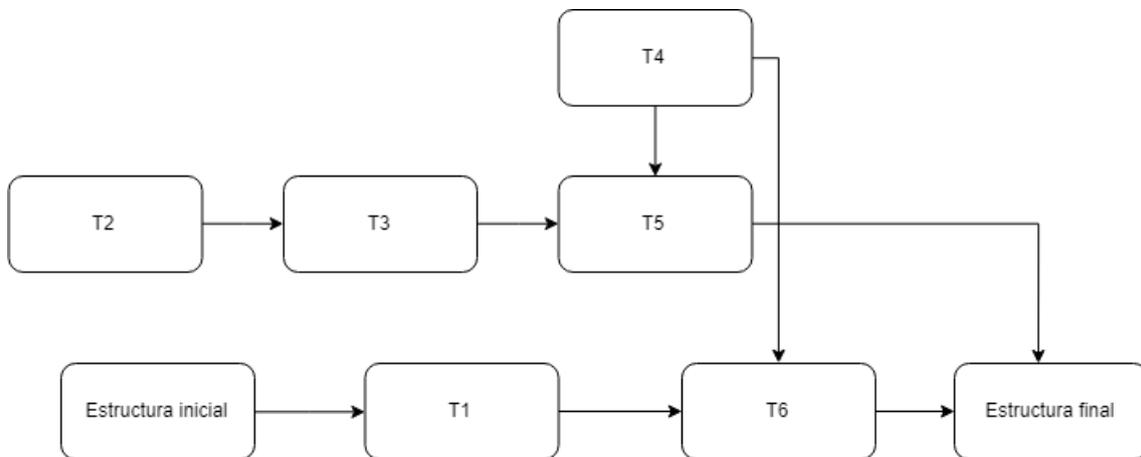


Ilustración 37. Diagrama de transformaciones del gráfico de araña

- Estructura inicial: Lista de anotaciones únicamente del alumno obtenidas de la llamada a la API de Hypothesis.
- T1: De cada anotación, se obtiene pregunta, nota y la ID del examen.
- T2: Para obtener las notas medias de la clase, se obtienen todas las anotaciones de cada uno de los grupos de exámenes. De cada una de estas anotaciones, se obtienen la URI, la pregunta y la nota.
- T3: La estructura de T2, se agrupa por pregunta y se calcula la nota media de cada una.
- T4: Se obtiene cual es la máxima puntuación posible para cada ejercicio de los exámenes, es decir, sobre cuanto se evalúa la pregunta.
- T5: A cada elemento de T3 se le asignan nuevos atributos: sobre cuánto se evalúa la pregunta y el porcentaje obtenido de la nota del ejercicio.
- T6: A cada elemento de T1 se le añade sobre cuánto se evalúa la pregunta (información obtenida en T4) y el porcentaje obtenido del ejercicio.
- Estructura final: Se concatena el resultado de T5 y T6.

#### 5.4.3. Diagrama de dispersión de la evaluación continua

Para dibujar el diagrama de dispersión es necesario llegar a la siguiente estructura:

```
[
  {
    "exam": string,
    "mark": number,
    "numStudents": number
  }
]
```

*Ilustración 38. Diagrama de dispersión de evaluación continua: Estructura final*

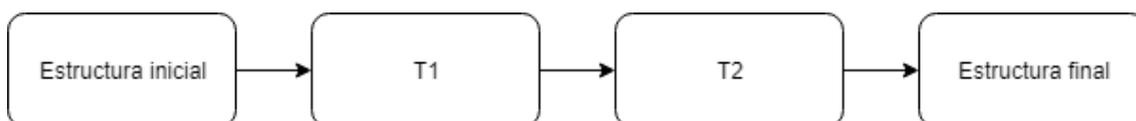
- exam: Nombre del examen.
- mark: Nota obtenida.
- numStudents: Número de alumnos que han obtenido la nota en el examen indicado.

Por ejemplo:

```
[
  {
    "exam": "18.2 XQuery",
    "mark": 8.5,
    "numStudents": 11
  },
  {
    "exam": "18.2 XQuery",
    "mark": 7,
    "numStudents": 9
  },
  {
    "exam": "18.3 OR",
    "mark": 10,
    "numStudents": 10
  },
  {
    "exam": "18.3 OR",
    "mark": 5,
    "numStudents": 10
  }
]
```

*Ilustración 39. Ejemplo de estructura final de diagrama de dispersión de la evaluación continua*

Para la realización de este diagrama es necesario identificar cuáles de los grupos del profesor contienen las anotaciones de un examen. Una vez identificados esos grupos, por cada uno hay que repetir el siguiente proceso:



*Ilustración 40. Diagrama de transformaciones del diagrama de dispersión de la evaluación continua*

- Estructura inicial: Lista de anotaciones del grupo obtenidas de la llamada a la API de Hypothesis.
- T1: De cada anotación, se obtiene únicamente la URI para identificar al alumno, la pregunta y la nota.
- T2: A continuación, se calcula la nota de cada alumno. Se agrupan las preguntas por la URI de cada alumno y se suman las notas para obtener la nota final. A continuación, esa nota se pasa sobre 10.
- Estructura final: Se cuentan las distintas notas y se crea la estructura final deseada previamente vista.
- Se vuelve a repetir el proceso con los grupos restantes relativos a un examen y finalmente se concatena el resultado final de cada iteración.

#### 5.4.4. Diagrama de dispersión de un examen

Los pasos son similares al anterior diagrama de dispersión explicado. El objetivo es llegar a la siguiente estructura final:

```
[
  {
    "question": string,
    "mark": CriterioRubricaType,
    "numAStudents": number
  }
]
```

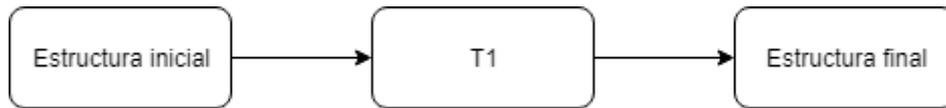
*Ilustración 41. Diagrama de dispersión de un examen: Estructura final*

- question: Nombre de la pregunta.
- mark: Nota obtenida.
- numStudents: Número de alumnos que han obtenido la nota en la pregunta indicada.

Por ejemplo:

```
[
  {
    "question": "XQuery 1",
    "mark": 20,
    "numStudents": 12
  },
  {
    "question": "XQuery 1",
    "mark": 10,
    "numStudents": 8
  },
  {
    "question": "XQuery 1",
    "mark": 0,
    "numStudents": 3
  },
  {
    "question": "XQuery 2",
    "mark": 15,
    "numStudents": 23
  }
]
```

*Ilustración 42. Ejemplo de la estructura final de un diagrama de dispersión de un examen*



*Ilustración 43. Diagrama de transformaciones del diagrama de dispersión de un examen*

- Estructura inicial: Lista de anotaciones del grupo obtenidas de la llamada a la API de Hypothesis.
- T1: De cada anotación, se obtiene únicamente la URI para identificar al alumno, la pregunta y la nota.
- Estructura final: Se agrupa por pregunta y nota, y se cuenta el número de elementos de cada agrupación obteniendo así la estructura deseada.

## 5.5. Diagramas de secuencia

### 5.5.1. Visualizar diagrama de dispersión de un examen

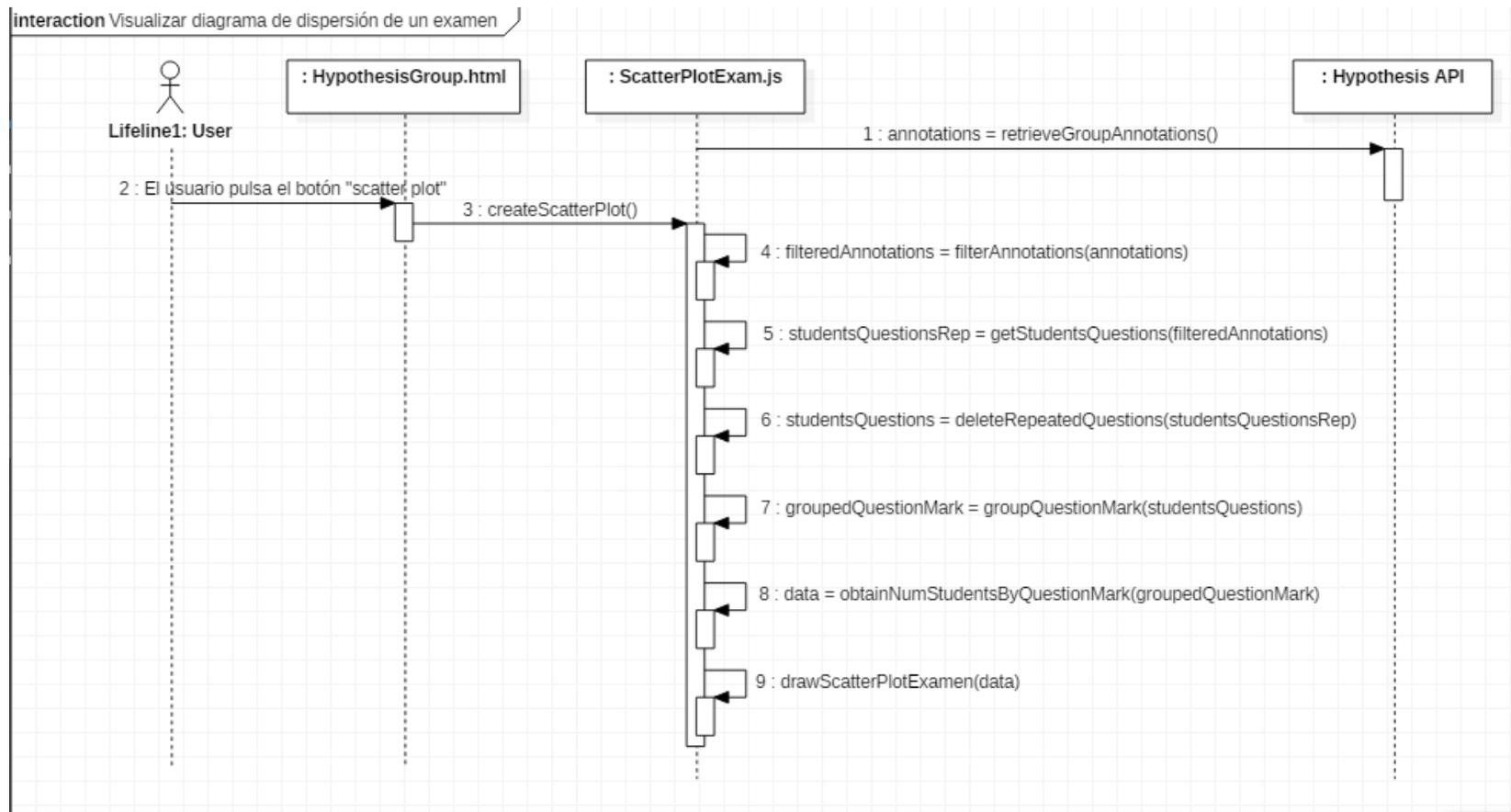


Ilustración 44. Diagrama de secuencia: Visualizar diagrama de dispersión de un examen

### 5.5.2. Visualizar diagrama de dispersión de la evaluación continua

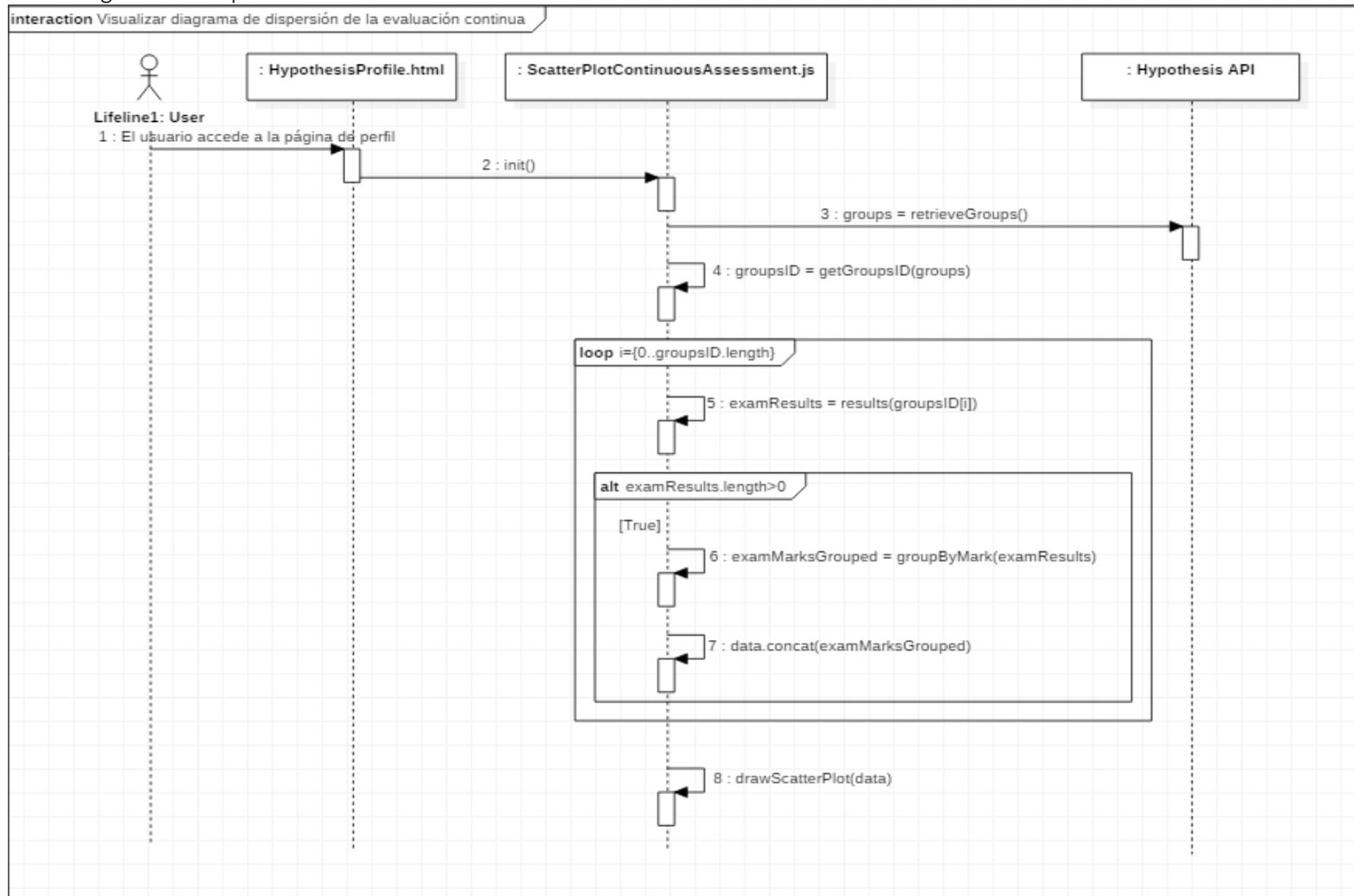


Ilustración 45. Diagrama de secuencia: Visualizar diagrama de dispersión de la evaluación continua

### 5.5.3. Visualizar diagrama de araña

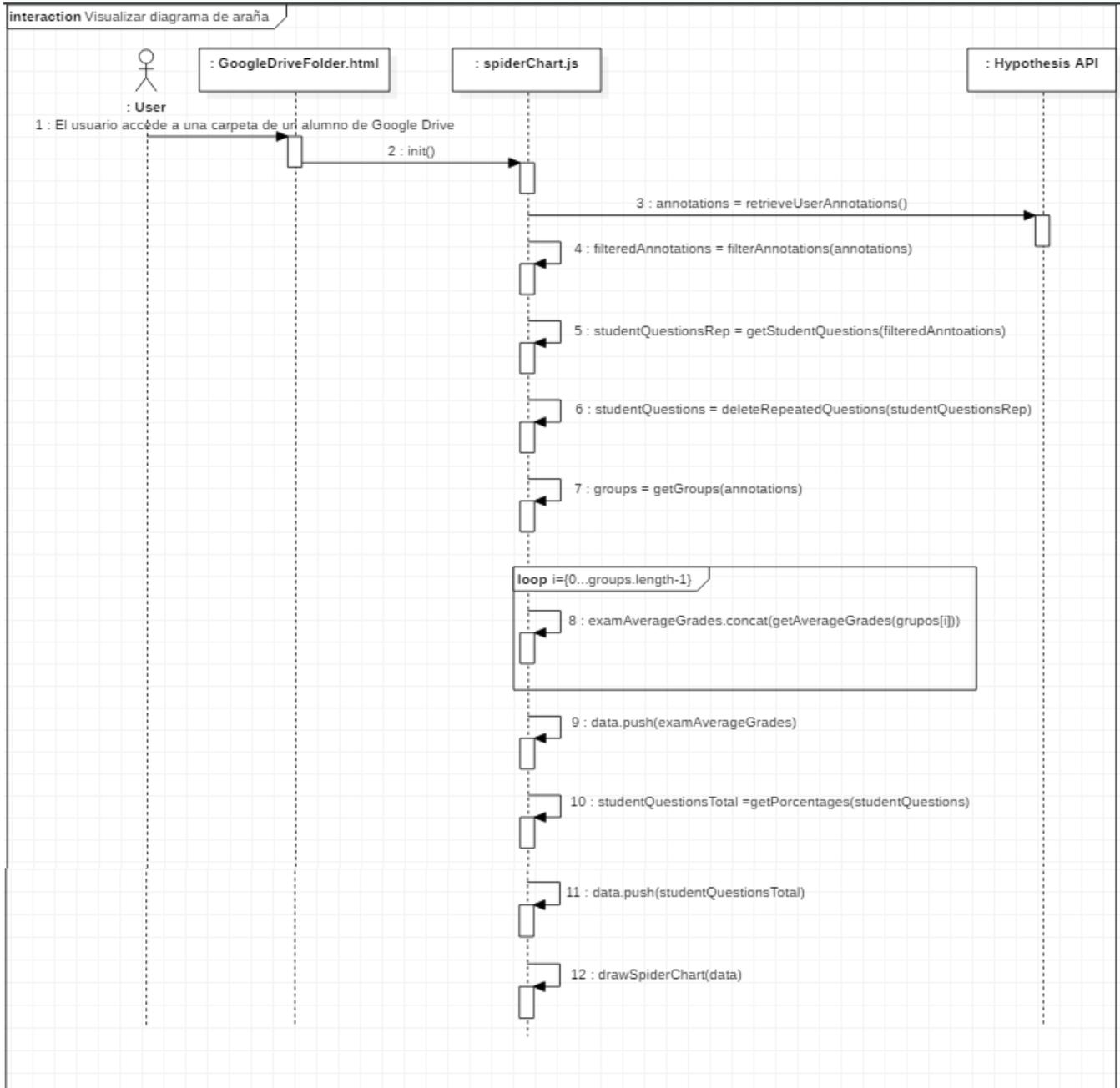


Ilustración 46. Diagrama de secuencia: Visualizar diagrama de araña

### 5.5.4. Visualizar diagrama alluvial

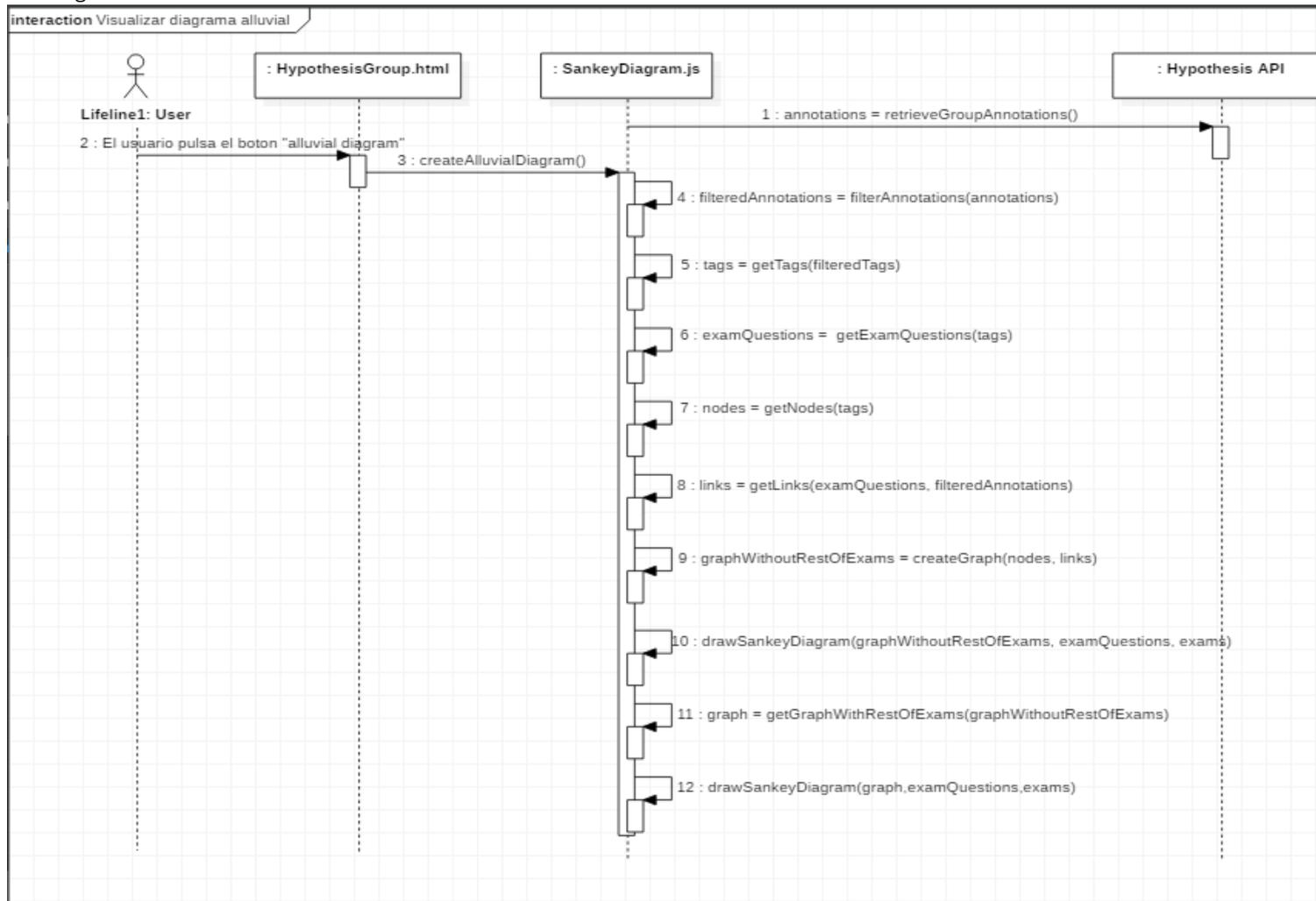


Ilustración 47. Diagrama de secuencia: Visualizar diagrama alluvial

## 5.6. Dificultades durante el desarrollo

En este apartado se describen las dificultades encontradas a lo largo del desarrollo del proyecto. La mayor dificultad del proyecto surgió en torno a la implementación del diagrama de araña. La idea inicial era agregar a una carpeta de un alumno de Google Drive el diagrama de araña, pero debido a que la web de Google Drive va generando los elementos de su página de manera dinámica y toda la navegación se realiza sin recargar la página, provoca una serie de complicaciones al intentar la actualización web.

### 5.6.1. Problema 1

Como la navegación se realiza sin recargar la página, esto provoca en un principio que la extensión no se active cuando se accede a una carpeta al no producirse dicha recarga.

#### 5.6.1.2. Solución 1

Para solucionar este problema se tiene que comprobar en un intervalo de tiempo si la URL ha cambiado y pertenece a una carpeta.

### 5.6.2. Problema 2

Otro problema se encuentra en que los elementos HTML de la carpeta donde se va a agregar el diagrama se generan después (necesita un tiempo para ello) de que el script se ejecute, impidiendo que se encuentren dichos elementos provocando que no funcione.

#### 5.6.2.2. Solución 2

La solución fue similar a la anterior. Durante un intervalo de tiempo comprobar si se han generado los elementos HTML (en este caso los elementos "div") y en ese caso agregar el diagrama.

### 5.6.3. Problema 3

El principal problema se encuentra una vez conseguido que se visualice el diagrama. La lista desplegable de los exámenes no responde cuando se clic sobre ella debido a que Google Drive le sobrescribe el comportamiento inhabilitando la selección de un examen para su visualización en el diagrama de araña.

#### 5.6.3.1. Solución 3

Tras llevar a cabo pruebas, investigar el problema e incluso preguntar en foros como el oficial de Google Drive [24], fue imposible solventarlo. Por ello, se decidió crear una ventana flotante en la esquina inferior derecha donde se añade el diagrama en vez de debajo de las carpetas

como estaba inicialmente planeado, pudiendo así evitar que Google Drive modifique el comportamiento. En las siguientes imágenes se puede ver el cambio.

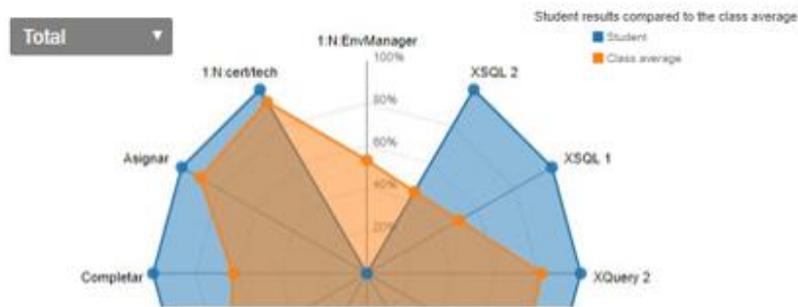
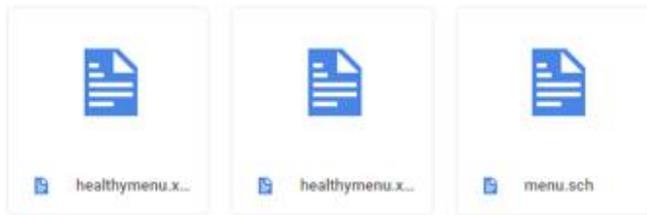


Ilustración 48. Diagrama de araña planificada

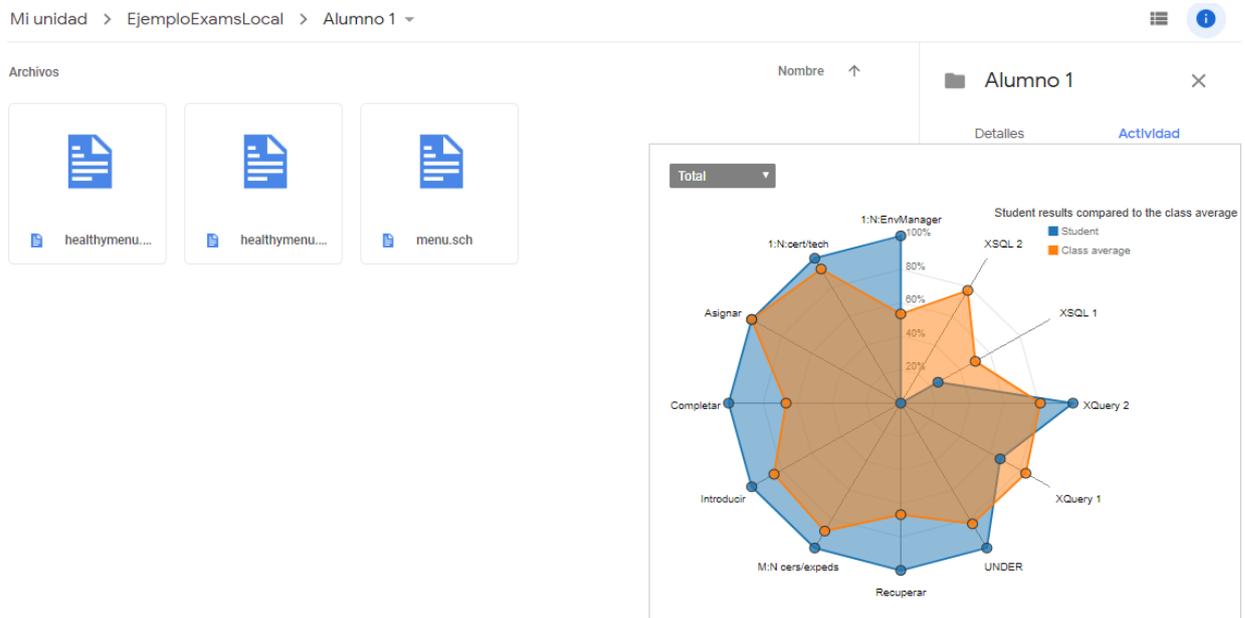


Ilustración 49. Versión final del diagrama de araña

Como alternativa a las soluciones propuestas, se estudia distintas opciones como trasladar en un futuro el diagrama a otra página como puede ser Egela donde en un principio no habría tantos impedimentos.

Durante el desarrollo se han encontrado más dificultades que no tienen la importancia de las ya citadas. Estos problemas se solucionaron a base de pruebas, investigación y preguntando en foros como StackOverflow [25].



# 6. Pruebas

---

En este capítulo se presentan las pruebas realizadas para asegurar que la extensión funciona correctamente y cumple con los requisitos establecidos. Es necesario asegurar que todas las funcionalidades funcionan tal y como se había previsto para evitar fallos que impidan una experiencia satisfactoria por parte de los usuarios.

Para ello se ha empleado el enfoque de caja negra [26], donde la aplicación es estudiada desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce, sin tener en cuenta su funcionamiento interno.

## 6.1. Pruebas

A continuación, se presentan las pruebas realizadas a la extensión web con datos reales de la asignatura de Gestión Avanzada de Información.

### 6.1.1. Prueba 1

**Descripción:** Comprobar que el diagrama de dispersión de evaluación continua aparezca en la página de perfil de Hypothesis.

**Entrada:** El usuario accede a la página de su perfil en Hypothesis.

**Salida esperada:** Se visualiza el diagrama de dispersión.

**Salida real:** Salida esperada

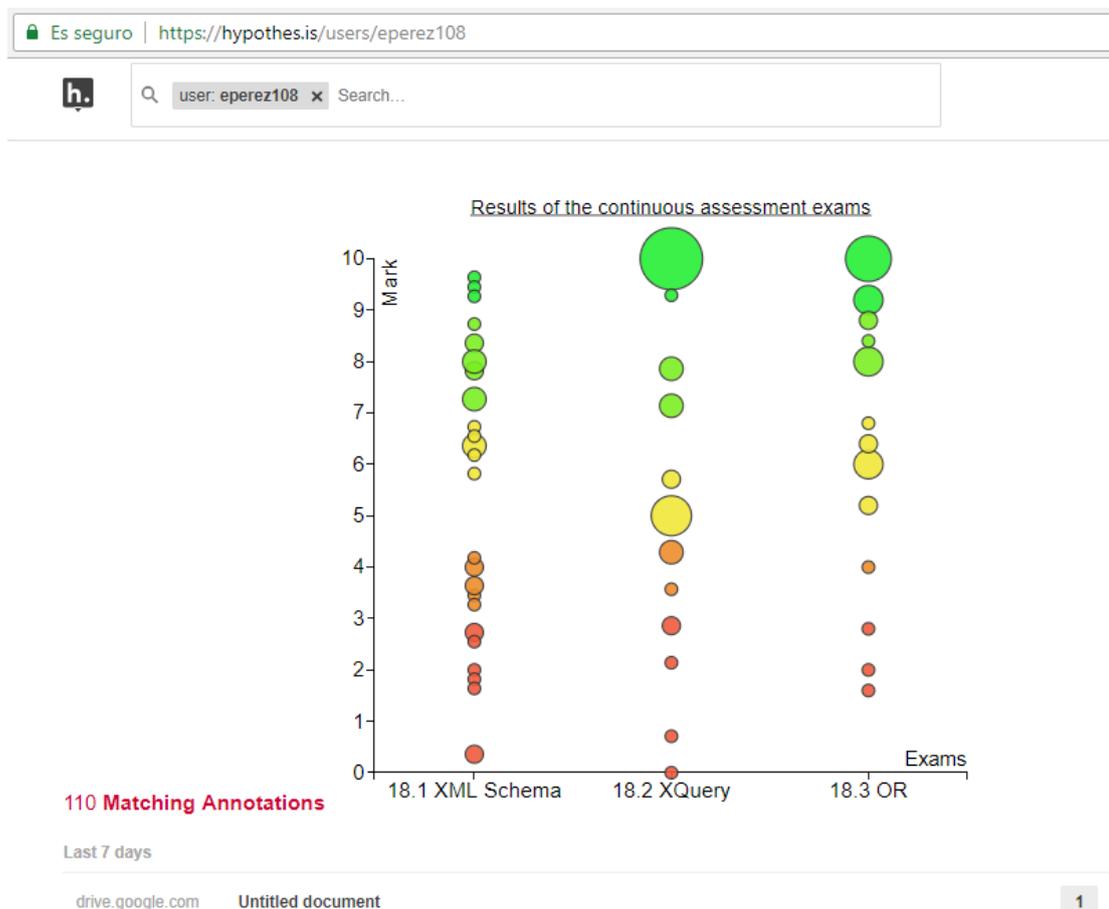


Ilustración 50. Prueba 1

### 6.1.2. Prueba 2

**Descripción:** En caso de que un usuario no tenga ningún grupo de examen, comprobar que el diagrama de dispersión de evaluación continua no aparezca en la página de perfil.

**Entrada:** El usuario accede a la página de su perfil en Hypothesis.

**Salida esperada:** No se visualiza el diagrama de dispersión.

**Salida real:** Salida esperada

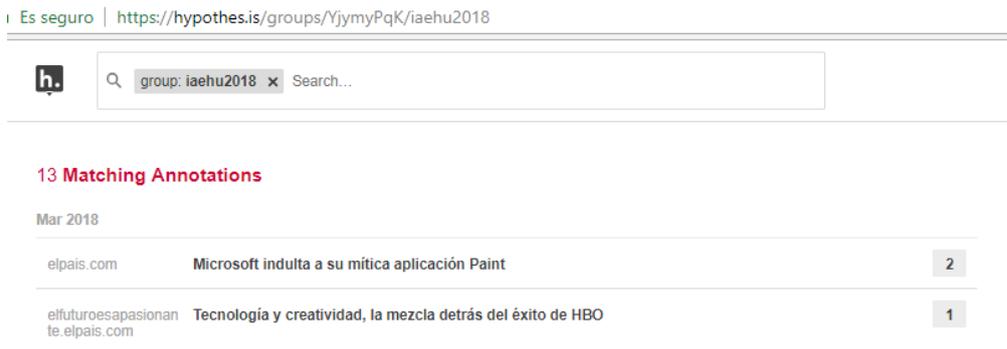


Ilustración 51. Prueba 2

### 6.1.3. Prueba 3

**Descripción:** Comprobar que al acceder a un grupo de exámenes en Hypothesis aparezcan los dos botones para visualizar los gráficos.

**Entrada:** El usuario accede a un grupo de Hypothesis.

**Salida esperada:** Se añaden dos botones.

**Salida real:** Salida esperada

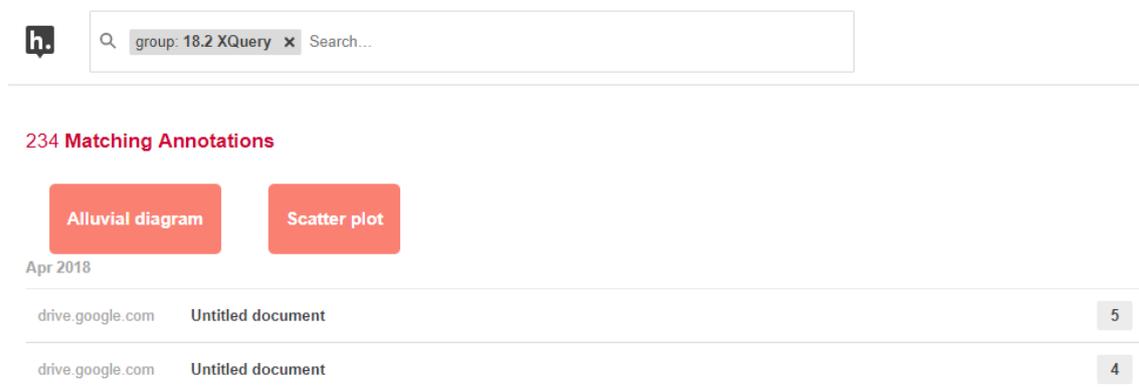


Ilustración 52. Prueba 3

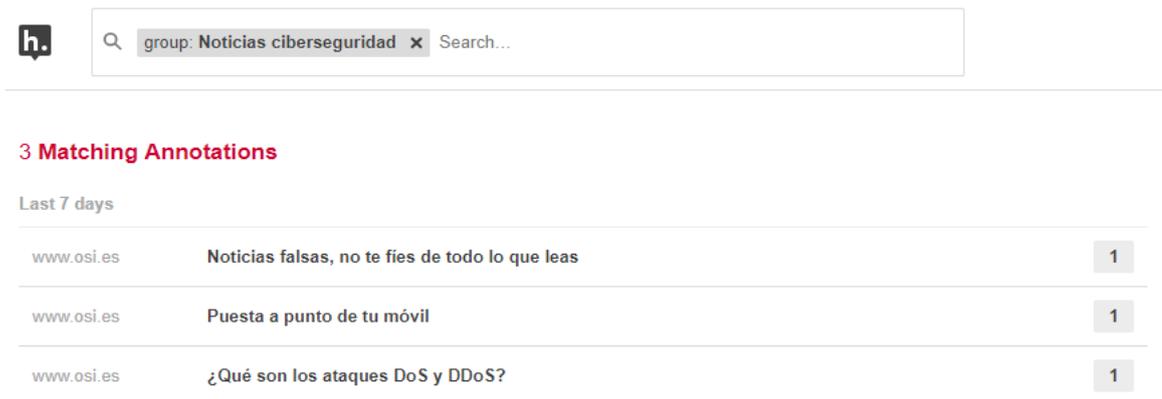
#### 6.1.4. Prueba 4

**Descripción:** Comprobar que al acceder a un grupo que no sea de exámenes no aparezcan los dos botones para visualizar los gráficos.

**Entrada:** El usuario accede a un grupo de hypothesis.

**Salida esperada:** No se añaden los dos botones.

**Salida real:** Salida esperada.



The screenshot shows a search bar with the text "group: Noticias ciberseguridad" and a search icon. Below the search bar, there is a section titled "3 Matching Annotations" with a sub-header "Last 7 days". A table lists three annotations from "www.osi.es":

Source	Annotation	Count
www.osi.es	Noticias falsas, no te fies de todo lo que leas	1
www.osi.es	Puesta a punto de tu móvil	1
www.osi.es	¿Qué son los ataques DoS y DDoS?	1

Ilustración 53. Prueba 4

#### 6.1.5. Prueba 5

**Descripción:** Comprobar que se visualiza el diagrama de dispersión de un examen.

**Entrada:** El usuario selecciona el botón "Diagrama de dispersión".

**Salida esperada:** Se dibuja el diagrama de dispersión.

**Salida real:** Salida esperada.

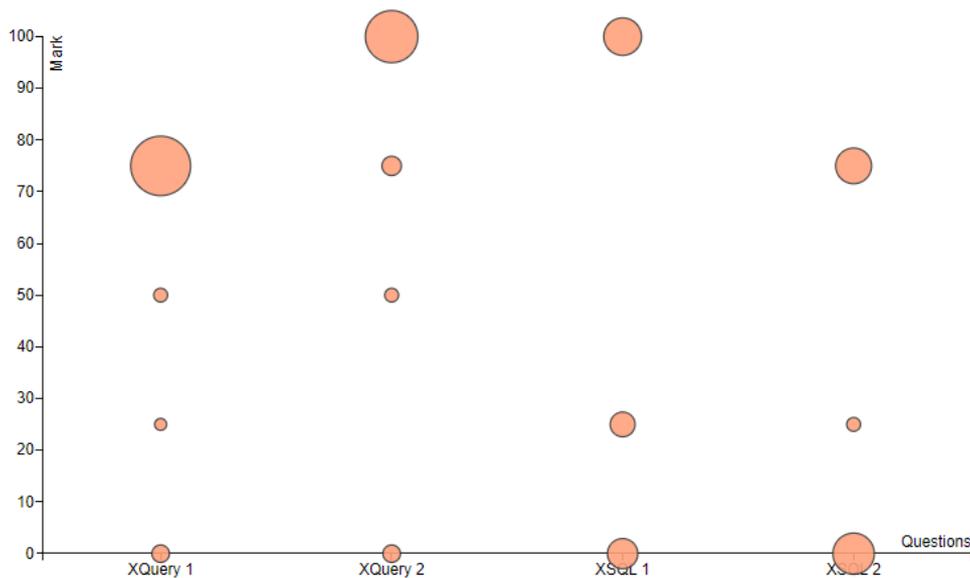


Ilustración 54. Prueba 5





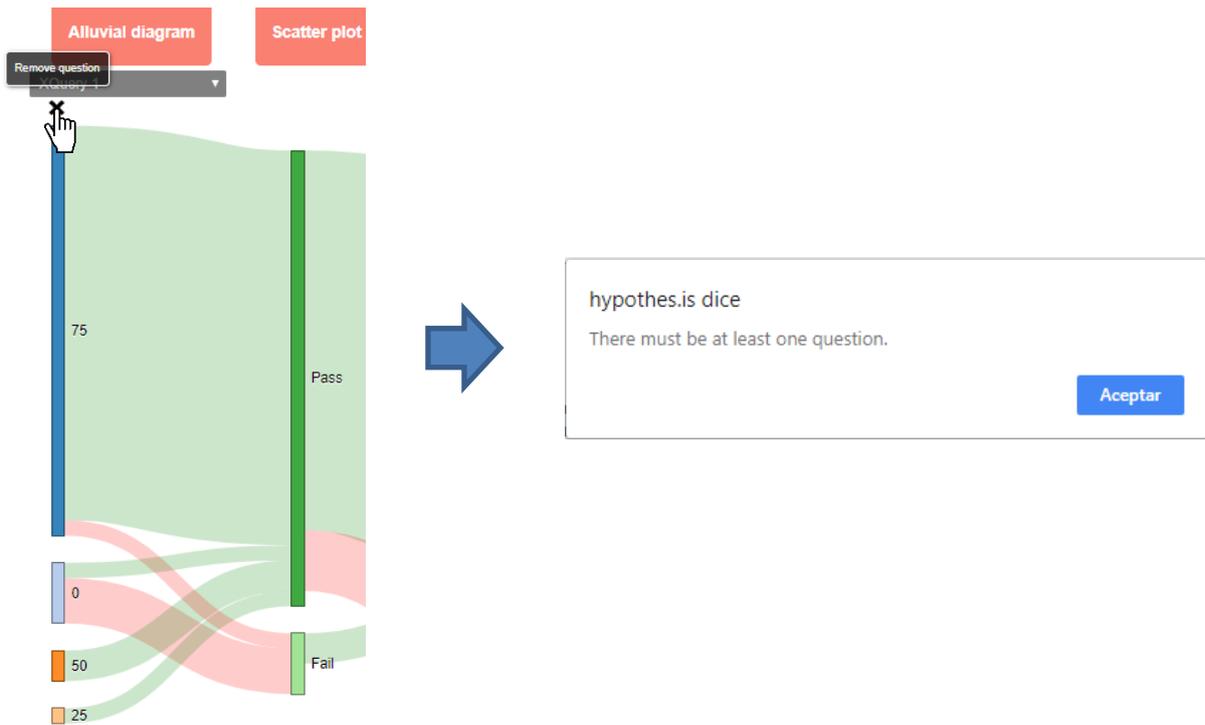


Ilustración 57. Prueba 9

#### 6.1.10. Prueba 10

**Descripción:** Comprobar que se puede intercambiar dos preguntas en el diagrama alluvial.

**Entrada:** El usuario selecciona la lista desplegable de una pregunta y selecciona con que otra quiere intercambiarla.

**Salida esperada:** Se dibuja el diagrama alluvial con las posiciones de las preguntas intercambiadas.

**Salida real:** Salida esperada.

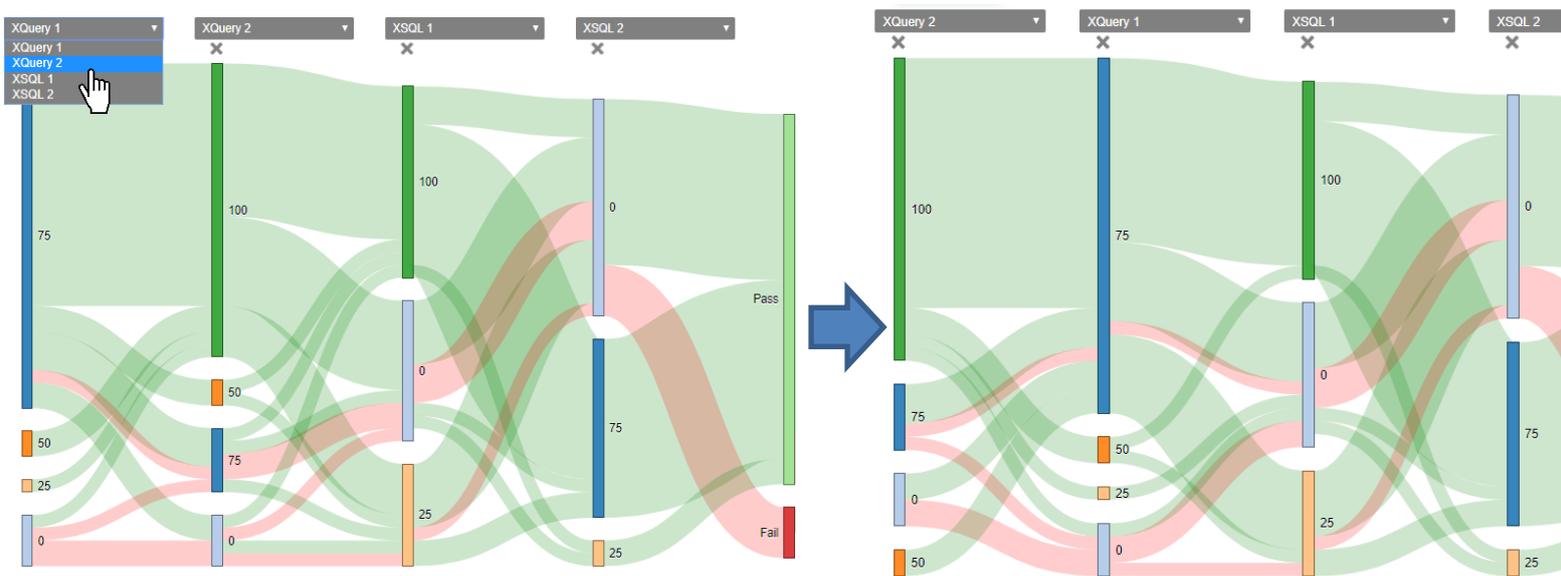


Ilustración 58. Prueba 10

### 6.1.11. Prueba 11

**Descripción:** Comprobar que se puede eliminar un examen del diagrama alluvial.

**Entrada:** El usuario selecciona el botón de eliminar examen.

**Salida esperada:** Se dibuja el diagrama sin el examen seleccionado.

**Salida real:** Salida esperada.

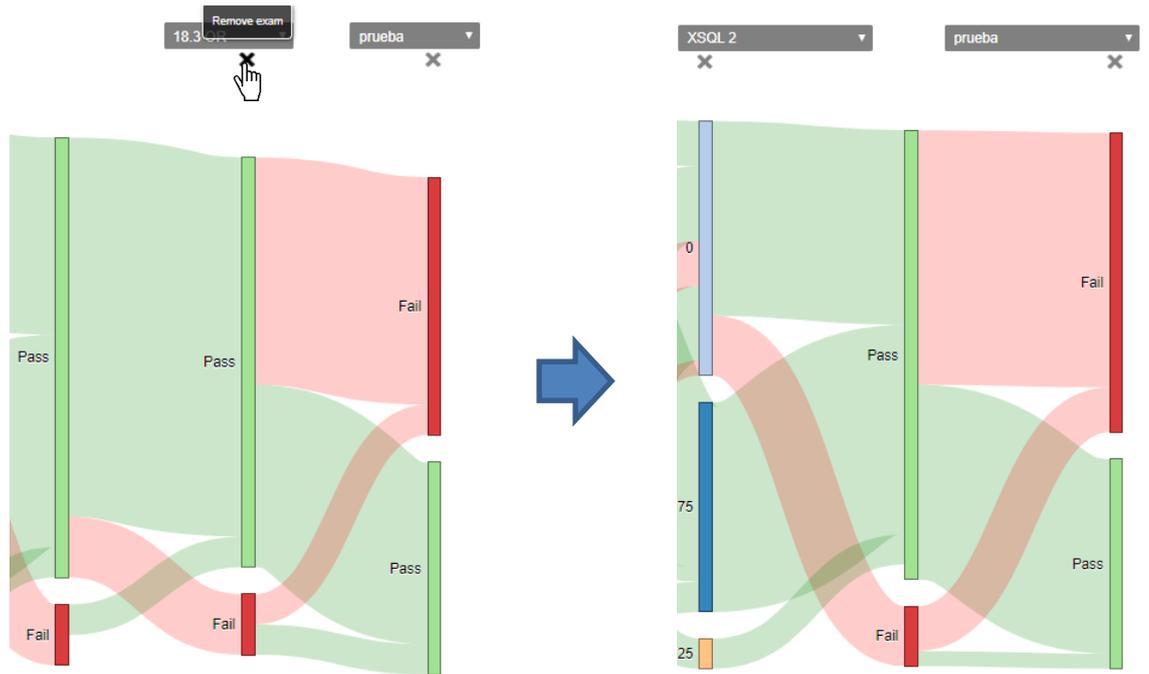


Ilustración 59. Prueba 11

### 6.1.12. Prueba 12

**Descripción:** Comprobar que se puede intercambiar dos exámenes en el diagrama alluvial.

**Entrada:** El usuario selecciona la lista desplegable de un examen y selecciona con que otra quiere intercambiarla.

**Salida esperada:** Se dibuja el diagrama con las posiciones de las preguntas intercambiadas.

**Salida real:** Salida esperada.

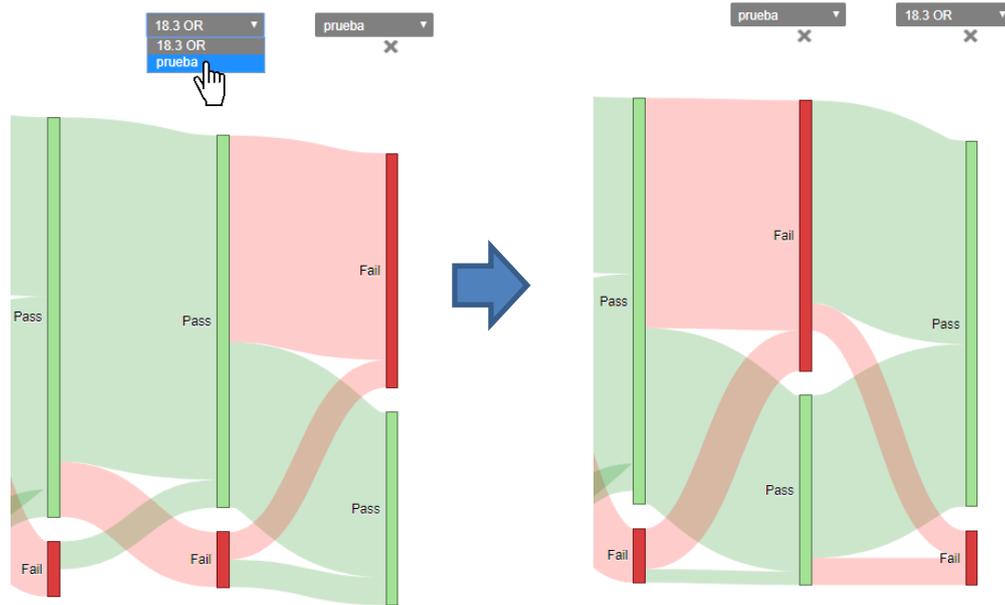


Ilustración 60. Prueba 12

### 6.1.13. Prueba 13

**Descripción:** Comprobar que se refresca el diagrama alluvial.

**Entrada:** Habiendo un diagrama alluvial modificado, el usuario selecciona el botón de “Alluvial diagram”.

**Salida esperada:** Se vuelve a dibujar el diagrama original deshaciendo los cambios realizados.

**Salida real:** Salida esperada.

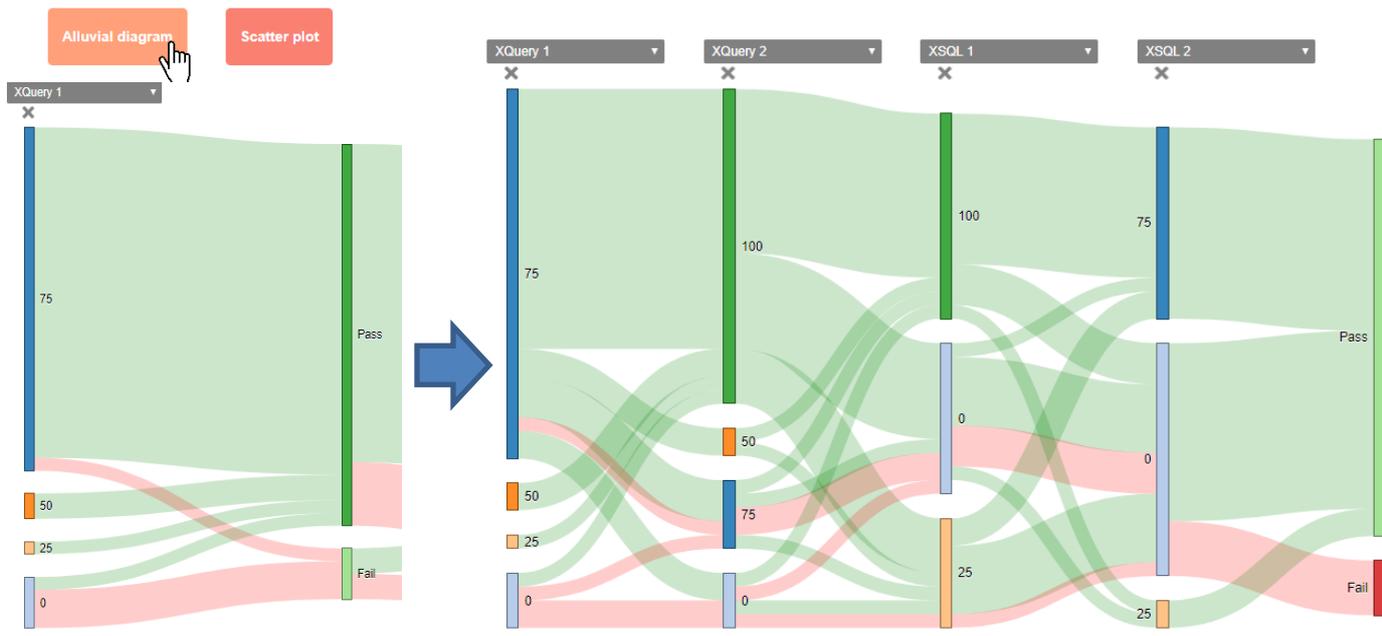


Ilustración 61. Prueba 13

#### 6.1.14. Prueba 14

**Descripción:** Comprobar que el diagrama alluvial no se visualice en un grupo de examen que no contiene ninguna anotación.

**Entrada:** El usuario selecciona el botón de “Alluvial diagram”.

**Salida esperada:** El diagrama alluvial no se muestra.

**Salida real:** Salida no esperada. Da una serie de errores y muestra elementos que no deberían aparecer. Tras realizar varias modificaciones se ha vuelto a repetir la prueba y se ha conseguido la salida esperada.

#### 6.1.15. Prueba 15

**Descripción:** Comprobar que el diagrama alluvial desaparezca cuando el usuario quiere ver el diagrama de dispersión.

**Entrada:** El usuario selecciona el botón de “Scatter plot” mientras esta visualizando el diagrama alluvial.

**Salida esperada:** El diagrama alluvial deja de mostrarse y se visualiza el diagrama de dispersión.

**Salida real:** Salida esperada.

#### 6.1.16. Prueba 16

**Descripción:** Comprobar que el diagrama de dispersión desaparezca cuando el usuario quiere ver el diagrama alluvial.

**Entrada:** El usuario selecciona el botón de “Alluvial diagram” mientras esta visualizando el diagrama de dispersión.

**Salida esperada:** El diagrama de dispersión deja de mostrarse y se visualiza el diagrama alluvial.

**Salida real:** Salida esperada.

#### 6.1.17. Prueba 17

**Descripción:** Comprobar que se visualiza el diagrama de araña.

**Entrada:** El usuario accede a la carpeta de los exámenes de un alumno en Google Drive.

**Salida esperada:** Se dibuja el diagrama de araña.

**Salida real:** Salida esperada.

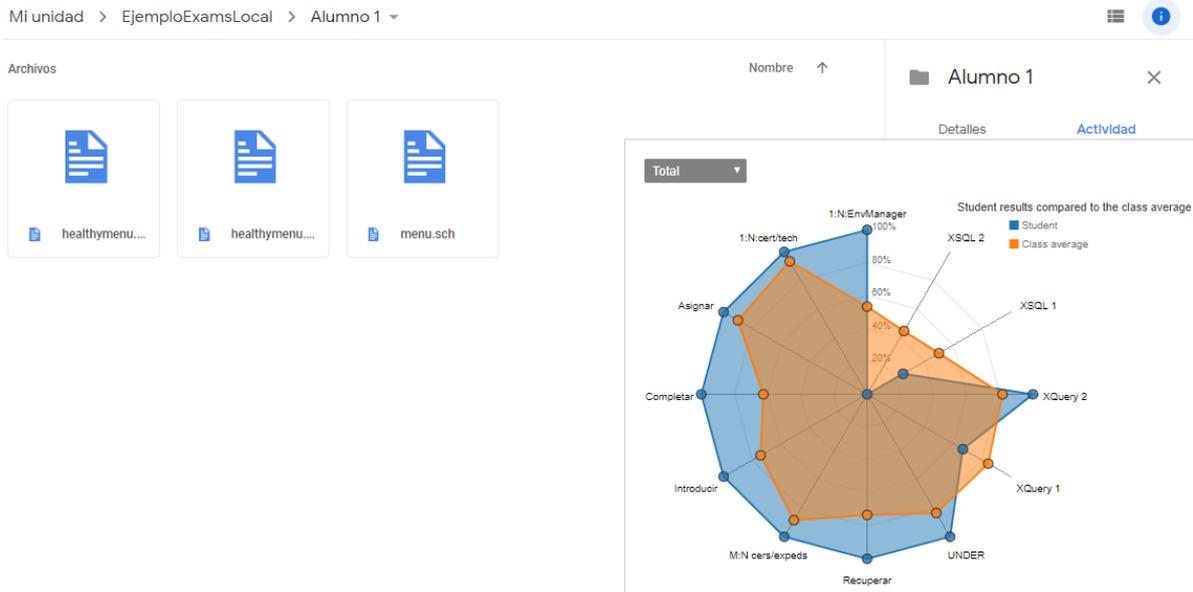


Ilustración 62. Prueba 17

#### 6.1.18. Prueba 18

**Descripción:** Comprobar que al acceder a una carpeta que no sea de un alumno no se visualice el diagrama de araña.

**Entrada:** El usuario accede a una carpeta que no sea de un alumno.

**Salida esperada:** No se dibuja el diagrama de araña.

**Salida real:** Salida esperada.

#### 6.1.19. Prueba 19

**Descripción:** Comprobar la visualización de un examen en el diagrama de araña.

**Entrada:** El usuario selecciona un examen de la lista desplegable.

**Salida esperada:** Se dibuja el diagrama de araña con únicamente las preguntas del examen.

**Salida real:** Salida no esperada. La lista desplegable no funciona ya que Google Drive sobrescribe su comportamiento. Tras realizar varias modificaciones se ha vuelto a repetir la prueba y ya funciona correctamente.

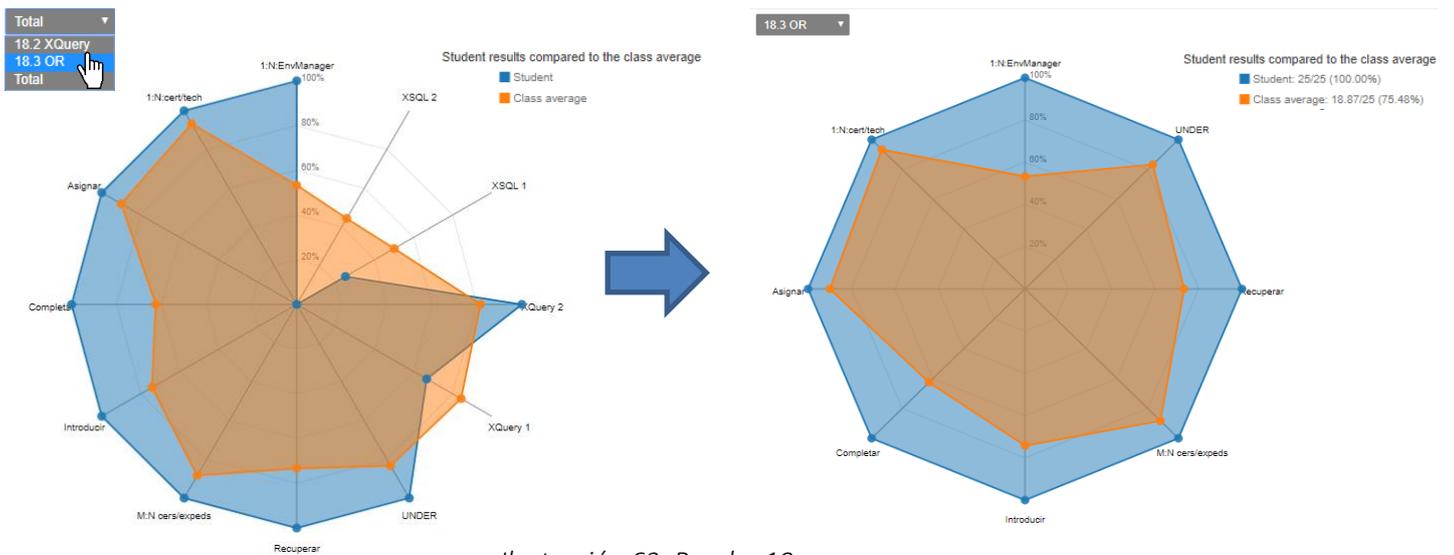


Ilustración 63. Prueba 19

### 6.1.20. Prueba 20

**Descripción:** Comprobar que al salir de la carpeta de un alumno el diagrama de araña desaparezca.

**Entrada:** El usuario sale de la carpeta de un alumno.

**Salida esperada:** El diagrama de araña desaparece del lateral inferior derecho de la página.

**Salida real:** Salida esperada.

# 7. Seguimiento del proyecto

---

Este capítulo está dedicado al seguimiento y control del proyecto. En él se muestra el diagrama de Gantt que representa el desarrollo real del proyecto y se analizan las dedicaciones reales en comparación con las estimadas.

El alcance del proyecto no ha variado a lo largo del desarrollo del mismo y aunque en un principio el proyecto estaba planificado para que finalizase en junio, finalmente, tras ver que no iba a ser posible cumplirlo, hubo que retrasar su entrega hasta la convocatoria de septiembre para poder completar satisfactoriamente el trabajo, haciendo que el número de horas realizadas hayan aumentado considerablemente superando las estimadas en un principio.

La siguiente tabla muestra la comparación entre las horas estimadas y las dedicadas realmente en el proyecto.

Tareas	Dedicación estimada	Estimación real	Diferencia
Gestión	36	32	-4
Análisis y diseño	30	26	-4
Gestión del conocimiento	30	35	+5
Desarrollo	100	145	+45
Documentación	77	92	+15
Pruebas	10	20	+10
Defensa del proyecto	25	*	*
<b>TOTAL</b>	<b>308</b>	<b>350*</b>	<b>+67</b>

\* No se han contabilizado las horas reales de la defensa del proyecto debido a que no ha finalizado

Tabla 8. Comparativa entre la dedicación estimada y real

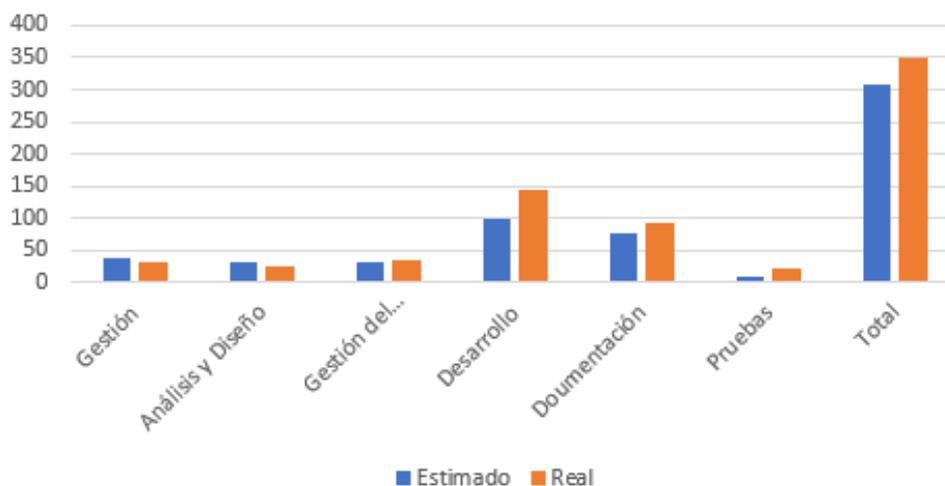


Ilustración 64. Gráfico comparativa dedicación estimada y real

El resultado ha sido una desviación de 67 horas adicionales (sin contar la defensa del proyecto) respecto al tiempo estimado. Esto se debe en gran medida a los problemas que ha habido a lo largo del proyecto como se ha descrito en el capítulo de implementación y la dificultad de la implementación de alguno de los diagramas con la librería d3.js, como por ejemplo el diagrama alluvial por todas las funcionalidades que tenía que hacer que su implementación fuera compleja.

Además del desarrollo, la estimación de las horas para realizar la documentación también resultó ser escasa a la hora de realizar la memoria. Debido a los múltiples cambios que hubo que hacer en ella, el número de horas aumentaron considerablemente.

Por último, debido a los problemas que ha habido a lo largo del desarrollo, las pruebas también tuvieron una dedicación mayor a la esperada.

El resto de tareas no han sufrido grandes diferencias respecto a la dedicación estimada en la planificación.

A continuación, se muestra el diagrama de Gantt que se ha obtenido tras finalizar el desarrollo del proyecto.

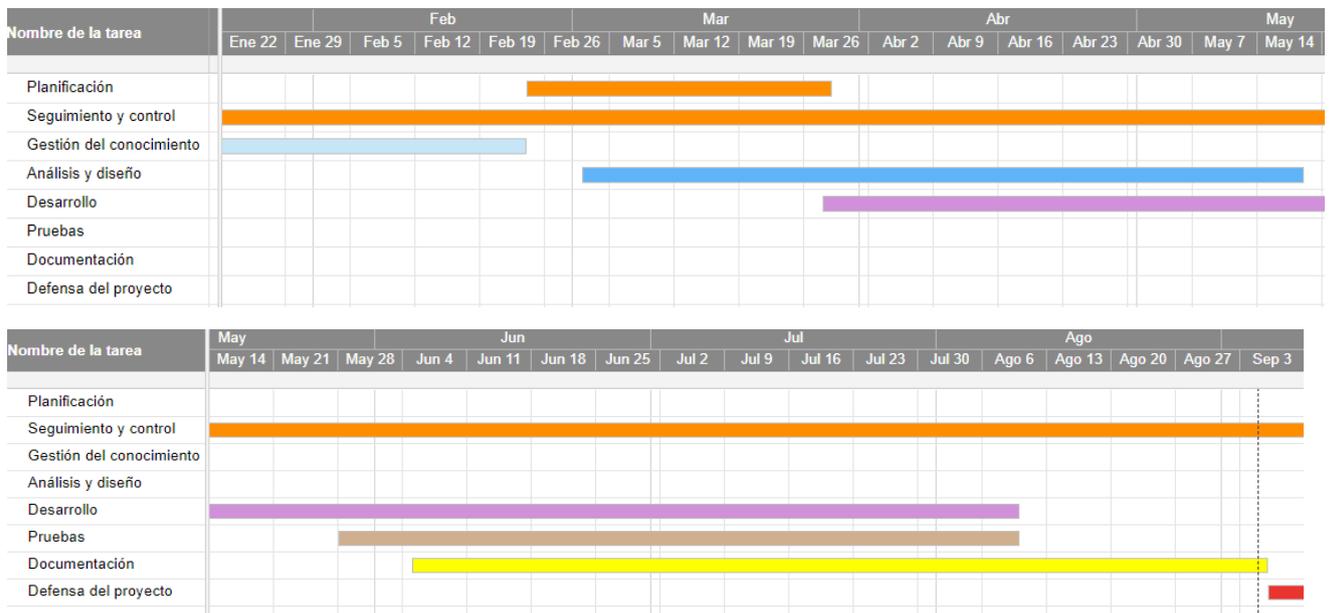


Ilustración 65. Versión final diagrama de Gantt

# 8. Conclusiones

---

En este capítulo se presentan las conclusiones del proyecto, para ello se analizan varios aspectos del proyecto y su desarrollo. Primero se hace una valoración del proyecto realizado. Posteriormente se valoran los conocimientos adquiridos y la experiencia personal. Finalmente se proponen puntos de mejora para la extensión web.

## 8.1. Conclusión del proyecto

Han sido 8 meses largos y duros que, a pesar de todas de todas las dificultades y problemas encontrados, se han conseguido superar y lograr cumplir con todos los requerimientos que inicialmente se habían establecido por lo que se puede concluir que el proyecto ha sido desarrollado de forma exitosa.

## 8.2. Conclusiones del proyecto

En cuanto a las conclusiones personales, sin duda el proyecto ha merecido la pena por todo lo aprendido y logrado en estos últimos meses. A pesar de haber sido muy duro y lleno de dificultades, no se puede estar más satisfecho del trabajo personal.

El proyecto ha resultado muy beneficioso por todos los nuevos conceptos que se han adquirido. De cara al futuro, con seguridad será de gran utilidad el aprendizaje de tecnologías como las extensiones web, la aumentación web, la visualización de datos gracias a una de las librerías más usadas hoy en día como lo es D3.js, lodash.js o las APIs. Además de un mayor conocimiento y manejo de las que ya conocía como: JavaScript, CSS y HTML.

Uno de las principales preocupaciones existentes durante el desarrollo ha sido el uso de estas nuevas tecnologías, puesto que conllevan mayor tiempo debido a la necesidad de su aprendizaje. Sin embargo, gracias a la cantidad de tutoriales y la gran comunidad existente en torno a estas tecnologías ha facilitado la tarea.

## 8.3. Mejoras y líneas futuras

Aunque el proyecto haya llegado a su fin habiendo completado todas las funcionalidades acordadas aún es posible realizar varias mejoras en el, entre lo que se encuentran:

- Comprobar su correcto funcionamiento en otros navegadores webs distintos a Google Chrome.
- Integrar la extensión en Mark&Go.
- A pesar de que la cantidad de texto es mínima en la extensión, sería interesante poder traducirla a otras lenguas que se imparten en la facultad como el euskera o castellano.
- Se podría considerar la inclusión de otros gráficos que no se hayan considerado en este proyecto.
- Añadir la opción de poder trabajar con la extensión en más de una asignatura ya que hay profesores que imparten varias clases.
- Poder utilizar la extensión para corregir, además de exámenes, trabajos o ejercicios de alumnos.



# 9. Bibliografía

---

En este capítulo se muestran las referencias bibliográficas más destacadas en relación al Trabajo de Fin de Grado.

- [1] Evaluación continua. [https://es.wikipedia.org/wiki/Evaluaci%C3%B3n\\_continua](https://es.wikipedia.org/wiki/Evaluaci%C3%B3n_continua)
- [2] Rúbrica. [https://es.wikipedia.org/wiki/R%C3%BAbrica\\_\(docencia\)](https://es.wikipedia.org/wiki/R%C3%BAbrica_(docencia))
- [3] Anotaciones web. [https://en.wikipedia.org/wiki/Web\\_annotation](https://en.wikipedia.org/wiki/Web_annotation)
- [4] Annotary. <https://chrome.google.com/webstore/detail/annotary/fncckipieakpgpjiifiadfiigdeanbond>
- [5] Evernote Web Clipper. <https://evernote.com/intl/es/products/webclipper>
- [6] Diigo. <https://www.diigo.com/>
- [7] A.nnotate. <http://a.nnotate.com/>
- [8] Hypothesis. <https://web.hypothes.is>
- [9] World Wide Web Consortium (W3C). <https://www.w3.org>
- [10] Chrome Web Store. Mark&Go. <https://chrome.google.com/webstore/detail/markgo/kjedcndgienemldgjjijnhdhfoacfa>
- [11] Plantilla Google Spreadsheets. <https://docs.google.com/spreadsheets/d/13Q-Lf8DvRuhPD7SkB5J17UGewmpk3b6lb-bAhPRrB6l/edit?usp=sharing>
- [12] Visualización de datos. [https://es.wikipedia.org/wiki/Visualizaci%C3%B3n\\_de\\_datos](https://es.wikipedia.org/wiki/Visualizaci%C3%B3n_de_datos)
- [13] Diagrama alluvial. [https://en.wikipedia.org/wiki/Alluvial\\_diagram](https://en.wikipedia.org/wiki/Alluvial_diagram)
- [14] Ejemplo diagrama alluvial. <https://cran.r-project.org/web/packages/alluvial/vignettes/alluvial.html>
- [15] Diagrama de araña. [https://en.wikipedia.org/wiki/Radar\\_chart](https://en.wikipedia.org/wiki/Radar_chart)
- [16] Ejemplo diagrama de araña. <http://bl.ocks.org/nbremer/6506614>
- [17] Diagrama de dispersión. [https://es.wikipedia.org/wiki/Diagrama\\_de\\_dispersi%C3%B3n](https://es.wikipedia.org/wiki/Diagrama_de_dispersi%C3%B3n)
- [18] Ejemplo diagrama de dispersión. <https://www.mathsisfun.com/data/scatter-xy-plots.html>
- [19] Aumentación web. [https://en.wikipedia.org/wiki/Augmented\\_browsing](https://en.wikipedia.org/wiki/Augmented_browsing)
- [20] Api de Hypothesis. <http://h.readthedocs.io/en/latest/api-reference/>
- [21] ECMAScript 6. <http://es6-features.org>
- [22] D3.js. <https://d3js.org/>
- [23] Lodash.js. <https://lodash.com/>
- [24] Foro Google Drive. <https://productforums.google.com/forum/#!forum/drive>
- [25] Stack Overflow. <https://stackoverflow.com/>
- [26] Caja negra. [https://es.wikipedia.org/wiki/Caja\\_negra\\_\(sistemas\)](https://es.wikipedia.org/wiki/Caja_negra_(sistemas))

# Anexo A.

## Desarrollo

---

En este anexo se profundiza lo visto en el capítulo de desarrollo y se explica paso a paso las transformaciones que se han ido realizando a las estructuras de datos para obtener cada uno de los diagramas.

## A.1. Diagrama alluvial

Una vez obtenidas las anotaciones de la llamada de la API, solo hace falta conseguir la estructura de datos necesaria para dibujar este tipo de diagramas. Para ello, es necesario ir haciendo distintas transformaciones a partir de las anotaciones obtenidas e ir creando estructuras intermedias. La estructura que se está buscando ha de tener de la siguiente forma:

```
{
  "nodes": [
    {
      "id": string,
      "visualization": string
    }
  ],
  "links": [
    {
      "source": string,
      "target": string,
      "value": number,
      "finalResult": string
    }
  ]
}
```

Ilustración 66. Diagrama alluvial: estructura final

- nodes: Escala de calificaciones de cada pregunta según los criterios de rúbrica del examen. Contiene los atributos "id" y "visualization" para identificarlo y que valor mostrar en el diagrama respectivamente.
- links: Lista de enlaces que conectan las calificaciones de dos preguntas. Los enlaces tienen la siguiente estructura:
  - source: Indica la "id" de la calificación (nodo) de la pregunta de origen.
  - target: Indica la "id" de la calificación (nodo) de la pregunta de destino.
  - finalResult: Indica si aprobaron o no el examen. Puede tomar dos valores posibles: "Pass" (Aprobado) o "Fail" (Suspense).
  - value: Indica el número de alumnos que sacaron como notas las indicadas en "source" y "target", y obtuvieron el resultado final indicado.

### Obtener calificaciones (nodos)

El primer objetivo va a ser obtener las calificaciones de cada una de las preguntas. Para ello, lo primero que hay que hacer es filtrar las anotaciones ya que inicialmente, además de las anotaciones que corresponden a las preguntas de los alumnos, existen anotaciones con otras etiquetas que no son necesarias. Para ello se mantendrán solo las anotaciones cuyo atributo "tags" sea una lista con dos valores que el primero empiece por "exam:isCriteriaOf:" (hace referencia a la pregunta) y el segundo por "exam:mark:" (indica la nota que obtuvo).

Una vez filtrado las anotaciones, hay que quedarse solo con las etiquetas (atributo "tags") de las anotaciones, es decir, el resultado va a ser una lista de pares de etiquetas.

```
etiquetas= [[string,string]]
```

Por ejemplo:

```
[
  ["exam:isCriterioOf:pregunta1", "exam:mark:50"],
  ["exam:isCriterioOf:pregunta1", "exam:mark:0"],
  ["exam:isCriterioOf:pregunta2", "exam:mark:25"],
  ["exam:isCriterioOf:pregunta2", "exam:mark:10"]
]
```

A raíz de la lista de etiquetas, se obtienen las preguntas del examen cogiendo el primer valor de cada par de la lista y eliminando el prefijo "exam:isCriterioOf:" de cada uno de ellos. Por ejemplo:

```
preguntas=["pregunta1", "pregunta2", "pregunta3","pregunta4"]
```

El siguiente paso es transformar la lista de tuplas de etiquetas en una lista de objetos nodos. Esta estructura está formada por una "id" para distinguirla del resto que será compuesto por la concatenación del nombre de la pregunta y la calificación en dicho ejercicio, además de la visualización en el diagrama de dicha calificación.

```
[
  {
    "id": string,
    "visualization": string
  }
]
```

Por ejemplo:

```
[
  {
    "id": "pregunta1:50",
    "visualization": "50"
  },
  {
    "id": "pregunta1:0",
    "visualization": "0"
  },
  {
    "id": "pregunta2:20",
    "visualization": "20"
  }
]
```

La lista tendrá objetos repetidos que hay que eliminar. Esto se debe a que un alumno en un ejercicio puedes tener varias anotaciones (en todas con las mismas etiquetas) debido a que puede haber más de una evidencia por lo que hay que deshacerse de estas.

Por último, se concatena a la lista dos objetos más que corresponden al aprobado y al suspenso. Para la "id" se concatena la ID del grupo con el resultado. Por ejemplo:

```
[
  {
    "id": "xDa7sz:Pass",
    "visualization": "Pass",
  },
  {
    "id": "xDa7sz:Fail",
    "visualization": "Fail"
  }
]
```

Con esto último, ya están los nodos del diagrama (sin contar con los de los exámenes) y el siguiente paso es obtener la lista de enlaces.

### Obtener enlaces

Primero, hay que recuperar la lista de anotaciones del principio y transformarlo de tal forma que la nueva estructura tenga solo los atributos que interesan. Esta estructura estará compuesta por tres atributos siendo la primera la URI del examen del alumno donde se han realizado las anotaciones, la pregunta y la nota obtenida. Esta lista se llamará “preguntasAlumnos”.

```
[
  {
    "uri": string,
    "question": string,
    "mark": CriterioRubricaType*
  }
]
```

\* *CriterioRubricaType* es un subconjunto del tipo number que no va a admitir cualquier valor, sino solo unos muy concretos para poder referirse a las posibles notas para una pregunta en concreto.

Por ejemplo:

```
preguntasAlumnos=
[
  {
    "uri": "drive.google.com/drive/folders/xxxxx1",
    "question": "pregunta1",
    "mark": 50
  },
  {
    "uri": "drive.google.com/drive/folders/xxxxx2",
    "question": "pregunta1",
    "mark": 20
  }
]
```

Como se ha realizado a la hora de obtener los nodos, es necesario eliminar los objetos repetidos (cuyos atributos sean iguales) ya que un alumno puede tener varias anotaciones en una misma pregunta.

A partir de la lista preguntasAlumnos, hay que agrupar por el atributo URI (que hace referencia a un alumno) y sumar todas sus notas. En caso de haber aprobado, su resultado final en el examen será "Pass" (Aprobado); en caso contrario "Fail" (Suspenso).

```
[
  {
    "uri": string,
    "finalResult": string
  }
]
```

Por ejemplo:

```
resultadosAlumnos=
[
  {
    "uri": "drive.google.com/drive/folders/xxxxx1",
    "finalResult": "Pass"
  },
  {
    "uri": "drive.google.com/drive/folders/xxxxx2",
    "finalResult": "Fail"
  },
  {
    "uri": "drive.google.com/drive/folders/xxxxx3",
    "finalResult": "Pass"
  }
]
```

Una vez conocido el resultado final de cada alumno, se actualiza la lista preguntasAlumnos añadiendo un nuevo campo llamado "finalResult" con la información que se acaba de obtener en resultadosAlumnos. Por ejemplo:

```
preguntasAlumnos=
[
  {
    "uri": "drive.google.com/drive/folders/xxxxx1",
    "question": "pregunta1",
    "mark": 50,
    "finalResult": "Pass"
  },
  {
    "uri": "drive.google.com/drive/folders/xxxxx2",
    "question": "pregunta1",
    "mark": 20,
    "finalResult": "Fail"
  }
]
```

Este nuevo campo añadido es importante ya que se necesitará más en adelante para saber de qué color dibujar los enlaces del diagrama.

A continuación, se agrupa la lista preguntasAlumnos por pregunta. De esta manera, se puede saber para cada pregunta que nota ha sacado cada alumno. Por ejemplo:

```

{
  "pregunta1": [
    {
      "uri": "drive.google.com/drive/folders/xxxxx1",
      "question": "pregunta1",
      "mark": 50,
      "finalResult": "Pass"
    },
    {
      "uri": "drive.google.com/drive/folders/xxxxx2",
      "question": "pregunta1",
      "mark": 20,
      "finalResult": "Fail"
    }
  ],
  "pregunta2": [
    {
      "uri": "drive.google.com/drive/folders/xxxxx1",
      "question": "pregunta2",
      "mark": 25,
      "finalResult": "Pass"
    },
    {
      "uri": "drive.google.com/drive/folders/xxxxx1",
      "question": "pregunta2",
      "mark": 25,
      "finalResult": "Fail"
    }
  ]
}

```

Ahora, hay que crear los enlaces y para ello hay que relacionar pares de preguntas. Para ello se iterará (bucle exterior) tantas veces como preguntas del examen haya en la lista preguntas obtenidas al principio. En cada iteración se cogen las preguntas en las posiciones  $i$  e  $i+1$ . Por ejemplo, en la primera iteración, la primera y segunda pregunta serán la pregunta de origen y destino respectivamente.

Se obtienen las distintas notas (eliminando repetidos) que han obtenido los alumnos en la pregunta de origen.

```
notasPreguntaOrigen= [CriterioRubricaType]
```

```
notasPreguntaOrigen= [0, 20, 25, 50]
```

A continuación, se iterará (bucle interior) sobre la lista de notas que se acaba de obtener y para cada una de las notas, se obtiene qué alumnos, que aprobaron el examen, sacaron dicha nota y qué nota obtuvieron en la pregunta de destino. Se cuentan las distintas notas que se han obtenido en la pregunta de destino y se va creando una lista de objetos enlace.

```

enlacesAprobados=
[
  {
    "source": "pregunta1:50",
    "target": "pregunta2:25",
    "value": 8,
    "finalResult": "Pass"
  },
  {
    "source": "pregunta1:50",
    "target": "pregunta2:10",
    "value": 4,
    "finalResult": "Pass"
  }
]

```

Se vuelve a repetir el proceso anterior pero esta vez solo con los alumnos que suspendieron y se crean los enlaces de éstos. Finalmente se concatena las dos listas de enlaces y se repite este proceso en el bucle interior. El bucle exterior termina de iterar hasta que se llega a las últimas dos preguntas.

Lo último que falta es la relación de la última pregunta con el resultado final del examen: "Pass" y "Fail". Se obtiene el nombre de la última pregunta, se agrupa a los alumnos por nota y resultado final, y se cuenta cuantos alumnos hay en cada caso. Se crean los enlaces de la misma manera que antes.

```

[
  {
    "source": "pregunta5:20",
    "target": "xDa7sz:Pass",
    "value": 10,
    "finalResult": "Pass"
  },
  {
    "source": "pregunta5:20",
    "target": "xDa7sz:Fail",
    "value": 5,
    "finalResult": "Fail"
  }
]

```

Por último, hay que unir las dos listas obtenidas, la de los nodos y los enlaces, para lograr una estructura como la vista al principio. Con esta información ya se puede dibujar el diagrama alluvial, pero aún falta la segunda parte del diagrama que muestra las relaciones entre el resultado del examen del grupo actual con los resultados del resto de exámenes. Este diagrama se realiza en dos pasos ya que la obtención de los nodos y enlaces del resto de exámenes requiere de más tiempo por lo que primero se visualiza el diagrama sin el resto de exámenes y después se actualiza.

## Obtener los enlaces del resto de exámenes

Lo primero es obtener todos los grupos de Hypothesis del usuario mediante la API y de cada uno obtener una anotación (se excluye el grupo del examen actual). Si la anotación contiene etiquetas con el prefijo "exam:" significa que el grupo es relativo a un examen. En ese caso, cuando se encuentra el primer grupo relativo de un examen se obtienen todas sus anotaciones y se realizan las mismas transformaciones que se han realizado anteriormente para saber si aprobó o no cada alumno.

```
[
  {
    "uri": "drive.google.com/drive/folders/xxxxx1",
    "finalResult": "Pass"
  },
  {
    "uri": "drive.google.com/drive/folders/xxxxx2",
    "finalResult": "Pass"
  },
  {
    "uri": "drive.google.com/drive/folders/xxxxx3",
    "finalResult": "Fail"
  }
]
```

Estos resultados se añaden junto a los resultados del examen actual obtenidos al principio (lista "resultadosAlumnos") consiguiendo la siguiente estructura.

```
[
  {
    "uri": string,
    "originExamResult": string,
    "targetExamResult": string
  }
]
```

Por ejemplo:

```
[
  {
    "uri": "drive.google.com/drive/folders/xxxxx1",
    "originExamResult": "Pass",
    "targetExamResult": "Pass"
  },
  {
    "uri": "drive.google.com/drive/folders/xxxxx2",
    "originExamResult": "Fail",
    "targetExamResult": "Pass"
  },
  {
    "uri": "drive.google.com/drive/folders/xxxxx3",
    "originExamResult": "Pass",
    "targetExamResult": "Fail"
  }
]
```

A continuación, se obtienen cuantos alumnos que aprobaron el examen actual, en el siguiente aprobaron o suspendieron y lo mismo con los alumnos que suspendieron. De esta manera se generan los enlaces:

```
[
  {
    "source": "x0a7sz:Pass",
    "target": "d84fCs:Pass",
    "value": 12,
    "finalResult": "Pass"
  },
  {
    "source": "x0a7sz:Pass",
    "target": "d84fCs:Fail",
    "value": 10,
    "finalResult": "Fail"
  },
  {
    "source": "x0a7sz:Pass",
    "target": "d84fCs:Pass",
    "value": 3,
    "finalResult": "Pass"
  }
]
```

Al mismo tiempo también se crean los nodos de aprobado y suspenso del grupo de destino:

```
[
  {
    "id": "d84fCs:Pass",
    "visualization": "Pass"
  },
  {
    "id": "d84fCs:Fail",
    "visualization": "Fail"
  }
]
```

Se vuelve a repetir el proceso esta vez tomando los resultados del examen de destino como origen y como destino el siguiente examen que se encuentre.

Una vez obtenido todas los nodos y enlaces, se actualiza el grafo obtenido anteriormente y se vuelve a visualizar el diagrama.

## A.2. Gráfico de araña

En este apartado se va a explicar la consecución de la estructura de datos necesaria para dibujar un diagrama de araña. Para ello, es necesario ir haciendo distintas transformaciones a partir de las anotaciones obtenidas e ir creando estructuras intermedias. La estructura que se está buscando ha de tener la siguiente forma:

```
[
  [
    {
      "axis": string,
      "mark": number
      "maximumValue": CriterioRubricaType,*
      "porcentage": number,
      "group": string
    }
  ]
]
```

\* *CriterioRubricaType* es un subconjunto del tipo *number* que no va a admitir cualquier valor, sino solo unos muy concretos para poder referirse a las posibles notas para una pregunta en concreto.

Esta estructura está formada por un array de 2 arrays de objetos. Estos objetos corresponden a cada una de las preguntas con el siguiente formato:

- *axis*: Nombre del eje que corresponde a una pregunta de examen.
- *mark*: Nota del alumno y media de la clase en la pregunta en el primer y segundo array respectivamente.
- *maximumValue*: Nota máxima que se puede obtener en la pregunta.
- *porcentage*: Porcentaje de la puntuación de la pregunta obtenida.
- *group*: ID del examen al que pertenece la pregunta. Corresponde a la ID Grupo de Hypothesis del examen.

El primero de los dos arrays corresponden a las preguntas del alumno mientras que el segundo a las medias de los alumnos de la clase.

Nada más acceder a la carpeta del alumno en Google Drive, se obtiene la URL, se guarda en la variable "alumno" y se hace una llamada a la API de Hypothesis para obtener las anotaciones del alumno en los grupos del profesor.

[https://hypothes.is/api/search?uri=+ alumno +&limit=200](https://hypothes.is/api/search?uri=+alumno +&limit=200)

Para empezar, hay que filtrar las anotaciones obtenidas al igual que se ha hecho en el diagrama alluvial por lo que se mantendrán solo las anotaciones cuyo atributo "tags" sea una lista con dos valores que el primero empiece por "exam:isCriteriaOf:" (hace referencia a la pregunta) y el segundo por "exam:mark:" (indica la nota que obtuvo).

A continuación, se crea una nueva estructura que tenga solo los atributos que interesan de una anotación. Esta estructura estará compuesta por tres atributos: la pregunta, la nota obtenida y el grupo (examen) al que pertenece.

```
[
  {
    "question": string,
    "mark": CriterioRubricaType,
    "exam" : string
  }
]
```

Ejemplo:

```
[
  {
    "question": "pregunta1",
    "mark": 25,
    "exam": "examen1"
  },
  {
    "question": "pregunta2",
    "mark": 50,
    "exam": "examen1"
  },
  {
    "question": "pregunta1",
    "mark": 4,
    "exam": "examen2"
  },
  {
    "question": "pregunta2",
    "mark": 1,
    "exam": "examen2"
  }
]
```

La lista puede tener objetos repetidos que haya que eliminar. Esto se debe a que un alumno en un ejercicio puede tener varias anotaciones (en todas con las mismas etiquetas) por lo que hay que deshacerse de estas.

A continuación, modificamos la estructura de las preguntas del alumno añadiéndole sobre cuánto es la pregunta (más tarde se explica) y el porcentaje de la nota obtenida.

```
[
  {
    "axis":string,
    "mark": CriterioRubricaType,
    "maximumValue": CriterioRubricaType,
    "porcentage": number,
    "group": string
  }
]
```

Ejemplo:

```
[
  {
    "axis": "pregunta1",
    "mark": 10,
    "maximumValue": 10,
    "percentage": 1,
    "group": "examen1"
  },
  {
    "axis": "pregunta2",
    "mark": 8,
    "maximumValue": 10,
    "percentage": 0.8,
    "group": "examen1"
  }
]
```

Falta saber las notas medias de la clase para cada uno de los grupos (exámenes). Para ello, se obtiene los distintos grupos de las anotaciones del alumno y para cada grupo se obtiene las notas medias mediante una función que se explicará a continuación.

Por ultimo las dos listas se concatenan y se obtiene la estructura final.

#### Obtener notas medias de un examen

Dado un grupo, se obtienen todas las anotaciones realizadas en dicho grupo y se filtran por tags como hemos hecho antes. Se crea un array de objetos que están compuestos por tres atributos siendo el primera la URI del examen del alumno, la pregunta y la nota obtenida.

Una vez eliminado los elementos repetidos, se agrupan por pregunta (atributo axis) y se calcula la media (atributo mark).

```
[
  {
    "axis": string,
    "mark": number
  }
]
```

Ejemplo:

```
[
  {
    "axis": "pregunta1",
    "mark": 7.5
  },
  {
    "axis": "pregunta2",
    "mark": 5.4
  }
]
```

Falta por obtener la puntuación máxima de cada ejercicio y para ello se llama a una función que la obtiene que se explicará más adelante. Una vez obtenida la puntuación máxima se obtiene la estructura final:

```
[
  {
    "axis": string,
    "mark": number,
    "maximumValue": CriterioRubricaType,
    "percentage": number,
    "group": string
  }
]
```

Tomando el último ejemplo tendríamos el siguiente resultado:

```
[
  {
    "axis": "pregunta1",
    "mark": 7.5,
    "maximumValue": 10,
    "percentage": 0.75,
    "group": "examen1"
  },
  {
    "axis": "pregunta2",
    "mark": 5.4,
    "maximumValue": 10,
    "percentage": 0.54,
    "group": "examen1"
  }
]
```

#### Obtener puntuación máxima de cada pregunta de un examen

Antes de empezar a corregir los exámenes, Mark&Go, a la hora de crear el grupo de Hypothesis, genera automáticamente las primeras anotaciones con la escala de calificaciones de cada pregunta que se han establecido en la hoja de cálculo. Estas, para diferenciarse del resto de anotaciones, tienen dos etiquetas "exam:mark:" y "exam:isCriteriaOf:" que como se puede apreciar están en orden inverso al resto. Primero se obtienen las anotaciones de un examen, se filtran por esas dos etiquetas, se agrupa por la etiqueta "exam:isCriteriaOf:" y en cada caso se saca la etiqueta "exam:mark:" con valor más alto. Se obtiene la siguiente estructura:

```
[
  {
    "question": string,
    "maximumValue": CriterioRubricaType
  }
]
```

Por ejemplo:

```
[
  {
    "question": "pregunta1",
    "maximumValue": 50
  },
  {
    "question": "pregunta2",
    "maximumValue": 25
  },
  {
    "question": "pregunta3",
    "maximumValue": 50
  }
]
```

### A.3. Diagrama de dispersión sobre la evaluación continua

Para dibujar el diagrama de dispersión es necesario llegar a la siguiente estructura:

```
[
  {
    "exam": string,
    "mark": number,
    "numStudents": number
  }
]
```

- exam: Nombre del examen.
- mark: Nota obtenida.
- numStudents: Número de alumnos que han obtenido la nota en el examen indicado.

Lo primero es obtener todos los grupos de Hypothesis del usuario mediante la API y de cada uno obtener una anotación. Si la anotación contiene etiquetas con el prefijo “exam” significa que el grupo es relativo a un examen. En ese caso, se obtienen todas las anotaciones del grupo.

Estas anotaciones hay que filtrarlas, de igual manera que se ha realizado en los otros gráficos, por lo que se mantendrán solo las anotaciones cuyo atributo “tags” sea una lista con dos valores que el primero empiece por “exam:isCriteriaOf:” (hace referencia a la pregunta) y el segundo por “exam:mark:” (indica la nota que obtuvo).

A raíz de las anotaciones filtradas, se crea un array de objetos compuestos por tres atributos siendo el primera la URI del examen del alumno donde se han realizado las anotaciones, la pregunta y la nota obtenida.

```
{
  "uri" : string,
  "question" : string
  "mark" : CriterioRubricaType
}
```

Por ejemplo:

```
preguntasAlumnos=
[
  {
    "uri": "drive.google.com/drive/folders/xxxxx1",
    "question": "pregunta1",
    "numStudents": 50
  },
  {
    "uri": "drive.google.com/drive/folders/xxxxx2",
    "question": "pregunta1",
    "numStudents": 20
  }
]
```

La lista tendrá objetos repetidos que hay que eliminar. Esto se debe a que un alumno en un ejercicio puede tener varias anotaciones (en todas con las mismas etiquetas) por lo que hay que deshacerse de estas.

A continuación, se agrupan las preguntas por la URI de cada alumno, se suman las notas para obtener la nota final y estas se pasan sobre 10.

```
[
  {
    "uri": string,
    "result": number
  }
]
```

Por ejemplo,

```
[
  {
    "uri": "https://drive.google.com/drive/folders/xxxxx1",
    "result": 8.5
  },
  {
    "uri": "https://drive.google.com/drive/folders/xxxxx2",
    "result": 3
  },
  {
    "uri": "https://drive.google.com/drive/folders/xxxxx3",
    "result": 6.5
  }
]
```

Por último, se cuentan las distintas notas y se crea la estructura final añadiéndole un atributo indicando el nombre del examen actual. Así se obtiene la estructura deseada y se guarda en una lista.

```
[
  {
    "exam": string,
    "mark": number,
    "numStudents": number
  }
]
```

Por ejemplo,

```
[
  {
    "exam": "examen1",
    "mark": 6,
    "numStudents": 4
  },
  {
    "exam": "examen1",
    "question": 10,
    "numStudents": 2
  },
  {
    "exam": "examen1",
    "mark": 4,
    "numStudents": 5
  }
]
```

Una vez visitado todos los grupos del usuario, la lista resultante del proceso es enviada a la función que dibuja el diagrama de dispersión.

#### A.4. Diagrama de dispersión sobre un examen

Los pasos son similares al anterior diagrama de dispersión explicado. El objetivo es llegar a la siguiente estructura final:

```
[
  {
    "question": string,
    "mark": CriterioRubricaType*,
    "numAStudents": number
  }
]
```

\* *CriterioRubricaType* es un subconjunto del tipo *number* que no va a admitir cualquier valor, sino solo unos muy concretos para poder referirse a las posibles notas para una pregunta en concreto.

- question: Nombre de la pregunta.
- mark: Nota obtenida.
- numStudents: Número de alumnos que han obtenido la nota en la pregunta indicada.

Para empezar, se obtienen las anotaciones del grupo actual y se filtran por etiqueta. A continuación, se crea la ya conocida estructura compuesta por la URI del examen del alumno, la pregunta y la nota obtenida.

```
{
  "uri" : string,
  "question" : string
  "mark" : CriterioRubricaType
}
```

Por ejemplo:

```
preguntasAlumnos=
[
  {
    "uri": "drive.google.com/drive/folders/xxxxx1",
    "question": "pregunta1",
    "mark": 50
  },
  {
    "uri": "drive.google.com/drive/folders/xxxxx2",
    "question": "pregunta1",
    "mark": 20
  }
]
```

A continuación, se agrupa por pregunta y nota, y se cuenta el número de elementos en cada caso obteniendo la estructura final necesaria.

```
[
  {
    "question": string,
    "mark": CriterioRubricaType,
    "numAStudents": number
  }
]
```

Por ejemplo:

```
[
  {
    "question": "XQuery 1",
    "mark": 0,
    "numStudents": 8
  },
  {
    "question": "XQuery 1",
    "mark": 10,
    "numStudents": 10
  },
  {
    "question": "XQuery 1",
    "mark": 25,
    "numStudents": 12
  }
]
```



# Anexo B.

## Actas reuniones

---

En este capítulo se incluyen las actas de las reuniones realizadas a lo largo del proyecto. Se incluye las tareas realizadas desde la anterior reunión, los temas tratados en dicha reunión y las tareas a realizar para la próxima reunión.

## B.1. Acta 1

Fecha: 02/02/2018      Hora: 10:45      Duración: 30'

### Asistentes:

- Oscar
- Haritz
- Eduardo

### Orden del día

1. Familiarización con la API de Hypothesis
2. Familiarización con JavaScript
3. Familiarización con el desarrollo de extensiones
4. Dudas
5. GAUR
6. Próxima reunión

### Descripción de tareas realizadas

El alumno se ha familiarizado con:

- el uso de Postman y las APIs. En concreto la API de Hypothesis.
- Extensiones web
- Javascript
- JSON

### Temas tratados

El alumno ha comentado los conocimientos adquiridos, las tecnologías con las que se ha familiarizado y como lo ha hecho. A continuación, el alumno ha realizado varias preguntas para resolver varias dudas que tenía respecto al proyecto y las tecnologías, y se han resuelto. Además, el tutor le ha explicado cómo funcionan los grupos de Hypothesis para poder mandarle la tarea a realizar para la próxima reunión.

### Tareas futuras

Realizar un pequeño plugin para la página de grupos de Hypothesis que muestre la distribución cronológica de las anotaciones. Objetivo: familiarizar al alumno con Chrome + API Hypothesis + D3Js.

## B.2. Acta 2

Fecha: 22/02/2018      Hora: 13:00      Duración: 45'

### Asistentes:

- Oscar
- Haritz
- Eduardo

### Orden del día

1. Extensión web desarrollada
2. GAUR
3. Próxima reunión

### Descripción de tareas realizadas

- Realización de la extensión de prueba para la página de grupos de Hypothesis que muestra la distribución cronológica de las anotaciones. Se ha utilizado la librería D3.js para la realización del gráfico de barras. Enlace a GitHub: <https://github.com/eduardopercap/first-web-extension>

### Temas tratados

El alumno ha mostrado la extensión web donde ha recibido el visto bueno por parte del tutor. A continuación, el tutor ha explicado las distintas posibilidades de visualizar los datos que ofrece Hypothesis y los distintos gráficos que se pueden utilizar para ello. Por último, se ha hablado sobre el registro del proyecto en GAUR.

### Tareas futuras

- Quedar con Leticia para que explique al alumno la utilidad, el funcionamiento y la implementación del gráfico alluvial con la librería D3.js.

### B.3. Acta 3

Fecha: 16/03/2018    Hora: 13:30    Duración: 40'

#### Asistentes:

- Oscar
- Haritz
- Eduardo

#### Orden del día

1. Presentar propuesta
2. Próxima reunión

#### Descripción de tareas realizadas

- Reunión con Leticia para ver el funcionamiento y la implementación de los diagramas Alluvial en D3.js.
- Pensar que gráficos pueden ser útiles para la visualización de datos.

#### Temas tratados

El alumno ha explicado cómo fue la reunión con Leticia. Además, le ha propuesto al tutor diferentes gráficos que se podrían utilizar en la extensión web, entre las que se encuentra el diagrama Alluvial. A continuación, el tutor le ha explicado el dominio del trabajo, en este caso, exámenes de una asignatura. La idea es mostrar estadísticas e información relevantes de un examen (corregido mediante Hypothesis) que puedan ser útiles y de ayuda a un profesor. El tutor ha explicado cómo se corrigen los exámenes con la extensión Mark&Go. Por último, se han mirado los distintos tipos de gráficos existentes y que puedan ser útiles para el proyecto.

#### Tareas futuras

- Realizar una propuesta con distintos gráficos que puedan ser útiles en el dominio de los exámenes.

## B.4. Acta 4

Fecha: 28/03/2018    Hora: 10:45    Duración: 25'

### Asistentes:

- Oscar
- Haritz
- Eduardo

### Orden del día

1. Dudas
2. Propuestas
3. Próxima reunión

### Descripción de tareas realizadas

- Realización de una propuesta

### Temas tratados

El alumno ha mostrado distintas propuestas. Después de considerar y valorar cada uno de los distintos gráficos propuestos, finalmente se ha seleccionado el diagrama Alluvial que relaciona cada pregunta con las distintas puntuaciones obtenidas. Además, se ha propuesto uno nuevo, un diagrama de dispersión que muestra para cada examen, las distintas puntuaciones y cuántos alumnos.

### Tareas futuras

- Implementar los dos gráficos (el diagrama Alluvial y el gráfico de dispersión) en D3.js.

## B.5. Acta 5

Fecha: 20/04/2018    Hora: 13:15    Duración: 30'

### Asistentes:

- Oscar
- Haritz
- Eduardo

### Orden del día

1. Mostrar gráficos
2. Dudas
3. Próxima reunión

### Descripción de tareas realizadas

- Realización de una extensión web que muestra los dos distintos gráficos acordados en la anterior reunión. Enlace a GitHub:  
<https://github.com/eperez108/extensionHypothesis>

### Temas tratados

El alumno ha mostrado la extensión realizada. Se ha decidido realizar un cambio en el diagrama alluvial para que vaya mostrando la relación entre las notas de los ejercicios y que los enlaces tengan un color distinto en caso de ser aprobados o suspensos. La idea es similar a este otro ejemplo realizado en R:

<https://cran.r-project.org/web/packages/alluvial/vignettes/alluvial.html>

### Tareas futuras

- Realizar los cambios en el gráfico Alluvial.

## B.6. Acta 6

Fecha: 04/05/2018

Hora: 15:30

Duración: 90'

### Asistentes:

- Oscar
- Haritz
- Eduardo

### Orden del día

1. Mostrar gráfico
2. Dudas
3. Próxima reunión

### Descripción de tareas realizadas

- Modificación del diagrama alluvial añadiendo las relaciones entre preguntas del examen y colores para diferenciar los que han aprobado de los que no.

### Temas tratados

El alumno ha mostrado los cambios realizados en el diagrama aluvial. A continuación, ha explicado paso a paso las estructuras intermedias necesarias desde la obtención de los datos hasta la estructura de datos necesaria para dibujar el diagrama. Se ha decidido añadirle al diagrama nuevas funcionalidades como poder eliminar una pregunta e intercambiar la posición de dos preguntas. Por último, debido a la imposibilidad de realizar el segundo gráfico como estaba previsto, se ha decidido realizar un diagrama de dispersión que muestre las notas de los alumnos por pregunta de un examen en concreto.

### Tareas futuras

- Documentar para la memoria el desarrollo que se ha realizado en el diagrama alluvial.
- Actualizar el gráfico de dispersión.
- Añadir al gráfico alluvial las funcionalidades de eliminar e intercambiar columnas.

## B.7. Acta 7

Fecha: 21/05/2018

Hora: 15:15

Duración: 105'

### Asistentes:

- Oscar
- Haritz
- Eduardo

### Orden del día

1. Mostrar diagramas
2. Dudas
3. Próxima reunión

### Descripción de tareas realizadas

- Documentar para la memoria el desarrollo del diagrama alluvial. Paso por paso desde la entrada a la salida.
- Actualizar el gráfico de dispersión.
- Permitir eliminar y actualizar columnas del diagrama alluvial.

### Temas tratados

El alumno ha mostrado los cambios realizados en el diagrama aluvial. Se ha decidido añadirle al diagrama nuevas funcionalidades como poder comparar el resultado del examen actual con los otros exámenes de la asignatura. A continuación, se han planteado distintas propuestas para realizar un gráfico más y finalmente se ha optado por realizar un gráfico de araña en el que se pueda comparar las notas del alumno con las notas medias de la clase. Este gráfico debe aparecer al acceder el usuario a la carpeta del alumno en Google Drive.

### Tareas futuras

- Añadir columnas de otros exámenes (aprobados o suspensos) en el diagrama alluvial.
- Gráfico de araña comparando las notas de un alumno con la media.

## B.8. Acta 8

Fecha: 01/06/2018    Hora: 13:30    Duración: 30'

### Asistentes:

- Oscar
- Haritz
- Eduardo

### Orden de día

1. Mostrar diagramas
2. Dudas
3. Próxima reunión

### Descripción de tareas realizadas

- Se han añadido las columnas de los exámenes en el diagrama aluvial
- Se ha creado el gráfico de araña

### Temas tratados

El alumno ha mostrado los cambios realizados en el diagrama aluvial y el nuevo gráfico de araña. Ha explicado las dificultades que ha tenido con Google Drive lo que le ha imposibilitado el correcto funcionamiento del gráfico a la hora de incluirlo en Drive. Por último, se ha establecido una estructura para la memoria.

### Tareas futuras

- Conseguir el correcto funcionamiento del gráfico de araña en Google Drive.
- Mejorar algunos detalles del diagrama aluvial.
- Realizar la memoria del proyecto.

## B.9. Acta 9

Fecha: 17/06/2018    Hora: 17:00    Duración: 60'

### Asistentes:

- Oscar
- Haritz
- Eduardo

### Orden de día

4. Mostrar borrador de la memoria
5. Dudas de la memoria

### Descripción de tareas realizadas

- Se ha empezado con la memoria del proyecto

### Temas tratados

El alumno ha mostrado un primer borrador de la memoria y se han decidido que cambios hay que realizar, que falta por añadir y responder dudas. Por último se ha decidido posponer el proyecto a la convocatoria de septiembre ya que no era viable para la convocatoria de julio.

### Tareas futuras

- Continuar completando la memoria

## B.10. Acta 10

Fecha: 19/07/2018    Hora: 15:30    Duración: 30'

### Asistentes:

- Oscar
- Eduardo

### Orden de día

6. Revisar memoria

### Descripción de tareas realizadas

- Se ha avanzado en la redacción de la memoria y se han aplicado los cambios propuestos de la anterior reunión.

### Temas tratados

Se ha revisado el estado actual de la memoria y se han propuesto cambios y mejoras en varios apartados.

### Tareas futuras

- Realizar cambios y terminar la memoria del proyecto.



# Anexo C.

## Manual de usuario

---

El siguiente manual tiene como objetivo facilitar la experiencia de cualquier usuario de la extensión. En él se describen cada una de las funcionalidades disponibles y se explica cómo hacer uso de ellas.

# Índice

---

1. Instalar Mark&Go .....	1301
2. Como usar Mark&Go .....	131
2.1. Requerimientos mínimos .....	131
2.2. Definir rúbricas .....	131
2.3. Corregir exámenes .....	132
2.3.1. Encontrar evidencias .....	132
2.3.2. Corregir examen .....	133
3. Instalar Mark&GoViz .....	134
4. Como usar Mark&GoViz .....	134
4.1. Visualizar diagrama de dispersión de evaluación continua .....	134
4.2. Visualizar diagrama alluvial .....	135
4.2.1. Intercambiar preguntas .....	136
4.2.2. Eliminar pregunta .....	137
4.2.3. Intercambiar exámenes .....	137
4.2.4. Eliminar examen .....	137
4.3. Visualizar diagrama de dispersión de un examen .....	137
4.4. Visualizar gráfico de araña .....	138
4.4.1. Seleccionar examen .....	138

## 1. Instalar Mark&Go

Para instalar la extensión, hay que dirigirse al Chrome Web Store desde este enlace. Una vez allí, hay que pulsar el botón “Añadir a Chrome” y comenzará la instalación de la extensión.



*Ilustración 67. Botón de instalación*

## 2. Como usar Mark&Go

### 2.1. Requerimientos mínimos

La extensión solo funciona para el navegador Google Chrome. Para empezar, es necesario tener una cuenta de Google SpreadSheets y de Hypothes.is. La primera vez que se usa Mark&Go, este pedirá los permisos para tener acceso a estos dos servicios.

### 2.2. Definir rúbricas

La hoja de cálculo (Google SpreadSheet) es donde se definen las rúbricas y las notas se visualizan en formato tabular. En esta [plantilla](#) se puede ver como se define una hoja de cálculo para poder empezar a corregir con Mark&Go. La hoja de cálculo debe cumplir con el siguiente formato:

- La primera columna representa al alumno
- El resto de columnas son las competencias a evaluar (Solo hasta la columna “Total”)
- Para cada estudiante se debe proporcionar un enlace de la carpeta donde reside el examen. Por ejemplo,  
<https://drive.google.com/drive/folders/17f034xn3otZw96JcdTKiE8gINL3j25d>

Para establecer los valores o criterios posibles, haciendo clic en el botón derecho en la segunda fila, se puede establecer cuál es el rango de valores para cada competencia.

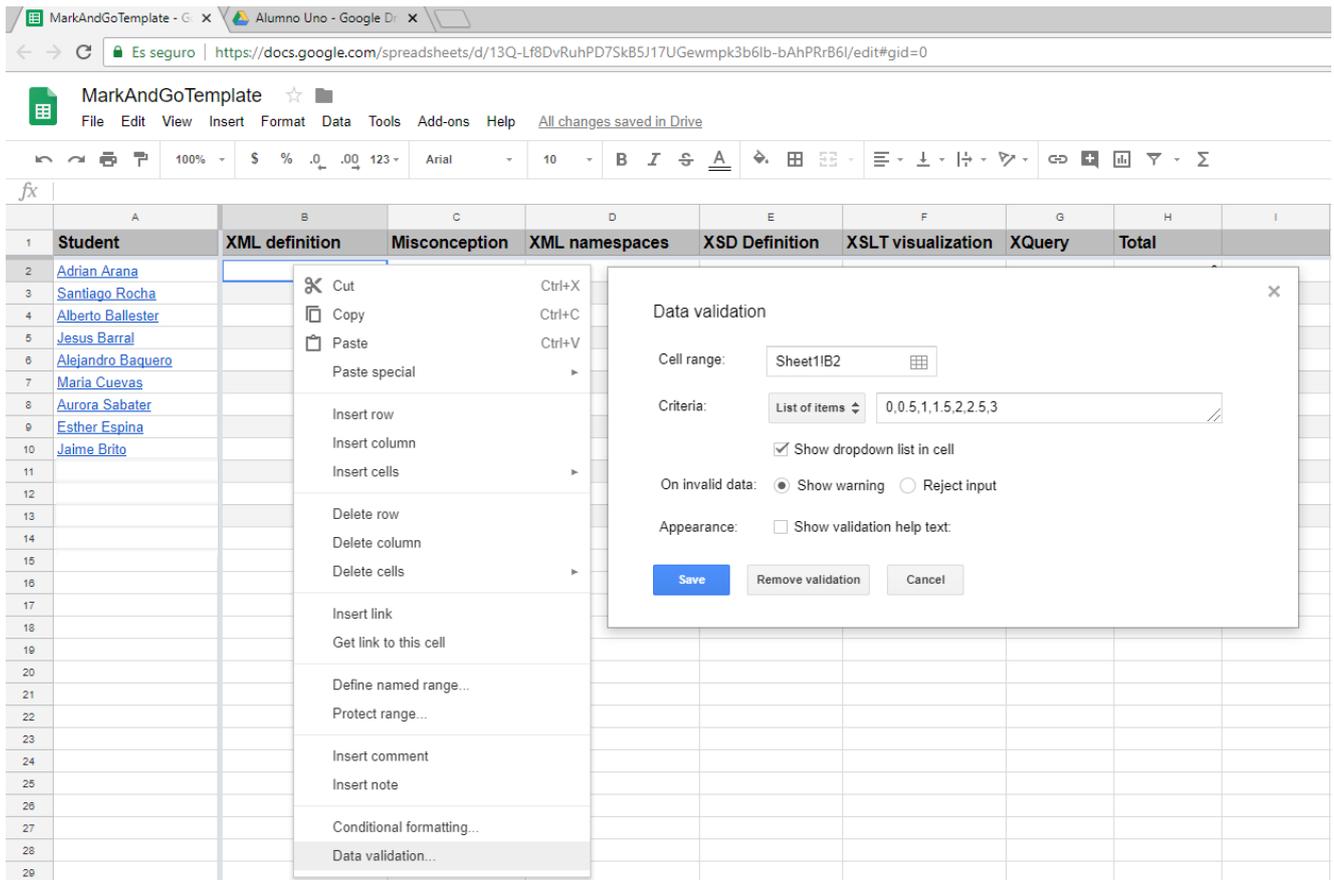


Ilustración 68. Establecer rango de valores

Después de finalizar la preparación de la hoja de cálculo, hay que hacer clic en el icono de la extensión en la parte superior derecha del navegador y la herramienta se configurará automáticamente para comenzar a corregir.

### 2.3. Corregir exámenes

Primero hay que abrir el examen del alumno en la web y después clicar en el icono de la extensión. El modo de evidencia (evidencing mode) se activará en la barra lateral. Haciendo clic en el interruptor se puede cambiar al modo de corrección (marking mode).

#### 2.3.1. Encontrar evidencias

Comenzando en modo de evidencia, hay que subrayar las evidencias para establecer más adelante la nota de los exámenes de los alumnos. Para anotar, hay que subrayar el texto que se usará como evidencia y la barra lateral se abrirá automáticamente para elegir cuál es la competencia que se está evaluando.

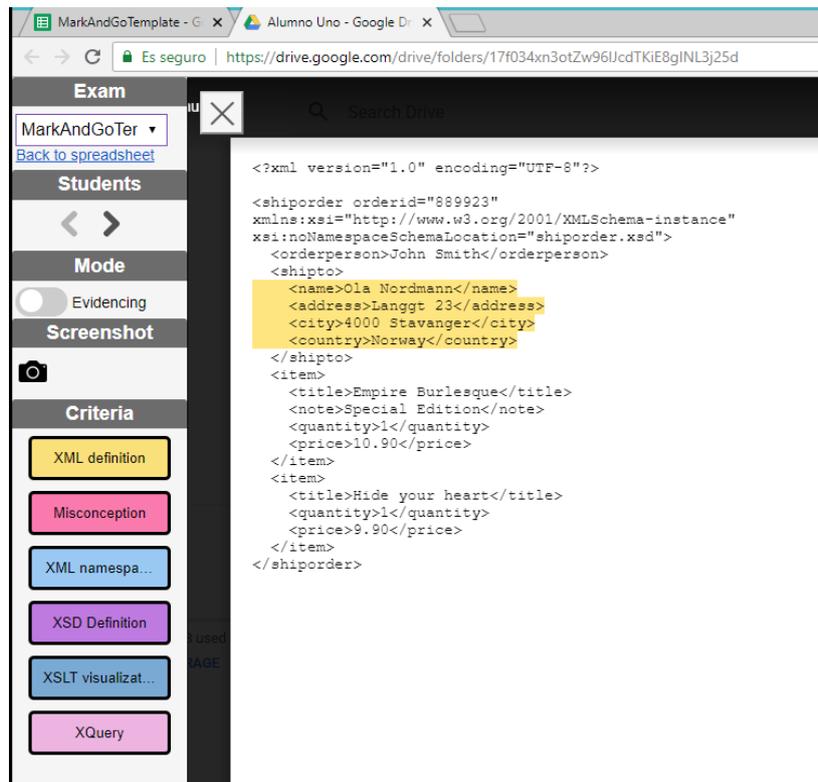


Ilustración 69. Identificar evidencias

### 2.3.2. Corregir examen

Para corregir hay que pasar al modo de corrección. Desde aquí se puede:

- Usar la barra lateral como índice para revisar las diferentes evidencias de la competencia.
- Calificar el examen seleccionando los botones de la barra lateral.
- La nota es automáticamente trasladada a la hoja de cálculo al clicar en ella, evitando así posibles errores o copia y pegas.

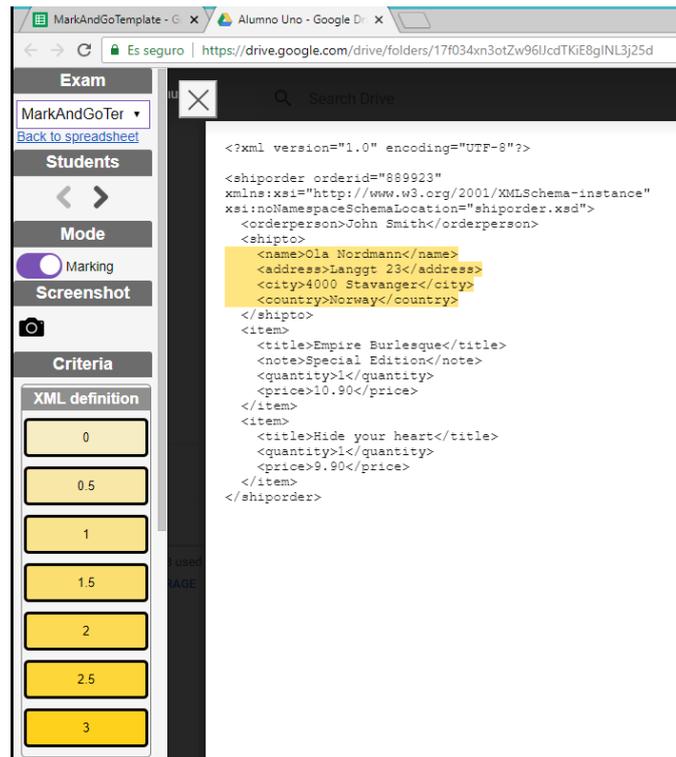


Ilustración 70. Corrección de un examen

### 3. Instalar Mark&GoViz

El primer paso es instalar la extensión en el navegador web. Para ello hay que dirigirse a la tienda de extensiones Chrome Web Store desde este enlace:

<https://chrome.google.com/webstore/detail/markandgoviz/lgpfmejklbhokgeohjjbhdinapcoljbo>

Una vez dentro, se selecciona el botón “Añadir a Chrome” para instalar la extensión.

### 4. Como usar Mark&GoViz

#### 4.1. Visualizar diagrama de dispersión de evaluación continua

Antes de nada, hay que iniciar sesión en la web de Hypothes.is y automáticamente esta lleva a la página de perfil del usuario. Tras una mínima espera de pocos segundos, en la parte superior de la página aparece el diagrama de dispersión con los datos de la evaluación continua.

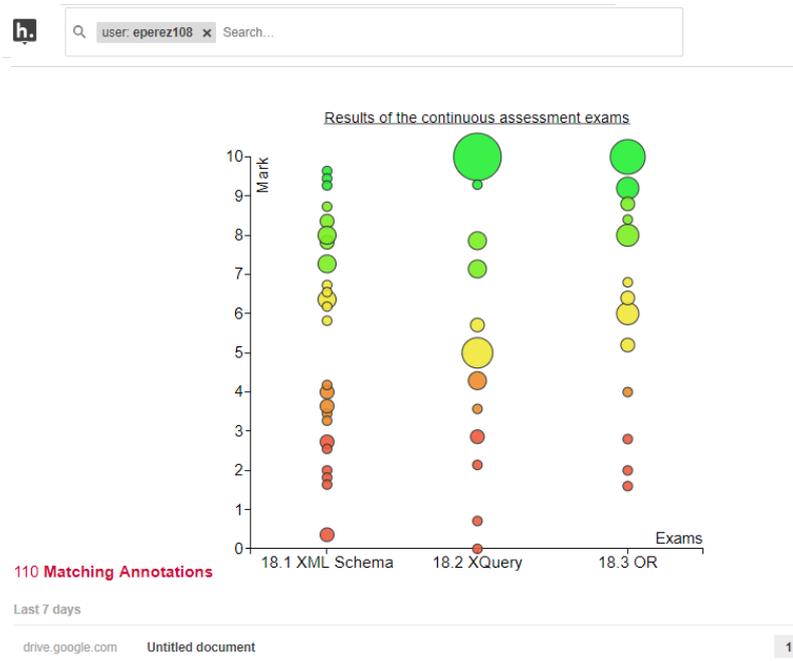


Ilustración 71. Diagrama de dispersión en la página de perfil

## 4.2. Visualizar diagrama alluvial

Desde la página de perfil, hay que dirigirse a la parte superior derecha donde sale la opción “Groups” y seleccionar un grupo que contenga las anotaciones de un examen.



Una vez dentro del grupo, aparecerán dos botones y hay que seleccionar el que se encuentra a la izquierda con el texto “Alluvial diagram”.

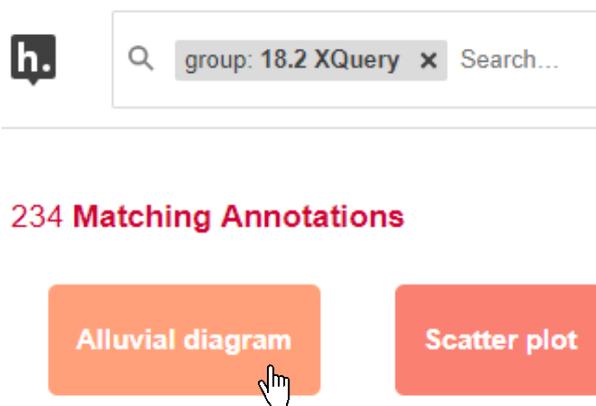


Ilustración 72. Botón para visualizar gráfico alluvial

Al momento aparecerá el diagrama alluvial debajo de los botones. En un primer momento se dibuja solo con las preguntas del examen sin las relaciones con los otros exámenes de la asignatura para el que habrá que esperar unos segundos más para que se actualice el diagrama, pero esta vez completo. Esto se debe a que la obtención de los datos del resto de exámenes requiere un mayor tiempo.

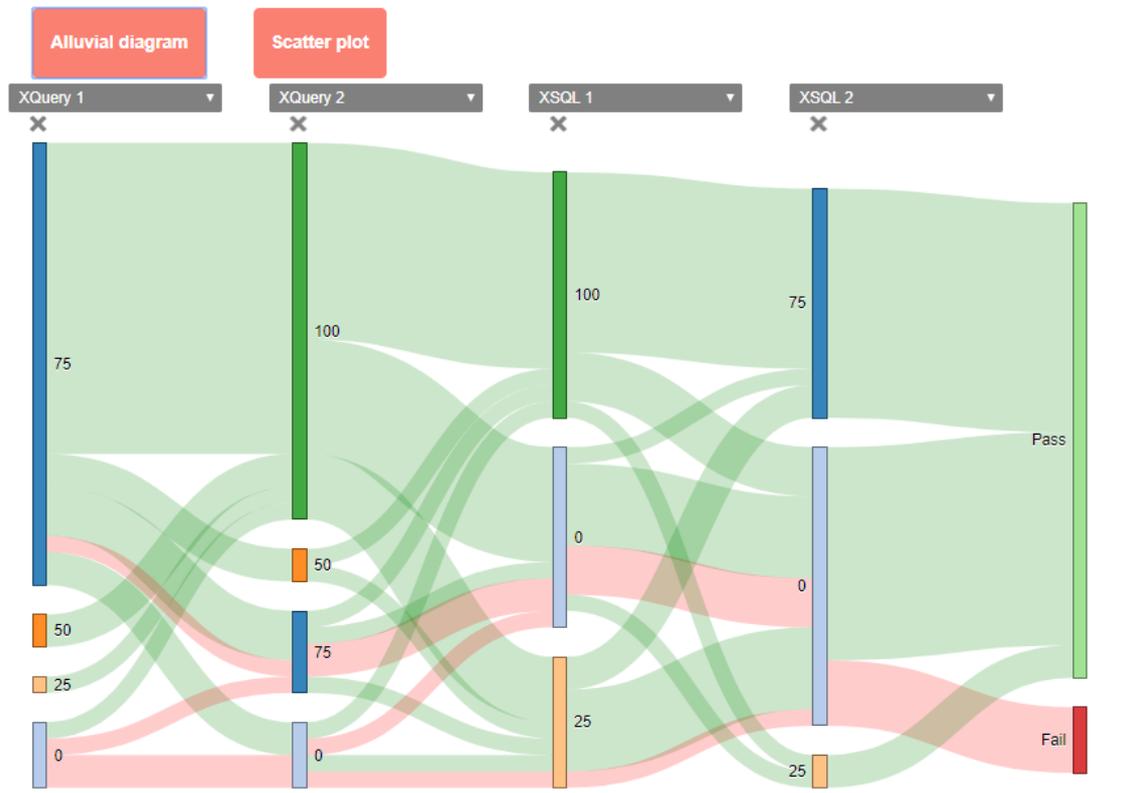


Ilustración 73. Diagrama alluvial

#### 4.2.1. Intercambiar preguntas

Para poder intercambiar las posiciones de las preguntas en el diagrama, hay que clicar en el nombre de la pregunta que se desea intercambiar y seleccionar en la lista desplegable con que pregunta realizar el intercambio.

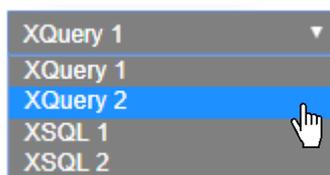


Ilustración 74. Intercambiar preguntas

#### 4.2.2. Eliminar pregunta

Para eliminar una pregunta, hay que clicar sobre el icono ✕ situado debajo del nombre de la pregunta. Existe un límite a la hora de eliminar preguntas, no se podrán eliminar todas ya que siempre tiene que haber al menos una.

#### 4.2.3. Intercambiar exámenes

El funcionamiento es igual al de intercambiar las posiciones de las preguntas. Hay que clicar en el nombre del examen y seleccionar en la lista desplegable con que otro examen se desea realizar el intercambio.

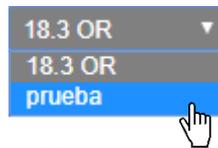


Ilustración 75. Intercambiar examen

#### 4.2.4. Eliminar examen

El funcionamiento es igual al de eliminar una pregunta. Hay que clicar sobre el icono ✕ situado debajo del nombre del examen. En este caso, no existe límite a la hora de eliminar los exámenes.

### 4.3. Visualizar diagrama de dispersión de un examen

Se visualiza de forma similar al diagrama alluvial. Al acceder a la página de un grupo con anotaciones de exámenes, hay que clicar en el botón “Scatter plot”.

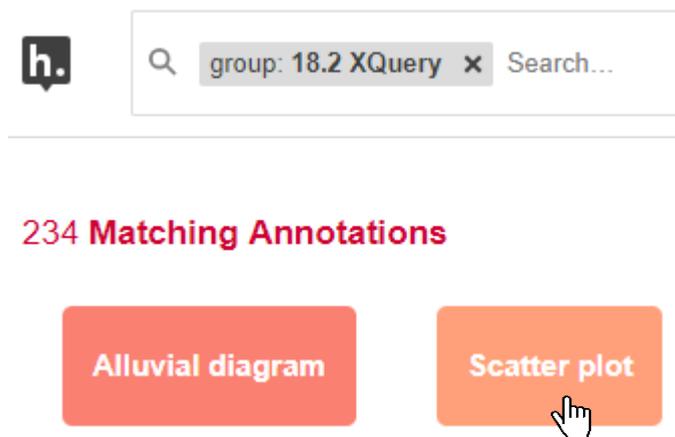


Ilustración 76. Botón para visualizar diagrama de dispersión

#### 4.4. Visualizar gráfico de araña

Lo primero es iniciar sesión en Google Drive. A continuación, se accede a la carpeta del alumno del que se desea visualizar la gráfica. Si se posiciona por encima de cada punto de corte del polígono con los ejes de las preguntas, aparece una ventana emergente con datos del alumno en dicha pregunta.

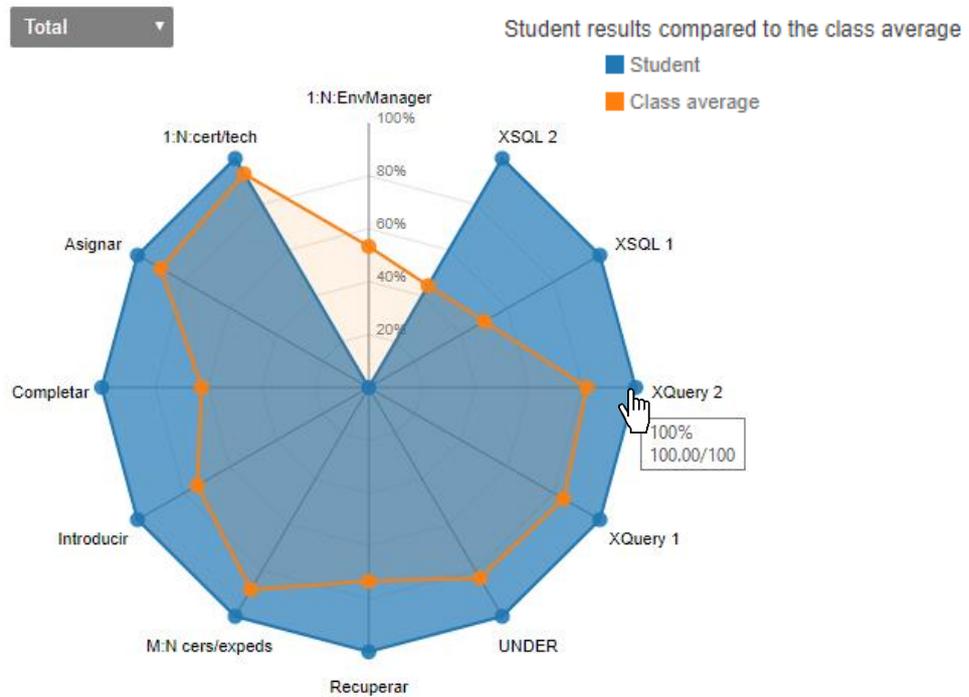


Ilustración 77. Visualizar gráfico de araña

##### 4.4.1. Seleccionar examen

Para la visualización de un examen en concreto del alumno, hay que seleccionar de la lista desplegable situado a la izquierda superior del gráfico el examen que se desea. Para regresar a la vista inicial únicamente hay que seleccionar la opción "Total".

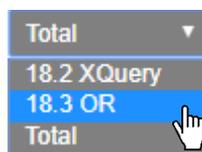


Ilustración 78. Seleccionar examen