

# Cambios en el uso del servicio de aulas virtuales de la Universidad de Huelva, antes y durante el confinamiento decretado por el COVID-19

por

Javier Cortegano Santiago

Trabajo de Fin de Máster presentado en conformidad con los requisitos del Máster en Economía, Finanzas y Computación

Universidad de Huelva & Universidad Internacional de Andalucía

**uhu**.es

**un**  
i Universidad  
Internacional  
de Andalucía  
**A**

Septiembre 2021

# Cambios en el uso del servicio de aulas virtuales de la Universidad de Huelva, antes y durante el confinamiento decretado por el COVID-19

Javier Cortegano Santiago

Máster en Economía, Finanzas y Computación

Iñaki Fernández de Viana y González  
Universidad de Huelva y Universidad Internacional de Andalucía

2021

## Abstract

Virtual learnings platforms such as Moodle have become in last months a key piece of educational institutions around the world, especially for those dedicated to face-to-face learning, because of the current situation caused by COVID-19. It is not only important to observe what use teachers and students make of these platforms, but also to analyze how the trend of use changes based on the circumstances that occur and over time, to improve and adapt the teaching, the platform and the servers to new trends and uses that are made of it. An educational institution adapted to the circumstances and with vision of the future will always be more appreciated and valued by its environment.

**Key words:** Moodle, COVID-19, Python, SQL, Stata, database, data pre-processing, clustering.

## Resumen

Las plataformas de enseñanza virtual como Moodle se han convertido en los últimos meses en una pieza clave de las instituciones educativas de todo el mundo, en especial para aquellas dedicadas a la docencia presencial, como consecuencia de la situación actual provocada por el COVID-19. No solo es importante observar qué uso realizan docentes y alumnos de dichas plataformas, sino también analizar cómo varía la tendencia de uso en base a las circunstancias que acontecen y a lo largo del tiempo, con objeto de mejorar y adaptar la docencia, la plataforma y los servidores a las nuevas tendencias y usos que se realizan de esta. Una institución educativa adaptada a las circunstancias y con visión de futuro siempre será más apreciada y valorada por su entorno.

# Tabla de Contenidos

1	Introducción .....	1
1.1	Moodle .....	1
1.2	El contexto .....	2
1.3	Análisis previos de Moodle .....	3
2	Base de datos .....	5
3	Extracción y preprocesamiento de datos .....	9
4	Resultados .....	18
4.1	Tiempo conectado y activo .....	19
4.2	Otras acciones .....	34
4.3	Foros .....	42
4.4	Recursos creados .....	48
4.5	Participación .....	57
4.6	Análisis cluster .....	62
4.6.1	Otras acciones .....	62
4.6.2	Foros .....	67
4.6.3	Recursos creados .....	70
4.6.4	Participación .....	77
5	Conclusiones .....	81
	Referencias .....	82
	Anexos .....	83

## Lista de Tablas

Tabla I. Extracto de la tabla “mdl_logstore_standard_log” del centro 3 para el año 2019 (1/2)....	7
Tabla II. Extracto de la tabla “mdl_logstore_standard_log” del centro 3 para el año 2019 (2/2) ..	8
Tabla III. Tiempo conectado.....	11
Tabla IV. Tiempo activo .....	12
Tabla V. Otras acciones .....	12
Tabla VI. Foros .....	13
Tabla VII. Recursos creados.....	14
Tabla VIII. Participación .....	15
Tabla IX. Número de usuarios por centro y año.....	18
Tabla X. Tiempo medio, máximo y mínimo conectado y desviación típica por año .....	19
Tabla XI. Tiempo medio, máximo y mínimo activo y desviación típica por año .....	20
Tabla XII. Tiempo medio, máximo y mínimo conectado y desviación típica, por centro y año..	27
Tabla XIII. Tiempo medio, máximo y mínimo activo y desviación típica, por centro y año.....	29
Tabla XIV. Tiempo medio, máximo y mínimo conectado y desviación típica, por mes y año....	30
Tabla XV. Tiempo medio, máximo y mínimo activo y desviación típica, por mes y año .....	31
Tabla XVI. Tiempo medio, máximo y mínimo conectado y desviación típica, por día y año .....	32
Tabla XVII. Tiempo medio, máximo y mínimo activo y desviación típica, por día y año .....	33
Tabla XVIII. Resumen del fichero de otras acciones por año .....	34
Tabla XIX. Foros, discusiones y posts creados y netos por año.....	42

Tabla XX. Recursos y actividades totales creados en 2019 y 2020.....	49
Tabla XXI. Recursos creados por centro y año .....	52
Tabla XXII. Actividades creadas por centro y año.....	54
Tabla XXIII. Resumen del fichero de participación.....	58
Tabla XXIV. Tabla de correlaciones de otras acciones (1/2) .....	62
Tabla XXV. Tabla de correlaciones de otras acciones (2/2) .....	63
Tabla XXVI. Análisis estadístico para la selección del número de clusters (otras acciones) .....	63
Tabla XXVII. Clusters para otras acciones.....	64
Tabla XXVIII. Medias de las variables para cada grupo (otras acciones).....	65
Tabla XXIX. Comparación de K-means y Ward (otras acciones).....	65
Tabla XXX. Clusters alternativos para otras acciones.....	66
Tabla XXXI. Medias alternativas de las variables para cada grupo (otras acciones).....	66
Tabla XXXII. Tabla de correlaciones de foros, discusiones y posts creados netos.....	67
Tabla XXXIII. Análisis estadístico para la selección del número de clusters (foros) .....	67
Tabla XXXIV. Clusters alternativos para foros.....	68
Tabla XXXV. Medias de las variables para cada grupo (foros) .....	68
Tabla XXXVI. Comparación de K-means y Ward (foros).....	69
Tabla XXXVII. Tabla de correlaciones de los recursos (1/2) .....	70
Tabla XXXVIII. Tabla de correlaciones de los recursos (2/2) .....	71
Tabla XXXIX. Análisis estadístico para la selección del número de clusters (recursos).....	71

Tabla XL Clusters para recursos creados.....	72
Tabla XLI. Medias de las variables para cada grupo (recursos).....	72
Tabla XLII. Comparación de K-means y Ward (recursos creados).....	73
Tabla XLIII. Tabla de correlaciones de las actividades (1/2).....	74
Tabla XLIV. Tabla de correlaciones de las actividades (2/2).....	74
Tabla XLV. Análisis estadístico para la selección del número de clusters (actividades).....	75
Tabla XLVI. Clusters para actividades creadas.....	75
Tabla XLVII. Medias de las variables para cada grupo (actividades).....	76
Tabla XLVIII. Comparación de K-means y Ward (actividades creadas).....	77
Tabla XLIX. Tabla de correlaciones de las participaciones (1/2).....	77
Tabla L. Tabla de correlaciones de las participaciones (2/2).....	78
Tabla LI. Análisis estadístico para la selección del número de clusters (participación).....	78
Tabla LII. Clusters para participación.....	79
Tabla LIII. Medias de las variables para cada grupo (participación).....	79
Tabla LIV. Comparación de K-means y Ward (participación).....	80

## Lista de Figuras

Figura I. Tiempo conectado en 2019 (0-180 minutos) .....	20
Figura II. Tiempo conectado en 2019 (120-800 minutos) .....	21
Figura III. Tiempo conectado en 2020 (0-180 minutos).....	22
Figura IV. Tiempo conectado en 2020 (120-800 minutos).....	22
Figura V. Tiempo activo en 2019 (0-60 minutos) .....	23
Figura VI. Tiempo activo en 2019 (60-250 minutos) .....	24
Figura VII. Tiempo activo en 2020 (0-60 minutos).....	25
Figura VIII. Tiempo activo en 2020 (60-300 minutos) .....	25
Figura IX. Número de inicios de sesión por centro y año .....	35
Figura X. Total de acciones por centro y año .....	36
Figura XI. Número de inicios de sesión y total de acciones por mes y año .....	37
Figura XII. Visualizaciones de calificaciones y mensajes enviados por mes y año .....	38
Figura XIII. Número de inicios de sesión por día y año .....	38
Figura XIV. Total de acciones por día y año .....	39
Figura XV. Números de inicios de sesión por hora y año .....	40
Figura XVI. Total de acciones por hora y año .....	41
Figura XVII. Foros creados en 2019 y 2020 por centro .....	43
Figura XVIII. Discusiones creadas en 2019 y 2020 por centro .....	44
Figura XIX. Discusiones creadas en 2019 y 2020 por día de la semana .....	45

Figura XX. Posts creados en 2019 y 2020 por centro.....	46
Figura XXI. Posts creados en 2019 y 2020 por día de la semana.....	47
Figura XXII. Principales recursos creados en 2019 y 2020 por meses .....	51
Figura XXIII. Actividades creadas en 2019 y 2020 por meses .....	53
Figura XXIV. Total de recursos creados por semana en 2019 y 2020 .....	55
Figura XXV. Total de recursos creados por semana en 2019 y 2020.....	56
Figura XXVI. Número de tareas subidas por centro y año.....	59
Figura XXVII. Número de tareas subidas por semana y año .....	59
Figura XXVIII. Número de cuestionarios respondidos por centro y año .....	60
Figura XXIX. Número de cuestionarios respondidos por semana y año.....	61
Figura XXX. Dendrograma de otras acciones .....	64
Figura XXXI. Dendrograma de foros .....	69
Figura XXXII. Dendrograma de recursos creados .....	73
Figura XXXIII. Dendrograma de actividades creadas.....	76
Figura XXXIV. Dendrograma de participación.....	80

## Lista de Anexos

Anexo I.....	83
Anexo II.....	94
Anexo III.....	95
Anexo IV.....	100
Anexo V.....	101
Anexo VI.....	103
Anexo VII.....	104
Anexo VIII.....	106
Anexo IX.....	109
Anexo X.....	112
Anexo XI.....	116
Anexo XII.....	119
Anexo XIII.....	122
Anexo XIV.....	124

# 1 Introducción

El propósito de este estudio es analizar cómo afectó el confinamiento decretado en España a causa del COVID-19 al uso que realizaban los docentes y alumnos de la Universidad de Huelva respecto al servicio de aulas virtuales, observando cómo influyó tanto a la conducta de los usuarios dentro de la plataforma como al uso de las herramientas disponibles. Se empleará, como muestra, las tres asignaturas del segundo cuatrimestre que hayan tenido mayor actividad en cada facultad, con sus respectivos profesores y alumnos, de cada una de las nueve facultades de la Universidad de Huelva, para los años 2019 y 2020, para así poder realizar una comparativa del cambio.

## 1.1 Moodle

La plataforma de enseñanza virtual Moodle es un software libre creado como sistema de gestión del aprendizaje (LMS), con licencia GNU GPL y programado en PHP, que se emplea en multitud de países como apoyo a la docencia presencial, principalmente, aunque también como herramienta de enseñanza puramente online.

Tras años de desarrollo por parte de su fundador, Martin Dougiamas, vio lanzada su primera versión en 2002. Desde entonces ha ido creciendo en sitios y usuarios sin cesar, pasando de apenas 1.000 sitios web en 2004 a más de 160 millones de usuarios y 106.000 sitios en junio de 2019, según las cifras ofrecidas en el sitio web oficial, y a más de 190 millones de usuarios y 145.000 sitios en marzo de 2020, coincidiendo con la expansión incontrolada de la pandemia originada por el COVID-19. (Moodle, 2021)

La plataforma Moodle ofrece a los docentes más de una veintena de recursos diferentes, de fácil configuración, con los que apoyar su docencia, ya sea presencial o virtual, y evaluar a sus alumnos, dando la posibilidad de publicar desde archivos hasta salas de chat y foros, e incluso de instalar plugins en el servidor de la plataforma con los que extender el número de recursos disponibles gracias a su diseño modular, así como monitorizar el uso que hacen sus alumnos del espacio dedicado a cada curso o asignatura.

Precisamente, durante el confinamiento que tuvo lugar en España desde marzo de 2020 a causa de la pandemia generada por el COVID-19, en la Universidad de Huelva se instalaron nuevos plugins en la plataforma Moodle para facilitar la adaptación a la docencia online, estos plugins ofrecían la posibilidad de subir y visionar videos a la plataforma y crear sesiones de Zoom para impartir las clases online.

Además de esto, la plataforma almacena la información sobre todo lo que ocurre en el sitio web en una base de datos SQL, permitiendo a los administradores monitorizar y analizar el uso que se realiza de la propia plataforma. “Los datos sobre la actividad de los estudiantes se almacenan como información de bajo nivel, es decir, cada click o interacción con la plataforma está asociado con un registro único en la base de datos, lo que la hace adecuada para el análisis de comportamientos de aprendizaje” (Hernández-García, Acquila-Natale, Iglesias-Pradas, & Chaparro-Peláez, 2018)

## 1.2 El contexto

La plataforma de enseñanza virtual Moodle lleva muchos años operativa en la Universidad de Huelva ofreciendo un servicio de aula virtual, sin embargo, al ser una institución dedicada a la docencia presencial, la plataforma había permanecido destinada a servir como apoyo a la docencia presencial, principalmente mediante la publicación online de los temarios de cada asignatura, la subida de tareas en formato digital y la realización de algunas pruebas de evaluación continua.

La expansión incontrolada de la pandemia causada por el virus COVID-19 supuso la paralización de la sociedad y un revés para todos los sectores, que se materializó en forma de confinamiento. Esto obligó a todos los colegios, institutos y universidades a adaptarse en tiempo récord a un formato que era nuevo para casi la totalidad de la población, la docencia online.

Había que adaptar las clases presenciales y los sistemas de evaluación a esta nueva situación, lo que llevó a los profesores a exprimir al máximo las posibilidades que ofrecía la plataforma. Es este cambio repentino en el uso de la plataforma el origen de este estudio, analizar cómo cambió el uso de la plataforma Moodle, en la Universidad de Huelva, de un año para otro, incluso podría

decirse que cambió de un día para otro, para adaptarse a la nueva situación, ya que nadie previó en aquel momento el cambio que nos acontecía. Sin lugar a duda, la excepcional situación vivida ha dado lugar a nuevas e innovadoras formas de enseñar, educar y evaluar, como reconocen Portillo Peñuelas, Castellanos Pierra, Reynoso González y Gavotto Nogales (2020) en su trabajo, “el contexto de emergencia sanitaria ha provocado que gran parte del profesorado aprenda sobre la marcha, improvisando y la vez, desarrollando propuestas creativas e innovadoras, que demuestran la capacidad de adaptación y flexibilidad ante los cambios.”.

Por ejemplo, Blackboard, una empresa de tecnología educativa, que ofrece servicios como Blackboard Learn, un sistema de gestión del aprendizaje, o Blackboard Collaborate, una sala virtual para clases online, reportó que, tan solo en marzo de 2020, el número diario de usuarios de Blackboard Collaborate se incrementó en más de un 3.600%, el número de logins diarios en Blackboard Learn aumento un 400%, el número de altas en Blackboard Community aumentó en un 55%, se realizaron más de 225 millones de llamadas, 130 millones de mensajes de texto y más de 160 millones de emails, además de 860.000 nuevas descargas de sus apps (Blackboard, 2020).

### 1.3 Análisis previos de Moodle

En una primera etapa, antes de la extracción y preprocesamiento de datos, se procedió a realizar una ligera revisión de la literatura previa, como el trabajo de Ruiz-de-Azcárate, Hernández-García, Iglesias-Pradas y Acquila-Natale (2017) en el que proponen un sistema de indicadores con los que analizar el trabajo en grupo que se realiza a través de la plataforma Moodle, y la revisión posterior por parte de Hernández-García et al. (2018) en el que diseñan un proceso para la extracción y el análisis de los indicadores anteriormente propuestos. Sin embargo, ambos trabajos solo pudieron ser tenidos en cuenta en la conceptualización del presente estudio, ya que solo se trabajará con una única tabla y se estudiará el uso de la plataforma, en lugar de las comunicaciones y el trabajo en equipo como ellos propusieron.

Otro de los inconvenientes de este tipo de análisis, centrados en el trabajo en grupo y las comunicaciones, tanto internas como externas, es que son análisis, en mi opinión, desfasados, principalmente por su antigüedad, ya que vivimos en un mundo cada vez más interconectado e

intercomunicado, en el que los alumnos se comunican y organizan a través de plataformas de mensajería instantánea, como WhatsApp o Telegram, y se comunican con los profesores a través del sistema de correo interno de la Universidad de Huelva, por lo que no suelen hacer uso del sistema de mensajería interna de la plataforma Moodle, salvo casos puntuales,. Algo similar sucede con los foros que, salvo actividades puntuales propuestas por los docentes o, como ha sucedido estos últimos meses, los establezcan como método para responder dudas de los alumnos, no suelen tener gran relevancia en el día a día de la docencia.

Por otro lado, buena parte de los estudios sobre la plataforma valoran las oportunidades que ofrece su implantación o analizan los resultados académicos obtenidos por los alumnos, o la satisfacción de los mismos, con la implantación de la plataforma en una facultad o en una universidad.

Sin embargo, durante estos últimos meses si que se han llevado a cabo algunos estudios en los que se analiza cómo ha afectado la pandemia al uso de la plataforma Moodle en las universidades. Un ejemplo de ello es el estudio realizado por Lapevska, Velinov y Zdravev (2021), en el que analizaron el uso de las actividades en Moodle, antes y durante la pandemia, en la Universidad de Goce Delcev (Shtip, Macedonia del Norte) y encontraron que, tan solo en abril de 2020, el número de actividades realizadas en la plataforma se incrementaron en más de un millón respecto al mismo mes del año anterior y que, el total de actividades, en el período marzo-diciembre, se triplicaron de un año para otro a consecuencia de la implantación de la docencia online por la situación derivada de la pandemia.

Muy similar a los dos primeros estudios mencionados en este apartado, es el estudio realizado por Dascalu et al. (2021) en la Universidad Politécnica de Bucarest, Rumanía, en el que analizaron la participación y el comportamiento de los estudiantes de la asignatura de Diseño de Algoritmos dentro de la plataforma Moodle, antes y durante el COVID-19, mediante minería de textos y análisis de redes, principalmente, para tratar de predecir los resultados académicos de los alumnos. En este estudio encontraron que, especialmente al comienzo del semestre objeto de estudio, la participación de los estudiantes en las actividades y la longitud y complejidad de las comunicaciones aumentó significativamente a la vez que se redujo el número de estudiantes que no interactuaban con otros de manera online.

## 2 Base de datos

La plataforma virtual Moodle genera un registro en una base de datos SQL por cada acción que se realiza en la propia plataforma, estos registros se almacenan en un elevado número de tablas interconectadas. La base de datos de la plataforma Moodle sigue un modelo relacional, es decir, la información principal es almacenada en una tabla denominada “mdl\_logstore\_standard\_log”, mientras que el resto de las tablas almacenan información ampliada sobre cada uno de los registros que tienen lugar en la tabla principal, así como otros registros que no se almacenan en dicha tabla.

Para la realización de este estudio se ha empleado, únicamente, extractos de la tabla “mdl\_logstore\_standard\_log”. Dichos extractos, realizados por el tutor Iñaki Fernández de Viana y González, se corresponden con las tres asignaturas que hayan tenido mayor actividad (entendiendo por actividad el número de registros generados en la base de datos) respecto al resto de asignaturas de la facultad a la que pertenecen, de las que solo se conoce su identificador numérico, además de que fueran del segundo cuatrimestre y de los años 2019 y 2020, de las nueve facultades de la Universidad de Huelva (cada una con un código numérico predefinido). También se ha dispuesto de las acciones de “curso 0”, las cuales son acciones que no están vinculadas a ningún curso, como son los inicios y cierres de sesión y el envío y visionado de mensajes dentro de la plataforma, de aquellos usuarios que tuvieran acciones realizadas en los cursos objeto de estudio o bien fueran profesores de las asignaturas seleccionadas.

Se eligieron las tres asignaturas con mayor actividad con objeto de tener el mayor número de registros posibles. Se eligió el segundo cuatrimestre por ser, durante dicho cuatrimestre, cuando tuvo lugar el confinamiento por la situación generada por la pandemia del COVID-19. Y, por último, se eligieron los años 2019 y 2020 para observar cómo cambió el uso de la plataforma en las asignaturas seleccionada antes y durante el confinamiento.

Como puede verse en la tabla I y la tabla II, “mdl\_logstore\_standard\_log” cuenta con un total de 21 columnas, sin embargo, para el presente estudio, principalmente se han empleado las columnas “eventname”, “userid”, “courseid”, “other”, “timecreated” y “origin”, ya que son las columnas que contienen la información de mayor interés.

La columna “eventname” contiene el nombre con el que se registra la acción, formado por el componente o módulo de la plataforma en que se realiza la acción y la propia acción realizada, la columna “userid” contiene el número que identifica al usuario que realiza la acción, “courseid” contiene el número que identifica el curso en el que se ha realizado la acción, “other” contiene información extra sobre la acción realizada (como, por ejemplo, el correo institucional del usuario que inicia sesión, el contenido de un mensaje enviado a través de la plataforma o qué tipo de recurso se ha añadido a un curso), “timecreated” contiene la fecha en formato “Unix Time” (cantidad de milisegundos transcurridos desde la medianoche UTC del 1 de enero de 1970) y, por último, “origin” indica desde qué plataforma se realizó la acción (“web”: acceso desde un navegador; “cli”: abreviatura de “Command Line Interface”, generalmente acciones realizadas por administradores; “ws”: acceso desde la plataforma para móviles).

Codificación numérica de los centros y de las respectivas asignaturas:

- Facultad de Ciencias Experimentales: 3; 600, 3572 y 5454.
- Facultad de Humanidades: 8; 24, 1107 y 2472.
- Facultad de Trabajo Social: 12; 490, 491 y 498.
- Facultad de Ciencias del Trabajo: 14; 530, 3308 y 3367.
- Facultad de Enfermería: 16; 316, 330 y 3330.
- ETS de Ingeniería Superior: 18; 248, 249 y 255.
- Facultad de Derecho: 28; 148, 150 y 311.
- Facultad de Ciencias Empresariales: 30; 516, 522 y 548.
- Facultad de Ciencias de la Educación: 36; 190, 2420 y 4140.

**Tabla I.** Extracto de la tabla “mdl\_logstore\_standard\_log” del centro 3 para el año 2019 (1/2)

id	eventname	component	action	target	objecttable	objectid	crud	edulevel	contextid	contextlevel
100556079	\core\event\message_viewed	core	viewed	message	message_read	10797852	c	0	291518	30
100556082	\core\event\message_viewed	core	viewed	message	message_read	10797855	c	0	291518	30
100556115	\core\event\message_sent	core	sent	message			c	0	1	10
100556118	\core\event\message_viewed	core	viewed	message	message_read	10797858	c	0	250487	30
100556121	\core\event\message_sent	core	sent	message			c	0	1	10
100556124	\core\event\message_viewed	core	viewed	message	message_read	10797861	c	0	608842	30
100556127	\core\event\message_sent	core	sent	message			c	0	1	10
100556130	\core\event\message_viewed	core	viewed	message	message_read	10797864	c	0	249411	30
100556133	\core\event\message_sent	core	sent	message			c	0	1	10
100556079	\core\event\message_viewed	core	viewed	message	message_read	10797852	c	0	291518	30

**Tabla II.** Extracto de la tabla “mdl\_logstore\_standard\_log” del centro 3 para el año 2019 (2/2)

contextinstanceid	userid	courseid	relateduserid	anonymous	other	timecreated	origin	ip	relateduserid
36448	36448	0	14434	0	a:1:{s:9:"messageid";s:8:"10775271";}	1548979202	web	217.217.70.244	
36448	36448	0	14434	0	a:1:{s:9:"messageid";s:8:"10775724";}	1548979202	web	217.217.70.244	
0	55177	0	33533	0	a:1:{s:9:"messageid";i:11010558;}	1548979207	cli		
33533	33533	0	55177	0	a:1:{s:9:"messageid";i:11010558;}	1548979207	cli		
0	55177	0	55177	0	a:1:{s:9:"messageid";i:11010561;}	1548979207	cli		
55177	55177	0	55177	0	a:1:{s:9:"messageid";i:11010561;}	1548979207	cli		
0	55177	0	33203	0	a:1:{s:9:"messageid";i:11010564;}	1548979207	cli		
33203	33203	0	55177	0	a:1:{s:9:"messageid";i:11010564;}	1548979207	cli		
0	55177	0	55357	0	a:1:{s:9:"messageid";i:11010567;}	1548979207	cli		
55357	55357	0	55177	0	a:1:{s:9:"messageid";i:11010567;}	1548979207	cli		

### 3 Extracción y preprocesamiento de datos

Antes de comenzar a trabajar con los datos reales, se procedió a generar una plataforma Moodle local con la que analizar e identificar por qué se generaban cada uno de los diferentes registros en la base de datos SQL, para ello se crearon 6 usuarios ficticios (2 profesores y 4 alumnos) y 3 cursos, uno de ellos común a todos los usuarios. Una vez creados los cursos y los usuarios, se procedió a realizar el mayor número posible de acciones dentro de la plataforma, así como crear todos los recursos disponibles, interactuar con los recursos creados y realizar comunicaciones entre los usuarios ficticios, con objeto de identificar en la base de datos qué registro generaba cada acción realizada. Esto se hizo para preparar y optimizar las consultas y códigos que posteriormente serían ejecutados en las tablas de la base de datos real. La versión de Moodle empleada para la preparación es la 3.5, la misma que emplea actualmente la Universidad de Huelva, así como el programa DBeaver, en su última versión (21.1.5), empleando el sistema de gestión de bases de datos MariaDB, derivado de MySQL, para conectar con la base de datos de la plataforma local y poder visionar las tablas SQL con los datos reales, las cuales fueron aportadas en formato script por el tutor Iñaki Fernández de Viana y González, así como las listas con los profesores de las asignaturas analizadas de cada centro. Estas tablas SQL en formato script fueron cargadas a la base de datos local para la ejecución posterior de los códigos desarrollados.

Para la realización de este estudio se partió, como se ha mencionado anteriormente, de un total de 18 extractos, en forma de tablas independientes, de la base de datos de la plataforma Moodle de la Universidad de Huelva, los cuales sumaban, entre sí, más de 40 GB de información repartidos en más de 110 millones de registros, de ahí la importancia de una adecuada preparación de los códigos que posteriormente serían ejecutados.

En primer lugar, se ejecutaron los códigos SQL del anexo I sobre las tablas cargadas en la base de datos local, dando lugar a un total de 774 ficheros CSV, los cuales recogieron, para todos los centros y años, el número de inicios de sesión, los mensajes enviados, las veces que se visionaron las calificaciones dentro de la plataforma, el total de acciones realizadas, los foros, discusiones y posts creados (así como el neto, es decir, la diferencia entre creados y borrados), además de todos los recursos y actividades creados en la plataforma y los envíos, participaciones y accesos a los recursos en los casos en que procediera, como, por ejemplo, accesos y

participaciones en glosarios o envío de archivos a tareas. Todos estos registros fueron agrupados por usuario y/o curso y por fecha, hora y semana y se estableció una condición para que solo fuera extraída la información de los registros generados que estuvieran dentro del período hábil académico del segundo cuatrimestre, el cuál es establecido cada año por el Rectorado de la Universidad de Huelva, dicho período abarca del 18 de febrero al 19 de julio para el año 2019 y del 17 de febrero al 20 de julio para el año 2020.

Las librerías de Python, así como las listas con las ID de los profesores y el resto de las variables empleadas en Python para los procesos que se describirán a continuación, se encuentran en el anexo II.

En segundo lugar, dadas las limitaciones que se encontraron en SQL, se procedió a generar tres funciones en Python destinadas a obtener las acciones de cada usuario, el tiempo conectado y el tiempo activo de cada uno de los usuarios, las cuales fueron ejecutadas en bucle (Anexo III) para todas las tablas y almacenados en dos ficheros CSV por cada centro y año. Para el tiempo activo se consideró un tiempo máximo de 15 minutos entre acciones o de una hora si la acción iterada se correspondía con un cierre de sesión, ya que se consideró que un tiempo mayor, en la mayoría de los casos, supondría que el usuario no permanecía trabajando en la plataforma de forma activa, mientras que, debido a las características de los datos analizados, se consideró, para el tiempo conectado, un tiempo máximo de dos horas entre acciones, por ser la duración de las clases, y cuatro horas si la acción iterada se correspondía con un cierre de sesión. La distinción en el tiempo cuando la acción iterada se trataba de un cierre de sesión se estableció por considerar que no es habitual que los usuarios cierren sesión de forma manual (cuando el cierre de sesión se produce por inactividad no queda registrado en la tabla analizada), por lo que podría suponer que el usuario podría haber estado realizando acciones de larga duración, como, por ejemplo, el visionado de ficheros o de videos.

En tercer lugar, se generó un código de Python para obtener las ID de los usuarios que fueran alumnos (Anexo IV), el cual se emplearía más adelante para eliminar los registros innecesarios, es decir, aquellos que no se correspondieran con alumnos o con los docentes de las asignaturas analizadas, y se procedió a la adecuación de todos los ficheros generados previamente (Anexo V), dicha adecuación consistió en: mover al final la columna correspondiente al recuento de las acciones, renombrar correctamente las columnas generadas, unir las columnas de día y mes en

una única columna de fecha, incluyendo el año, y añadir una nueva columna con el centro correspondiente. Esto se hizo con objeto de poder agrupar todos los CSV generados en seis únicos archivos, reduciendo así el gran volumen de archivos, y realizar, posteriormente, un mejor análisis de los datos obtenidos. Una vez obtenidas las listas de alumnos y corregidos los ficheros, se incorporó las listas de profesores a las de los alumnos (Anexo VI).

Por último, se procedió a unificar los 846 ficheros CSV generados anteriormente en 6 archivos (Anexo VII) que recogieran el tiempo conectado (Tabla III), el tiempo activo (Tabla IV), los ficheros con otras acciones, es decir, número de inicios de sesión, total de acciones por hora, visualización de calificaciones y mensajes enviados (Tabla V), la información de los foros creados y borrados, incluyendo discusiones y posts (Tabla VI), los ficheros relativos a creación de recursos y actividades (Tabla VII) y los ficheros de participación, es decir, acceso y participación en glosarios y chats, envíos de archivos a tareas, talleres y videos, acceso a sesiones de Zoom, registros de asistencias a clases virtuales y las encuestas, consultas y cuestionarios respondidos (Tabla VIII), así como eliminar los registros, en los archivos que procediera, que no se correspondieran con acciones de alumnos y docentes de los respectivos cursos de cada centro.

**Tabla III.** Tiempo conectado

usuario	centro	tiempo_conectado	plataforma	fecha
542	c03	01:33:13	web	2019-02-01
542	c03	00:00:10	cli	2019-02-01
542	c03	00:00:28	web	2019-02-03
...	...	...	...	...
72552	c36	00:00:51	web	2020-06-21
72552	c36	00:55:55	web	2020-06-24
72552	c36	00:01:32	web	2020-06-26

**Tabla IV.** Tiempo activo

usuario	centro	tiempo_activo	plataforma	fecha
542	c03	00:00:10	cli	2019-02-01
542	c03	00:00:28	web	2019-02-03
542	c03	00:03:37	web	2019-02-18
...	...	...	...	...
72552	c36	00:00:00	web	2020-06-11
72552	c36	00:00:51	web	2020-06-21
72552	c36	00:01:32	web	2020-06-26

**Tabla V.** Otras acciones

usuario	centro	hora	fecha	semana	nis	t_acc	vistas_calif	num_mensajes_enviados
13	c14	8	2019-05-06	18	1	0	0	0
13	c14	9	2019-02-18	7	2	5	0	0
13	c14	9	2019-02-25	8	2	8	0	0
...	...	...	...	...	...	...	...	...
72738	c08	23	2020-06-12	23	1	2	0	0
72738	c08	23	2020-06-15	24	0	10	0	0
72738	c08	23	2020-06-22	25	0	5	0	0

**Tabla VI.** Foros

<b>usuario</b>	<b>curso</b>	<b>centro</b>	<b>hora</b>	<b>fecha</b>	<b>semana</b>	<b>foros_creados</b>	<b>...</b>	<b>discusiones_creadas_netos</b>	<b>posts_creados</b>	<b>posts_creados_netos</b>
148	249	c18	10	2020-05-23	20	0	...	0	0	0
148	249	c18	13	2020-05-12	19	0	...	0	0	0
148	249	c18	13	2020-05-25	21	0	...	1	0	0
148	249	c18	13	2020-06-11	23	0	...	0	0	0
148	249	c18	14	2020-03-19	11	1	...	0	0	0
...	...	...	...	...	...	...	...	...	...	...
72378	2420	c36	20	2020-04-02	13	0	...	1	0	0
72378	2420	c36	20	2020-06-15	24	0	...	1	0	0
72378	2420	c36	21	2020-02-27	8	0	...	1	0	0
72738	24	c08	12	2020-03-19	11	0	...	0	1	1
72738	24	c08	16	2020-06-21	25	0	...	1	0	0

**Tabla VII.** Recursos creados

usuario	curso	centro	hora	fecha	semana	n_archivos	...	n_szoom	n_talleres	n_tareas	n_urls	n_videos	n_wikis
13	530	c14	9	2019-06-20	24	0	...	0	0	0	1	0	0
148	249	c18	10	2020-06-15	24		...	1	0	0	0	0	0
148	249	c18	11	2020-05-27	21	0	...	0	0	0	0	0	0
148	249	c18	14	2020-05-04	18	1	...	0	0	0	0	0	0
148	249	c18	17	2020-04-17	15	0	...	2	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
72378	2420	c36	19	2020-02-24	8	0	...	0	0	0	0	0	0
72378	2420	c36	19	2020-03-30	13	1	...	0	0	0	0	0	0
72378	2420	c36	19	2020-06-15	24	1	...	0	0	0	0	0	0
72378	2420	c36	20	2020-02-24	8	0	...	0	0	0	0	0	0
72378	2420	c36	20	2020-02-27	8	1	...	0	0	0	0	0	0

**Tabla VIII.** Participación

curso	centro	hora	fecha	semana	naz	nar	nac	npc	nag	npg	ntals	ntars	nvs	ncor	ner	ncur
24	c08	0	2020-04-15	15	0	0	0	0	0	0	0	1	0	0	0	0
24	c08	0	2020-05-26	21	0	0	0	0	0	0	0	1	0	0	0	0
24	c08	0	2020-05-28	21	0	0	0	0	0	0	0	1	0	0	0	0
24	c08	0	2020-06-09	23	0	0	0	0	0	0	0	1	0	0	0	0
24	c08	0	2020-06-13	23	0	0	0	0	0	0	0	1	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5454	c03	23	2020-04-16	15	0	0	1	0	0	0	0	0	0	0	0	0
5454	c03	23	2020-04-21	16	0	0	1	0	0	0	0	0	0	1	0	0
5454	c03	23	2020-05-06	18	0	0	0	0	0	0	0	4	0	0	0	0
5454	c03	23	2020-05-17	20	0	0	0	0	0	0	0	3	0	0	0	0
5454	c03	23	2020-05-22	20	0	0	0	0	0	0	0	0	0	1	0	0

Antes de continuar, es necesario realizar una serie de advertencias:

- En primer lugar, el tiempo activo y el tiempo conectado obtenidos debe tomarse con cautela, ya que, aunque en el código para sus respectivas extracciones se ha intentado plasmar la mayor variedad de circunstancias posibles, no han sido analizadas todas las asignaturas de cada centro, por lo que un alumno ha podido estar realizando acciones en otro curso, arrojando información incompleta.
  
- Por otro lado, y relacionado con la anterior, se podría profundizar, en una posterior revisión, en las acciones con origen “cli” o “ws”, ya que las acciones con estos orígenes han provocado que se tuviera que modificar el código inicialmente previsto para el tiempo conectado por la naturaleza de sus registros. Esta revisión recomendada se debe a que:
  - Las acciones con origen “cli” son, a priori, acciones ejecutadas por administradores o por la propia plataforma, por ejemplo, cuando un profesor publica un cuestionario en la plataforma, esto origina una entrada en el calendario y, además, se envía una notificación a los alumnos matriculados. Analizando los registros de la tabla de la base de datos, puede verse como, generalmente, aparece que es el profesor quien realiza esas acciones, pero con origen “cli”, lo cual, en realidad, significa que es la propia plataforma quien ha realizado dichas acciones desencadenadas por una acción previa del profesor. Este hecho no distorsiona los registros de tiempo conectado y activo, ya que la plataforma ejecuta dichas acciones en cuestión de segundos. Sin embargo, si se observan todas las acciones con origen “cli”, puede verse que aparecen registros como “\core\event\notification\_viewed” y “\core\event\message\_viewed” (visionado de notificaciones y mensajes respectivamente) realizados por alumnos, en la mayoría de los casos de forma aislada y sin inicio de sesión previo, por lo que sería necesario esclarecer qué es lo que origina realmente esos registros.

- Respecto a las acciones con origen “ws”, que son acciones realizadas a través de la aplicación para móviles, es importante señalar que gran parte de las acciones se registran como “\core\event\webservice\_function\_called”, con la columna de “objecttable” vacía, que es la columna que indica en qué tabla se almacena la información relacionada. Sin embargo, no todas las acciones se registran así, ya que pueden observarse acciones como publicaciones en foros o envío de tareas con el mencionado origen. También aparecen otras acciones como, por ejemplo, “\mod\_quiz\event\course\_module\_viewed”, que se corresponde con el visionado de un cuestionario, seguido de varios registros de “\core\event\webservice\_function\_called”, por lo que se podría suponer, aunque no se puede asegurar con la información disponible, que dicho usuario ha respondido un cuestionario, sin que se registre de igual manera que si el origen fuese “web”, por lo que habría que establecer un método que permitiera identificar cada una de las acciones que realmente se están realizando bajo el registro de “\core\event\webservice\_function\_called”, ya que, aunque la columna “other” ofrece cierta información sobre el registro, no es del todo clara.
  
- Por último, debido a que se ha trabajado con extractos de una única tabla por centro y año para los dos años objeto de estudio, principalmente por las limitaciones del equipo informático empleado y por el tamaño de los datos, no es posible conocer la cantidad de recursos que ya existían en los cursos analizados, debido a que la información exacta acerca de los recursos ya existentes en un curso se almacena en otras tablas. Este hecho afecta directamente a la observación de los recursos eliminados, ya que, sin la lista de recursos existentes, carece de sentido observar cuántos recursos de un curso fueron eliminados, sería más relevante observar cuántos se eliminaron respecto al total ya existente.

## 4 Resultados

Los resultados obtenidos se analizarán de forma separada para cada uno de los 6 ficheros resultantes, a excepción de los ficheros relativos al tiempo conectado y al tiempo activo, que se analizarán de forma conjunta.

En la tabla IX se aporta el número de usuarios (tanto profesores como alumnos) por facultad y año, sin embargo, aunque no se tendrá en cuenta, veremos en los siguientes epígrafes que, el hecho de que una facultad tenga menor número de usuarios que otra, no implica directamente que tenga un menor número de registros que otra con más usuarios.

**Tabla IX.** Número de usuarios por centro y año

	<b>2019</b>	<b>2020</b>
<b>c03</b>	159	140
<b>c08</b>	223	166
<b>c12</b>	366	373
<b>c14</b>	117	91
<b>c16</b>	291	312
<b>c18</b>	256	266
<b>c28</b>	325	336
<b>c30</b>	541	519
<b>c36</b>	454	379

## 4.1 Tiempo conectado y activo

Los códigos relativos al análisis de los ficheros correspondientes al tiempo conectado y al tiempo activo se encuentran en el anexo VIII y el anexo IX, respectivamente.

Mientras que el fichero del tiempo conectado contiene 483.247 registros, el fichero del tiempo activo contiene 548.107 registros. Esto se debe a que, como se expuso en el apartado anterior, se han tomado diferentes tiempos máximos entre acciones.

En las siguientes tablas tenemos el tiempo medio, máximo y mínimo conectado y la desviación típica (Tabla X), así como los correspondientes al tiempo activo (Tabla XI), para ambos años. Como puede verse, los datos relativos al tiempo activo son muy inferiores a los del tiempo conectado, ya que, aunque los alumnos y profesores puedan llegar a permanecer largas horas conectados a la Moodle, existe una diferencia de más de veinte minutos entre el tiempo medio conectado y el activo y de hasta casi diez horas en el tiempo máximo. Esto no implica necesariamente que, de media, se tomen descansos cada 5 minutos, pues no sabemos si esa diferencia se debe a actividades de larga duración, como pueden ser el visionado de documentos o de videos (otras actividades como la realización de cuestionarios generan registros cada vez que se responde a una pregunta), a que estuvieron realizando acciones en otra asignatura o a que efectivamente dejaron de interactuar con la plataforma. No obstante, llama la atención que haya usuarios que permanecieron activos en la plataforma hasta 5 horas, frente a un tiempo máximo conectado de 14 horas y media. También es destacable el hecho de que la media de tiempo conectado y tiempo activo, así como los máximos, aumentaron en 2020.

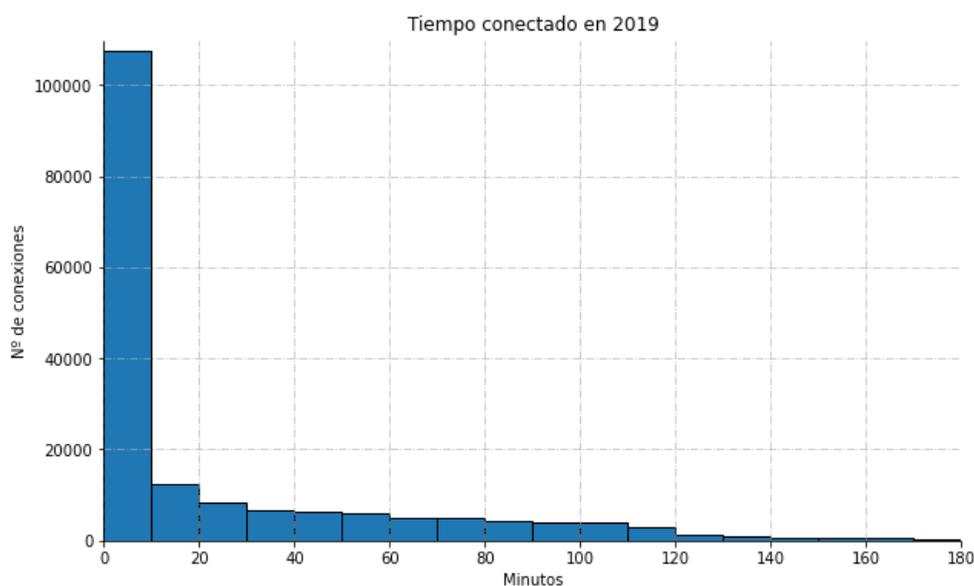
**Tabla X.** Tiempo medio, máximo y mínimo conectado y desviación típica por año

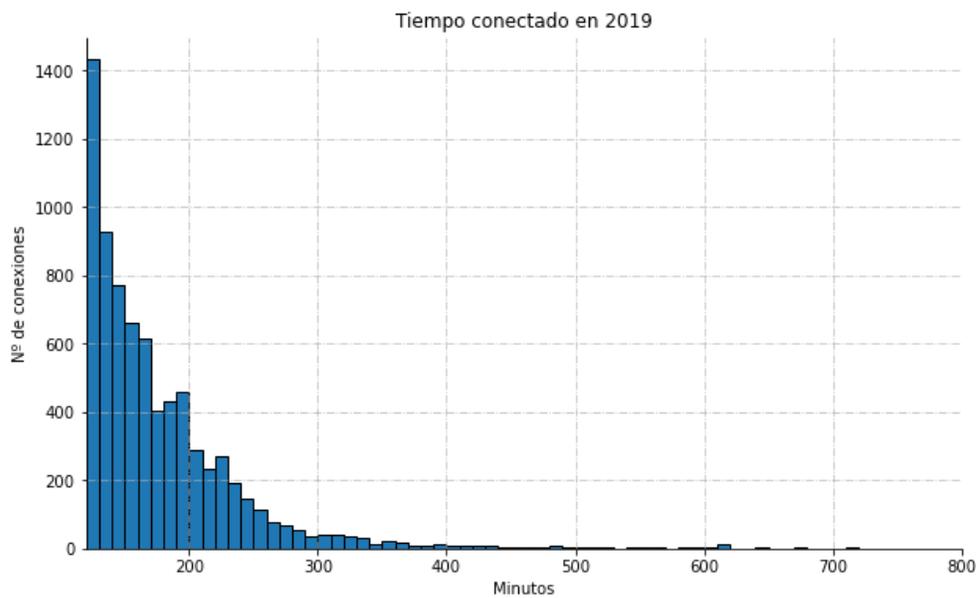
Año	Tiempo medio conectado	Tiempo máximo conectado	Tiempo mínimo conectado	Desviación típica
2019	00:27:34.204026	13:11:09	0 days	00:46:03.078802
2020	00:27:41.714894	14:31:44	0 days	00:49:49.009487

**Tabla XI.** Tiempo medio, máximo y mínimo activo y desviación típica por año

Año	Tiempo medio activo	Tiempo máximo activo	Tiempo mínimo activo	Desviación típica
2019	00:05:09.100876	03:35:19	0 days	00:09:45.139374
2020	00:05:38.416744	05:10:50	0 days	00:11:23.191542

En la figura I se puede observar que, del total de las 179.771 conexiones que se realizaron en 2019, más de 100.000 tuvieron una duración máxima de diez minutos, generando una gran diferencia con las conexiones superiores a diez minutos, ya que poco más de 12.000 conexiones tuvieron una duración de entre diez y veinte minutos. Si pasamos a observar la figura II, se puede apreciar que, aunque la mayoría de las conexiones tuvieron una duración inferior a las dos horas y, especialmente, a diez minutos, hubo varios miles de conexiones con una duración superior a las dos horas, de las cuales varios cientos de conexiones tuvieron una duración superior a las cuatro horas.

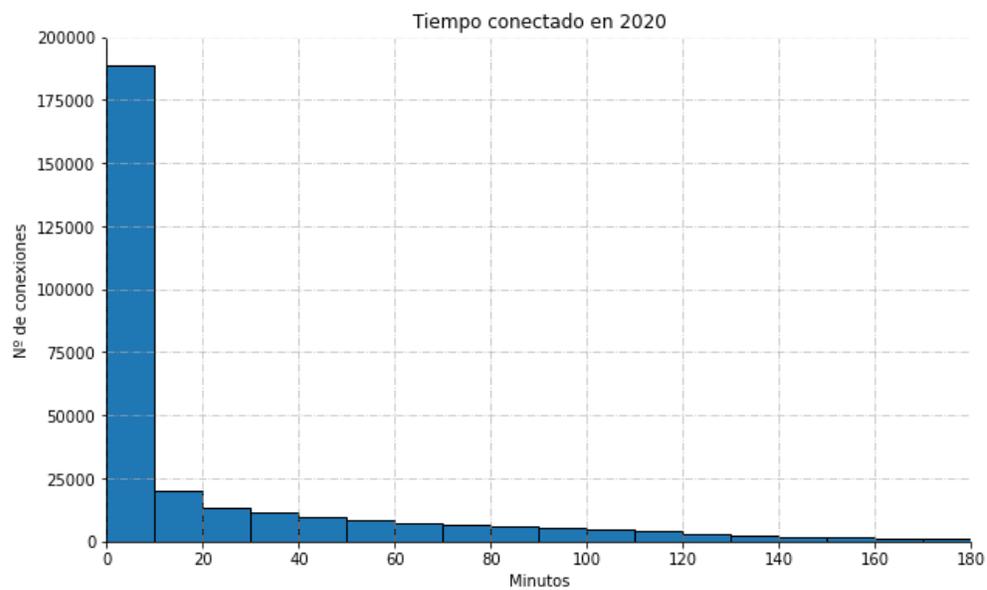
**Figura I.** Tiempo conectado en 2019 (0-180 minutos)



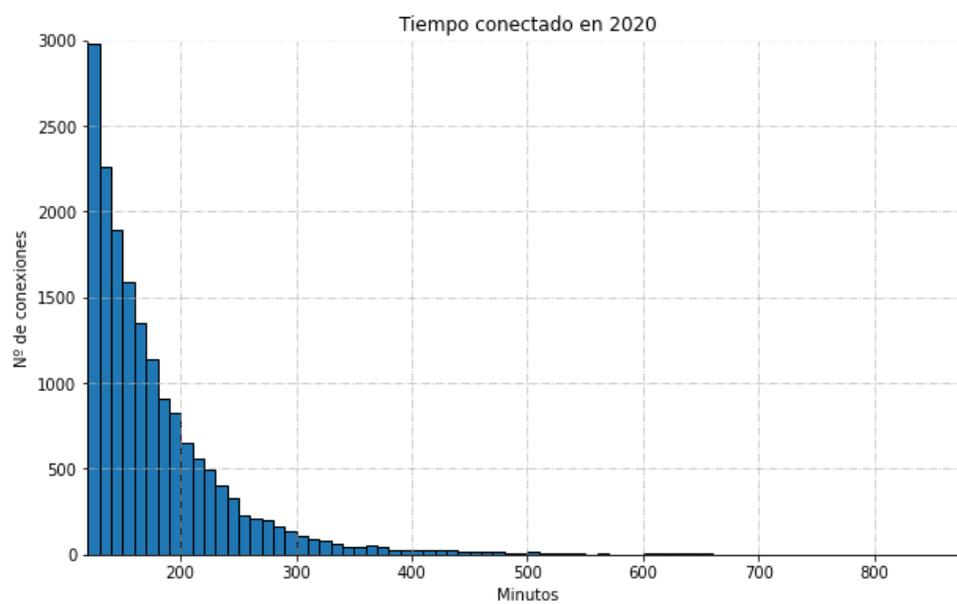
**Figura II.** Tiempo conectado en 2019 (120-800 minutos)

Si pasamos al año 2020, con un total de 303.477 conexiones, casi el doble que el año anterior, podemos ver en la figura III y en la figura IV que la distribución de la duración de las conexiones es prácticamente idéntica, aunque con unas cifras muy superiores, ya que se pasaron de 100.000 conexiones con duración inferior a diez minutos a casi 190.000 conexiones, más de la mitad del total de 2020, como ocurrió el año anterior. Sin embargo, y es destacable, aunque el número de conexiones de todos los tramos de duración aumentaron significativamente, hubo más del doble de conexiones con una duración superior a las dos horas que el año anterior, lo cual podría ser una consecuencia directa del confinamiento y la docencia online.

También cabe decir que, como se mencionó anteriormente, y aunque es imperceptible gráficamente dada la escala empleada, se dio una conexión de más de 790 minutos (13 horas) en 2019 y otra de casi 880 minutos (14 horas y media) en 2020.



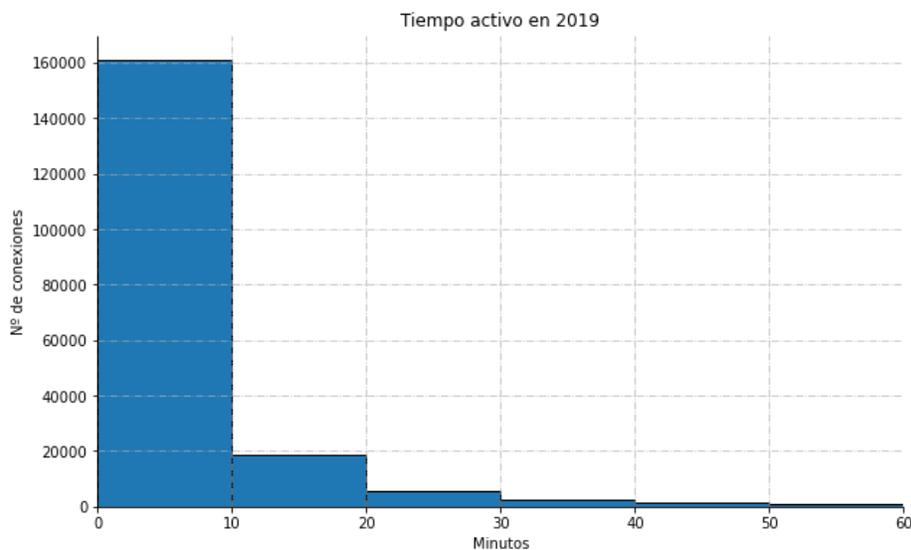
**Figura III.** Tiempo conectado en 2020 (0-180 minutos)



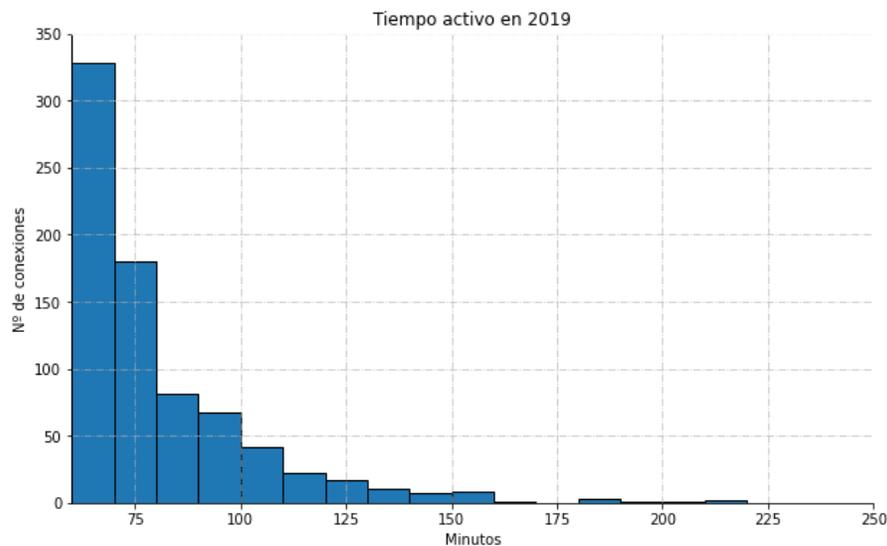
**Figura IV.** Tiempo conectado en 2020 (120-800 minutos)

A continuación, analizando el tiempo activo, podremos observar que, aunque la distribución del tiempo activo de ambos años es muy similar a la que acabamos de ver en el tiempo conectado, los minutos activos en los tramos superiores son bastante menores, puesto que ningún usuario ha sobrepasado los 310 minutos activos, que se dio en 2020, frente a la máxima de 880 minutos conectado del mismo año.

En la figura V podemos ver cómo, del total de los 191.324 registros del 2019, algo más de 160.000 tuvieron una duración de máximo diez minutos y casi 20.000 estuvieron entre diez y veinte minutos, cifras muy superiores a las del tiempo conectado del mismo año. A partir de los 20 minutos de tiempo activo, la cantidad de registros es bastante inferior a los del tiempo conectado, en torno a los 10.000, por lo que podría decirse que, en general, los usuarios permanecieron interactuando de forma activa con la plataforma un máximo de veinte minutos, con tan solo unos 500 registros de usuarios que permanecieron interactuando de forma activa más de una hora en el año 2019, como puede verse en la figura VI.



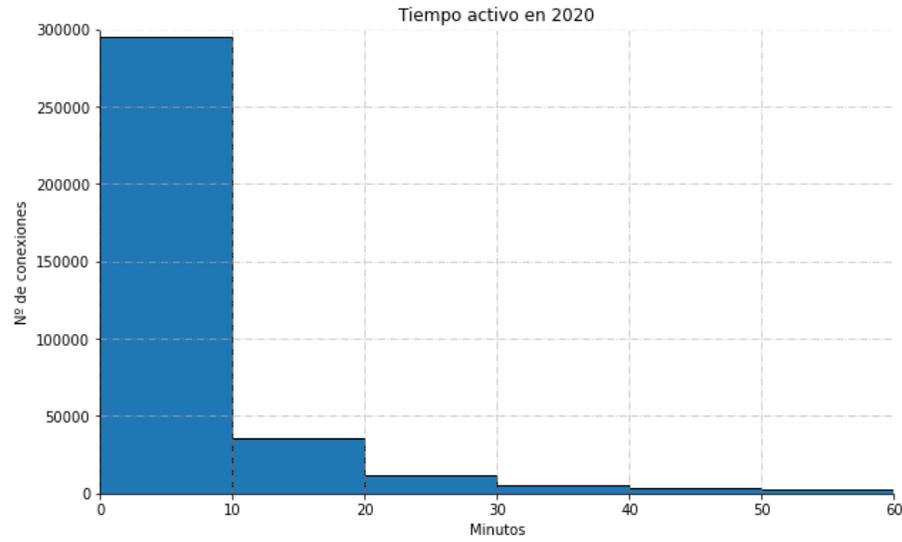
**Figura V.** Tiempo activo en 2019 (0-60 minutos)



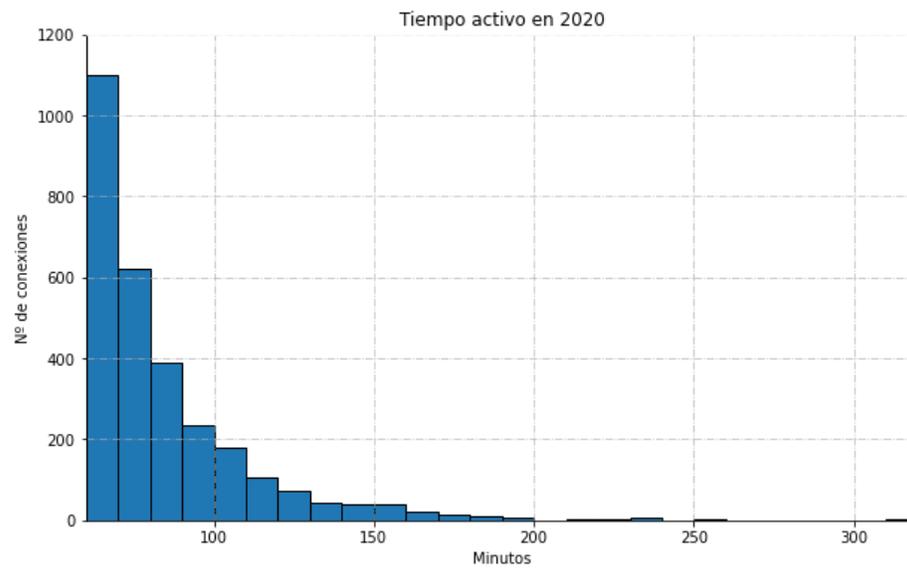
**Figura VI.** Tiempo activo en 2019 (60-250 minutos)

Por último, observando la figura VII y la figura VIII, podemos ver que, del total de los 356.784 registros del año 2020, casi 300.000 tuvieron una duración de máximo diez minutos, mientras que, algo más de 35.000 estuvieron, de forma activa, entre diez y veinte minutos, acumulando entre ambos tramos casi la totalidad de los registros, cifras, nuevamente, muy superiores a las del tiempo conectado para el mismo año. Por otro lado, y a diferencia del año anterior, existen en torno a 3.000 registros con una duración superior a los 60 minutos y unos 250 con una duración superior a las dos horas.

En general, puede afirmarse que los usuarios permanecieron interactuando con la plataforma de forma activa en más ocasiones y durante más tiempo durante el año 2020, es decir, durante el confinamiento, que durante el año anterior, al igual que sucedía con el tiempo conectado, ya que se registraron un mayor número de conexiones y de mayor duración durante el año 2020 que durante el 2019.



**Figura VII.** Tiempo activo en 2020 (0-60 minutos)



**Figura VIII.** Tiempo activo en 2020 (60-300 minutos)

Para terminar con el análisis del tiempo conectado y activo, vamos a analizar, de forma separada, los datos anteriores desglosados por centro, por mes y por día de la semana para ambos años.

En primer lugar, tenemos los datos del tiempo conectado desglosado por centro y año en la tabla XII, donde podemos ver que los usuarios de la ETSI fueron los que estuvieron, de media, mayor tiempo conectados a la plataforma durante el año 2019, con casi 32 minutos por sesión, seguidos muy de cerca, con una diferencia de dos segundos, por los usuarios de la Facultad de Ciencias Experimentales. En el extremo opuesto se encuentran los usuarios de la Facultad de Derecho, con casi 15 minutos y medio por sesión, una diferencia de ocho minutos y medio respecto a los usuarios de la segunda facultad con menor duración media, la Facultad de Humanidades.

Respecto a los datos del año 2020, fueron los usuarios de la Facultad de Ciencias Empresariales los que permanecieron, de media, mayor tiempo conectados, con poco más de 32 minutos por sesión, con una diferencia de 16 segundos respecto a la segunda con mayor duración media de la sesión, la ETSI, con 29 minutos y 50 segundos, mientras que los usuarios de la Facultad de Derecho fueron los que tuvieron una menor duración media, con 24 minutos y medio por sesión, seguidos de los usuarios de la Facultad de Ciencias de la Educación, con 25 minutos y cuarto por sesión.

En cuanto al tiempo máximo, en general todas las sesiones de ambos años fueron de entre 10 y 12 horas, salvo dos sesiones de 14 horas y media en el año 2020, una en la Facultad de Derecho y otra en la Facultad de Ciencias Empresariales, frente a una sesión de 8 horas y 20 minutos de la Facultad de Humanidades.

Si comparamos la evolución de un año respecto a otro, podemos ver que, sorprendentemente, tan solo aumentó el tiempo medio conectado por sesión en cuatro de las nueve facultades, es decir, en la Facultad de Humanidades, la Facultad de Trabajo Social, la Facultad de Derecho y la Facultad de Ciencias Empresariales, mientras que, en las demás las facultades, el tiempo medio conectado por sesión se vio disminuido. El aumento del tiempo medio de conexión, en general, fue de un minuto, salvo en el caso de la Facultad de Derecho que aumentó casi 9 minutos.

**Tabla XII.** Tiempo medio, máximo y mínimo conectado y desviación típica, por centro y año

<b>Centro</b>	<b>Año</b>	<b>Tiempo medio conectado</b>	<b>Tiempo máximo conectado</b>	<b>Tiempo mínimo conectado</b>	<b>Desviación típica</b>
c03	2019	00:31:56.448861	12:10:47	0 days	00:54:20.975957
	2020	00:26:49.899898	12:30:29	0 days	00:50:01.545580
c08	2019	00:23:59.964668	12:53:15	0 days	00:41:28.740400
	2020	00:25:25.486901	08:23:47	0 days	00:46:21.510418
c12	2019	00:25:03.197242	11:59:41	0 days	00:41:11.394447
	2020	00:26:10.581905	11:55:32	0 days	00:48:46.529350
c14	2019	00:29:13.178522	11:55:48	0 days	00:46:52.600740
	2020	00:26:38.076516	11:19:47	0 days	00:47:04.248939
c16	2019	00:27:42.635964	13:11:09	0 days	00:47:10.611779
	2020	00:27:32.202577	10:36:24	0 days	00:49:21.434784
c18	2019	00:31:58.391784	10:46:19	0 days	00:46:53.374770
	2020	00:29:51.907641	12:42:10	0 days	00:53:47.167316
c28	2019	00:15:27.021012	10:40:16	0 days	00:33:37.755702
	2020	00:24:32.886916	14:31:44	0 days	00:48:13.197512
c30	2019	00:28:59.813056	12:09:10	0 days	00:44:59.056546
	2020	00:32:07.996226	14:22:45	0 days	00:53:44.515638
c36	2019	00:30:28.190571	11:13:20	0 days	00:50:52.489257
	2020	00:25:16.716062	10:21:46	0 days	00:43:34.089186

En la tabla XIII tenemos los datos relativos al tiempo activo, en la que podemos ver que, durante el año 2019, fueron los usuarios de la Facultad de Enfermería los que, de media, permanecieron mayor tiempo activos en la plataforma, con 6 minutos y 42 segundos por sesión, con una diferencia de 40 segundos respecto a los usuarios de la Facultad de Ciencias de la Educación. En el otro extremo, fueron los usuarios de la Facultad de Derecho los que permanecieron menos tiempo activos en la plataforma, con 3 minutos y cuarto, seguidos por los usuarios de la Facultad de Ciencias Experimentales, con poco más de 4 minutos.

Respecto a los datos del año 2020, fueron nuevamente los usuarios de la Facultad de Enfermería los que, de media, permanecieron mayor tiempo activos, con 6 minutos y 20 segundos, seguidos muy de cerca por los usuarios de la Facultad de Ciencias Empresariales y de la Facultad de Ciencias de la Educación. Por el contrario, fueron los usuarios de la Facultad de Ciencias Experimentales los que permanecieron menor tiempo activos en la plataforma, con casi 4 minutos y 20 segundos por sesión, seguidos por los usuarios de la Facultad de Humanidades, con apenas 30 segundos de diferencia.

En cuanto al tiempo máximo activo, tal y como ocurría con el tiempo conectado, todas las facultades tienen unos datos muy similares, con sesiones entre 2 y 4 horas. Destaca una sesión de 5 horas, en 2020, por un usuario de la Facultad de Educación como la de mayor duración y otras dos sesiones de una hora y 45 minutos, en 2019, por usuarios de la Facultad de Humanidades y de la Facultad de Derecho.

Comparando la diferencia temporal, el tiempo medio activo creció en la mayoría de las facultades en torno a 30 segundos, los usuarios de la Facultad de Enfermería tuvieron unos segundos menos de tiempo medio activos y el de los usuarios de la Facultad de Derecho se incrementó en 2 minutos.

**Tabla XIII.** Tiempo medio, máximo y mínimo activo y desviación típica, por centro y año

Centro	Año	Tiempo medio activo	Tiempo máximo activo	Tiempo mínimo activo	Desviación típica
c03	2019	00:04:06.814612	02:16:23	0 days	00:07:26.572662
	2020	00:04:17.517041	03:51:34	0 days	00:08:25.976352
c08	2019	00:04:35.652576	01:44:34	0 days	00:08:04.479343
	2020	00:04:56.497894	02:37:32	0 days	00:09:56.232767
c12	2019	00:05:22.653333	03:35:19	0 days	00:10:41.734540
	2020	00:05:34.560215	03:58:02	0 days	00:11:50.373345
c14	2019	00:05:02.462335	02:15:05	0 days	00:08:31.505284
	2020	00:05:36.591024	02:02:08	0 days	00:10:32.474298
c16	2019	00:06:42.658235	03:30:20	0 days	00:12:59.112449
	2020	00:06:19.011639	04:24:20	0 days	00:12:37.113950
c18	2019	00:04:23.828216	02:12:32	0 days	00:08:16.208708
	2020	00:05:06.283892	02:38:10	0 days	00:10:05.443301
c28	2019	00:03:16.033277	01:49:17	0 days	00:06:36.342237
	2020	00:05:17.020567	03:20:49	0 days	00:10:39.730308
c30	2019	00:05:18.853706	02:26:01	0 days	00:09:46.268570
	2020	00:06:02.987882	04:00:43	0 days	00:11:48.094149
c36	2019	00:06:03.858117	02:24:54	0 days	00:10:23.819552
	2020	00:06:04.310716	05:10:50	0 days	00:12:36.208659

En segundo lugar, tenemos los datos desglosados por mes y año para el tiempo conectado en la tabla XIV, ordenado, de menor a mayor, en base al tiempo medio conectado. En la tabla puede apreciarse como, en 2019, junio y abril fueron los meses con menor tiempo medio conectado, frente a marzo y mayo como los meses con mayor tiempo conectado, comprendiéndose dichos tiempos entre 25 y 29 minutos. Sin embargo, en el año 2020 fueron los meses de febrero y marzo los que tuvieron menor tiempo medio de conexión, frente a mayo y abril como los meses con mayor media. Es destacable el hecho de que, en febrero de 2020, el tiempo medio de conexión

fue de casi 18 minutos y medio, muy inferior a los 26 minutos y 20 segundos de marzo de 2020 y los 25 minutos de junio de 2019 (el de menor media de ese año), y que junio de 2020, el mes en que tienen lugar los exámenes finales, fuera el tercer mes de su año con una media de 27 minutos, mientras que fue abril de 2020 el mes con mayor tiempo medio de conexión, con 32 minutos y 45 segundos, coincidiendo con la primera prórroga del Estado de Alarma en España, superando en 4 minutos y medio la media de mayo de 2020 y en más de 6 minutos y medio la media del mismo mes de 2019 .

**Tabla XIV.** Tiempo medio, máximo y mínimo conectado y desviación típica, por mes y año

<b>Año</b>	<b>Mes</b>	<b>Tiempo medio conectado</b>	<b>Tiempo máximo conectado</b>	<b>Tiempo mínimo conectado</b>	<b>Desviación típica</b>
2019	Junio	00:24:59.604902	11:55:48	0 days	00:46:35.372708
	Abril	00:26:03.038285	10:15:01	0 days	00:44:34.387379
	Febrero	00:28:49.283615	13:11:09	0 days	00:48:23.517115
	Marzo	00:28:56.475405	11:13:20	0 days	00:45:31.702411
	Mayo	00:29:08.086595	12:10:47	0 days	00:45:02.907675
2020	Febrero	00:18:29.401639	14:22:45	0 days	00:41:09.155178
	Marzo	00:26:18.614733	10:21:46	0 days	00:46:37.832443
	Junio	00:27:07.133084	12:42:10	0 days	00:51:41.662849
	Mayo	00:28:13.103114	14:15:24	0 days	00:48:47.684970
	Abril	00:32:45.153328	14:31:44	0 days	00:53:56.502197

Respecto a los datos del tiempo activo, ordenados de la misma forma que los del tiempo conectado, podemos ver, en la tabla XV, que existen leves variaciones en cuanto al orden de los meses en comparación con los del tiempo conectado. Por un lado, en el año 2019, fueron junio y febrero los meses de menor media y mayo y marzo los que tuvieron un mayor tiempo medio activo, con una diferencia de un minuto, mientras que, en el año 2020, fueron febrero y mayo los

que tuvieron menor media y abril y junio los meses con mayor tiempo medio activo, con una diferencia de hasta 2 minutos entre febrero y junio.

**Tabla XV.** Tiempo medio, máximo y mínimo activo y desviación típica, por mes y año

Año	Mes	Tiempo medio activo	Tiempo máximo activo	Tiempo mínimo activo	Desviación típica
2019	Junio	00:04:22.696472	03:07:48	0 days	00:09:28.796323
	Febrero	00:04:55.889741	03:35:19	0 days	00:08:55.643736
	Abril	00:05:28.393606	02:34:06	0 days	00:09:59.656073
	Mayo	00:05:29.598609	03:30:20	0 days	00:10:21.414163
	Marzo	00:05:31.424390	02:15:05	0 days	00:09:45.058835
2020	Febrero	00:04:08.279330	02:14:28	0 days	00:08:15.506346
	Mayo	00:05:03.869919	04:24:20	0 days	00:09:47.523805
	Marzo	00:05:48.796438	02:56:52	0 days	00:11:21.509389
	Abril	00:06:09.690472	03:55:41	0 days	00:11:13.329642
	Junio	00:06:14.885018	05:10:50	0 days	00:14:02.452854

Por último, tenemos los datos desglosados por día de la semana y año, ordenados también de forma ascendente en base al tiempo medio conectado, en la tabla XVI. En ella podemos ver que, tanto el sábado como el domingo fueron los días de la semana con menor media de tiempo conectado, con en torno a 7 minutos de diferencia entre el último y el primero, además la media de dichos días fue muy superior en 2019 respecto a 2020. En cuanto a los días con mayor media de tiempo conectado, en 2019 fueron el martes y el lunes con poco más de 29 minutos y muy poca diferencia entre ellos, y en 2020 fue el martes con una media de 29 minutos y medio, seguido del jueves, con una media de casi 29 minutos. Llama la atención el hecho de que, aunque la diferencia entre los días del fin de semana y el resto de días de la semana es mayor en 2020 que en 2019, la media de tiempo conectado para los días de entre semana es más estable en 2020 que durante el año anterior, con una diferencia menor entre días. También destaca que fuera un

sábado de 2020 uno de los días de la semana con mayor tiempo de conexión, mientras que el mismo día del año anterior tiene el registro más bajo para el tiempo máximo conectado.

**Tabla XVI.** Tiempo medio, máximo y mínimo conectado y desviación típica, por día y año

<b>Año</b>	<b>Día</b>	<b>Tiempo medio conectado</b>	<b>Tiempo máximo conectado</b>	<b>Tiempo mínimo conectado</b>	<b>Desviación típica</b>
2019	Domingo	00:22:50.607545	13:11:09	0 days	00:43:30.038922
	Sábado	00:23:40.060699	09:40:54	0 days	00:47:16.597008
	Jueves	00:26:52.985936	12:53:15	0 days	00:44:33.553728
	Miércoles	00:27:12.220456	12:09:10	0 days	00:45:36.247405
	Viernes	00:28:28.887226	11:59:41	0 days	00:46:31.122564
	Martes	00:29:04.297940	12:10:47	0 days	00:45:42.099866
	Lunes	00:29:16.444545	11:31:31	0 days	00:47:44.491624
2020	Sábado	00:21:42.493490	14:15:24	0 days	00:43:40.978414
	Domingo	00:22:02.020264	12:30:29	0 days	00:46:37.089337
	Miércoles	00:28:20.005472	10:59:28	0 days	00:49:09.919767
	Viernes	00:28:38.374165	11:55:32	0 days	00:49:40.985845
	Lunes	00:28:52.357354	14:31:44	0 days	00:51:42.050200
	Jueves	00:28:56.794776	12:42:22	0 days	00:50:26.122019
	Martes	00:29:28.335593	10:48:41	0 days	00:51:35.477266

Para los datos del tiempo activo, podemos ver en la tabla XVII que, aunque los días con menor tiempo medio activo no han variado respecto a los del tiempo conectado, los días con mayor tiempo medio activo sí que han variado, en este caso, para 2019 tenemos el lunes y el jueves con casi 5 minutos y medio, algo inferior a los días con mayor tiempo activo del 2020, que fueron el miércoles, con casi 6 minutos, y el viernes, con cerca de 6 minutos y medio. El tiempo medio activo de 2019 se reparte más uniformemente que el tiempo medio conectado, con casi un

minuto de diferencia entre el último y el primero, frente al minuto y medio de 2020. Casi todos los tiempos medios de actividad de 2020, especialmente fuera del fin de semana, fueron superiores a los del 2019, con una diferencia de entre medio minuto y un minuto, de un año respecto a otro, entre los mismos días de la semana, a excepción del jueves, que prácticamente registró el mismo tiempo en ambos años.

**Tabla XVII.** Tiempo medio, máximo y mínimo activo y desviación típica, por día y año

<b>Año</b>	<b>Día</b>	<b>Tiempo medio activo</b>	<b>Tiempo máximo activo</b>	<b>Tiempo mínimo activo</b>	<b>Desviación típica</b>
2019	Sábado	00:04:39.699783	03:16:51	0 days	00:09:29.659150
	Domingo	00:04:44.253306	03:30:20	0 days	00:09:10.212831
	Viernes	00:04:57.007182	03:07:37	0 days	00:08:57.660822
	Miércoles	00:05:04.585396	02:38:25	0 days	00:09:38.848457
	Martes	00:05:07.255393	03:35:19	0 days	00:09:59.671022
	Lunes	00:05:26.298298	02:34:06	0 days	00:09:49.991337
	Jueves	00:05:27.234132	02:21:18	0 days	00:10:23.692863
2020	Domingo	00:04:39.657287	02:55:11	0 days	00:09:51.550150
	Sábado	00:04:49.389657	03:51:34	0 days	00:09:53.950294
	Jueves	00:05:31.715087	03:15:10	0 days	00:10:43.054182
	Miércoles	00:05:44.076063	04:00:43	0 days	00:11:43.261055
	Martes	00:05:51.046746	04:24:20	0 days	00:11:38.731731
	Lunes	00:05:51.727522	03:48:10	0 days	00:11:48.484607
	Viernes	00:06:09.744010	05:10:50	0 days	00:12:23.829979

## 4.2 Otras acciones

Los códigos relativos al análisis del fichero correspondiente a otras acciones se encuentran en el anexo X, que, recordemos, recoge la información relativa al número de inicios de sesión, el total de acciones realizadas, el número de veces que se visionaron las calificaciones dentro de la plataforma y el número de mensajes enviados.

Hay que matizar que el número de inicios de sesión no coincidirá con el número de registros del tiempo conectado ni del tiempo activo, a pesar de estar directamente relacionados, ya que, como se mencionó en apartados anteriores, las acciones con origen “cli” y “ws” no siempre van precedidas por un inicio de sesión, por lo que, en su mayoría, los inicios de sesión habrán sido realizados desde la plataforma web. Por otro lado, el número de acciones excluye tanto los inicios como cierres de sesión que hayan realizado los usuarios.

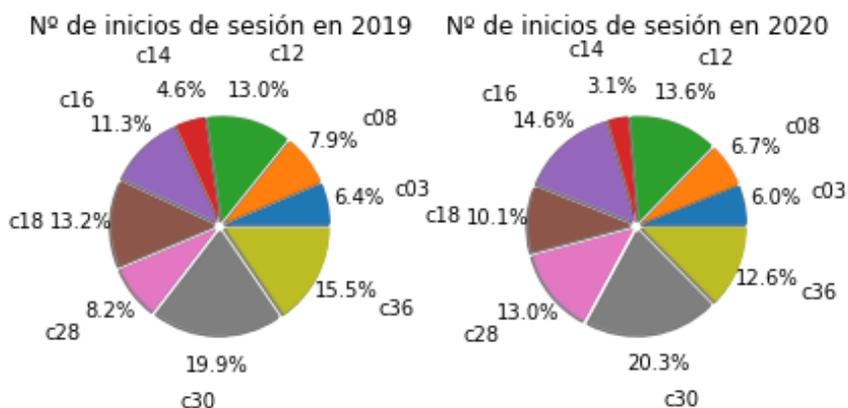
El fichero contiene la información mencionada anteriormente agrupada por usuario, centro, hora (en tramos de una hora, sin minutos), fecha y semana (número de semana del año), sumando un total de 956.924 registros.

En un primer vistazo al contenido de este fichero (Tabla XVIII) podemos ver que el número de acciones se duplicó y el total de acciones casi se triplicó, mientras, curiosamente, el número de mensajes enviados a través de la plataforma cayó muy por debajo de la mitad respecto al año anterior.

**Tabla XVIII.** Resumen del fichero de otras acciones por año

	<b>2019</b>	<b>2020</b>	<b>Total</b>
<b>nis</b>	299944	595763	895707
<b>t_acc</b>	1310874	3483853	4794727
<b>vistas_calif</b>	9433	14317	23750
<b>num_mensajes_enviados</b>	92236	36933	129169

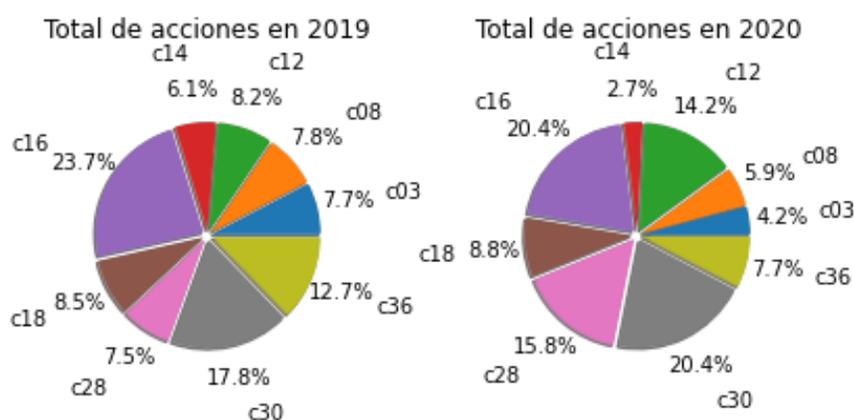
Si observamos como se reparten los inicios de sesión por centro (Figura IX) podemos ver que prácticamente ambos gráficos se dividen de igual modo, con ligeras variaciones de un año a otro, aunque todas tuvieron, aproximadamente, el doble de inicios de sesión en 2020 respecto a 2019. Fue la Facultad de Ciencias Empresariales la que registró mayor número de inicios de sesión en ambos años, con 59.803 conexiones en 2019 y 120.674 en 2020, la segunda de 2019 fue la Facultad de Ciencias de la Educación, con 46.422 conexiones, y la de 2020 fue la Facultad de Enfermería, con 86.895 conexiones. Sin embargo, fue la Facultad de Derecho la que tuvo mayor crecimiento, pasando de 24.667 conexiones en 2019 a 77.399 en 2020. La que registró menor número de conexiones fue la Facultad de Ciencias del Trabajo, con 13.668 conexiones en 2019 y 18.405 en 2020.



**Figura IX.** Número de inicios de sesión por centro y año

En cambio, si observamos la figura X con el total de acciones, podemos ver que la distribución del gráfico sufre más variaciones de un año para otro. Sin duda alguna, el cambio más drástico lo tuvo la Facultad de Derecho, que pasó de 98.364 acciones en 2019 a 549.806 en 2020, mientras que, las que menos cambios tuvieron fueron la Facultad de Ciencias Experimentales, que pasó de 100.363 acciones en 2019 a 145.169 acciones en 2020, y la Facultad de Ciencias del Trabajo, que pasó de 79.539 acciones en 2019 a 94.193 acciones en 2020.

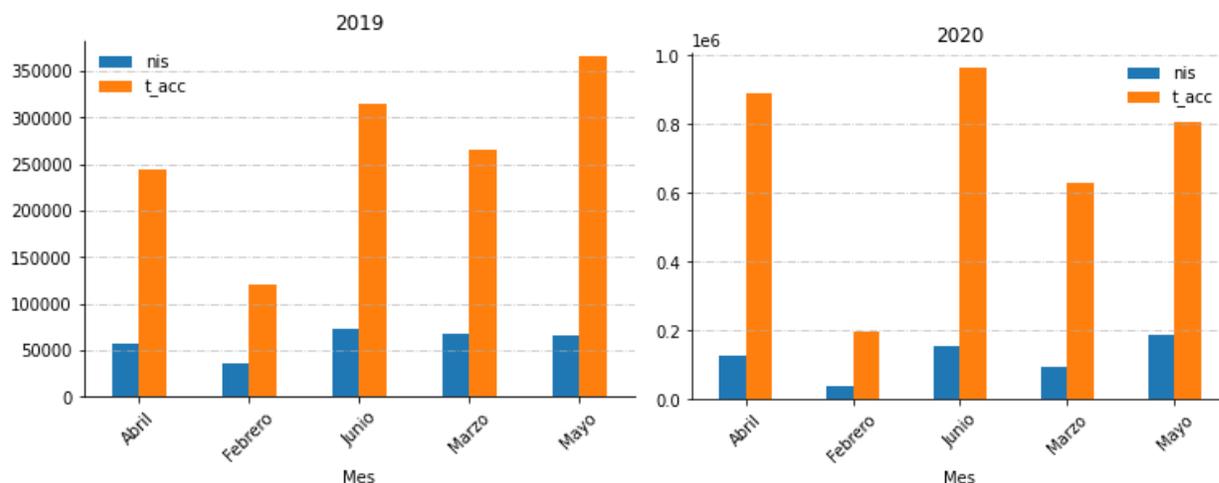
En 2019 fue la Facultad de Enfermería la que más acciones realizó, con un total de 310.842, mientras que la que menos acciones realizó fue la Facultad de Ciencias del Trabajo. En 2020, la Facultad de Ciencias Empresariales y la Facultad de Enfermería tuvieron cifras muy similares, con 711.563 y 709.710 acciones respectivamente, frente a la Facultad de Ciencias del Trabajo, que volvió a ser la que menos acciones realizó dentro de la plataforma.



**Figura X.** Total de acciones por centro y año

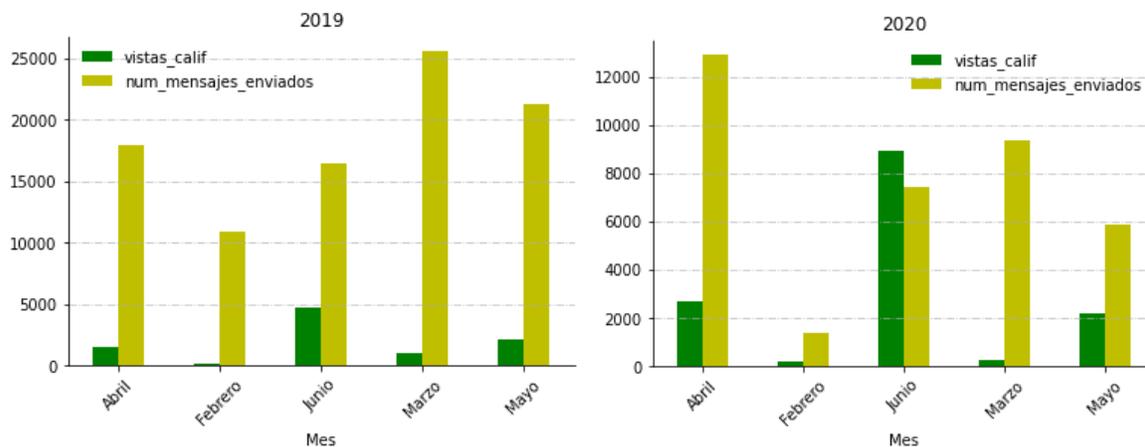
En cuanto a las cifras desglosadas por meses, se ha decidido separar el número de inicios de sesión y el total de acciones de las visualizaciones de las calificaciones y los mensajes enviados ya que, el número tan elevado de las dos primeras hace imperceptible, gráficamente, a las otras dos.

En primer lugar, en la figura XI tenemos el número de inicios de sesión y el total de acciones desglosados por mes y año. Observando las escalas de ambos gráficos puede apreciarse cómo las cifras de ambos datos prácticamente se duplican. En ambos años, el número de inicios de sesión tiene su mínimo en febrero y su máximo en junio, sin embargo, el pico de acciones realizadas en la plataforma pasa de ser mayo en 2019 a ser junio en 2020, probablemente como consecuencia del elevado número de cuestionarios y tareas que se crearon en dichos meses, como veremos en el apartado siguiente, mientras que el mínimo de acciones realizadas se produce en febrero en ambos años.



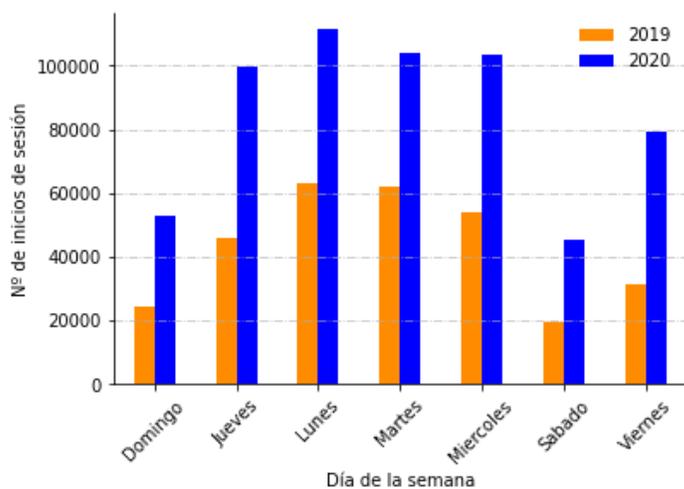
**Figura XI.** Número de inicios de sesión y total de acciones por mes y año

En segundo lugar, en la figura XII tenemos las visualizaciones de las calificaciones y los mensajes enviados, ambos dentro de la plataforma Moodle. Por un lado, y es bastante sorprendente, puede apreciarse gráficamente que el número de mensajes enviados a través de la plataforma descendió bastante en 2020 respecto al año anterior. Aunque hay que decir que ambos elementos no suelen utilizarse a menudo, ya que los profesores suelen publicar las calificaciones en los tabloneros de las facultades o mediante foros de avisos dentro de la plataforma. Sin embargo, se produjo un pico bastante notorio en cuanto a visualizaciones de calificaciones en junio de 2020, principalmente por la realización de exámenes online, entre otras formas de evaluación, para afrontar la docencia online. También llama la atención de que, mientras los mensajes enviados dentro de la plataforma se mantuvieron relativamente estables en 2019 a lo largo de todo el cuatrimestre, en 2020 apenas se enviaron mensajes en febrero, sino que la mayoría se enviaron el resto del cuatrimestre, es decir, durante el confinamiento.



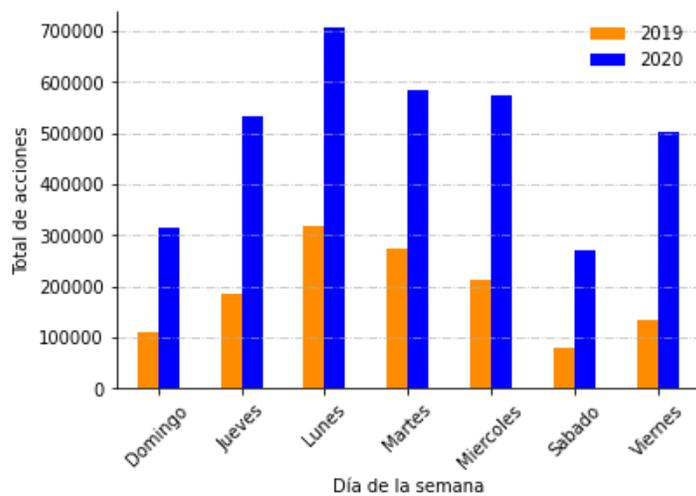
**Figura XII.** Visualizaciones de calificaciones y mensajes enviados por mes y año

Si en lugar de por mes, agrupamos el número de inicios de sesión por día de la semana, podemos observar, en la figura XIII, que las conexiones se realizan, sobre todo, entre semana, aunque en menor número los viernes, siendo el sábado el día con menor número de inicios de sesión y el lunes el día con mayor número en ambos años. Además, el número de conexiones se aprecia más estable de lunes a jueves en 2020 que en 2019.



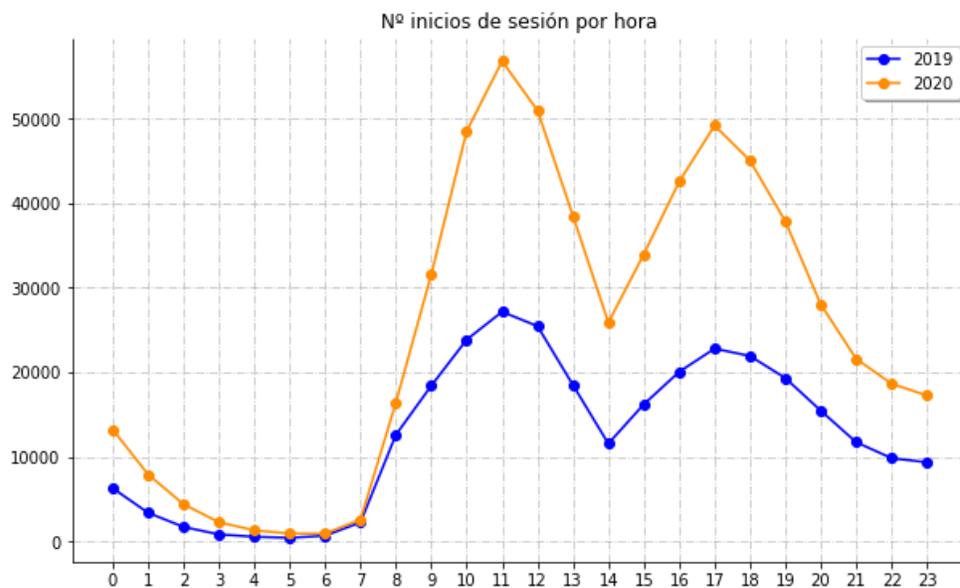
**Figura XIII.** Número de inicios de sesión por día y año

Si observamos el total de acciones en la figura XIV, agrupado de la misma forma que acabamos de ver con el número de inicios de sesión, podemos ver claramente que es el lunes el día en que se interactúa más veces con la plataforma y el sábado el día que menos se hace. Además, mientras que en 2019 el número de acciones en la plataforma descendía conforme avanzaba la semana, en 2020 se mantiene bastante estable de lunes a viernes.



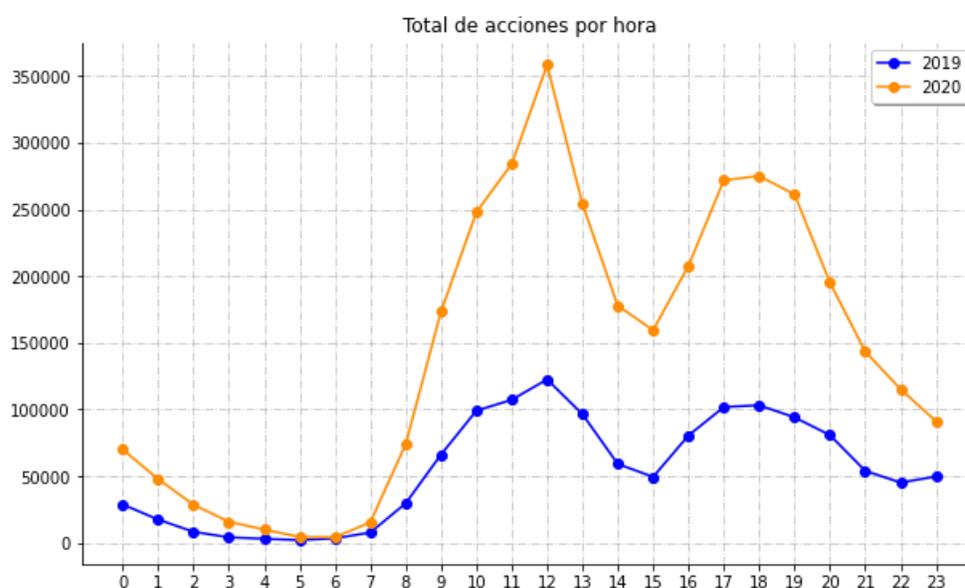
**Figura XIV.** Total de acciones por día y año

En la figura XV se encuentran los inicios de sesión agrupados por hora. Es muy llamativo el hecho de que, a pesar de que durante la madrugada los inicios de sesión van cayendo, en ningún momento del día la línea llega a tocar el 0, produciéndose los principales picos a las 11 de la mañana y a las 5 de la tarde, con una gran caída en cuanto a inicios de sesión a las 2 de la tarde.



**Figura XV.** Números de inicios de sesión por hora y año

Si observamos la misma gráfica, pero para el total de acciones (Figura XVI), podemos ver que, aunque los picos prácticamente coinciden, como sucedía con los inicios de sesión, en 2019 son mucho más suaves que en 2020, sin embargo no coincide del todo con la gráfica de inicios de sesión, ya que el pico de actividad de la mañana se produce a las 12, una hora después del pico de inicios de sesión, y el pico de la tarde es más alargado en el tiempo, produciéndose entre las 5 y las 7 de la tarde, mientras que el momento de menor actividad se produce a las 3 de la tarde en ambos años. Además, nuevamente, en ningún momento del día la gráfica llega a tocar el 0.



**Figura XVI.** Total de acciones por hora y año

### 4.3 Foros

Los códigos relativos al análisis del fichero correspondiente a los foros se encuentran en el anexo XI, el cual contiene las cantidades de foros, discusiones y posts que se crearon en los cursos, así como la diferencia entre los creados y los borrados (netos). Esta distinción se realizó para tener una visión fidedigna de cuántas publicaciones se hicieron en cada curso.

Se debe aclarar que, debido a que se trabajó exclusivamente con una tabla, no es posible saber cuántas discusiones se crearon por cada foro, ni cuantas de esas discusiones obtuvieron o no respuesta.

El fichero contiene 2.393 registros agrupados, al igual que los datos del fichero del apartado anterior, por usuario, centro, hora (en tramos de una hora, sin minutos), fecha y semana (número de semana del año), además del curso.

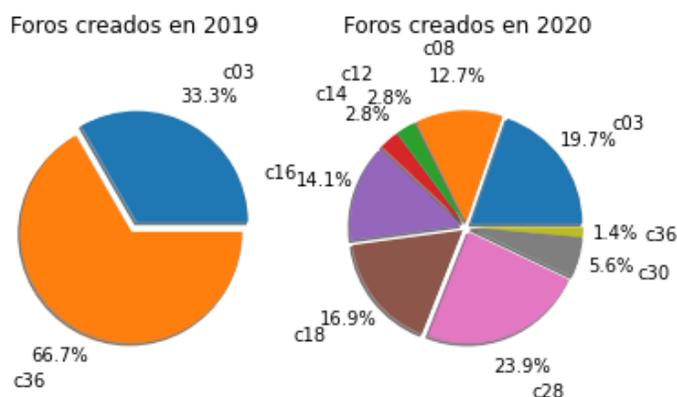
En la tabla XIX tenemos el número de foros, discusiones y post creados para ambos años, así como el total. Llama bastante la atención que, mientras el número de discusiones aumenta en algo más del doble, los foros y posts aumentaron considerablemente, pasando de tres foros creados en 2019 hasta 56 foros que permanecieron sin ser eliminados y de 175 posts netos a 948. Además, mientras que en 2019 había, aproximadamente, un posts por cada dos discusiones, en 2020 el número de posts superó, por poco, al número de discusiones.

**Tabla XIX.** Foros, discusiones y posts creados y netos por año

	2019	2020	Total
foros_creados	3	71	74
foros_creados_netos	3	56	59
discusiones_creadas	434	1159	1593
discusiones_creadas_netos	397	859	1256
posts_creados	189	1252	1441
posts_creados_netos	175	948	1123

Que tan solo aparezcan tres foros creados, en 2019, frente a un número tan elevado de discusiones y posts nos puede indicar que los profesores, en general, optaban por reutilizar los foros de años anteriores, mientras que, en 2020, optaron por crear nuevos foros, ya fuera para crear foros de dudas específicos para cada tema de la asignatura o para llevar a cabo, por ejemplo, actividades de debate.

A continuación, en la figura XVII, podemos ver la proporción de foros creados por centro. Como se mencionó anteriormente, de los tres foros creados en 2019, dos pertenecían a la Facultad de Ciencias de la Educación y uno a la Facultad de Ciencias Experimentales. En el año 2020, fue la Facultad de Derecho la que más foros creó, con un total de 17 foros (de los que fueron borrados 5), seguida de la Facultad de Ciencias Experimentales en las que se crearon 14 y se borraron 3. En el otro extremo, fue la Facultad de Ciencias de la Educación la que menos foros creó, con un nuevo foro, seguida de la Facultad de Trabajo Social y la Facultad de Ciencias del Trabajo, en las que se crearon dos nuevos foros.



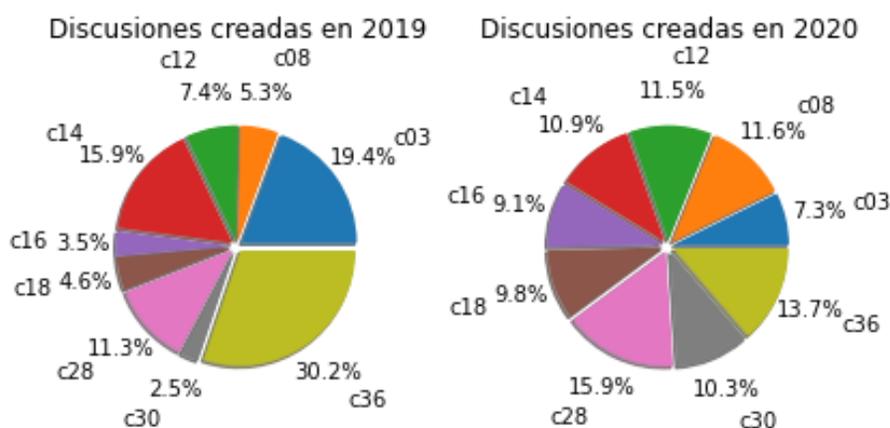
**Figura XVII.** Foros creados en 2019 y 2020 por centro

En cuanto a las discusiones creadas (Figura XVIII), aunque el gráfico de 2019 está repartido entre todas las facultades, destacan las dos únicas que crearon foros ese año, la Facultad de Ciencias de la Educación y la Facultad de Ciencias experimentales, con 131 y 84 discusiones, respectivamente, creadas ese año, que se quedaron en 126 y 80 respectivamente, considerando las borradas. En el extremo opuesto, fue la Facultad de Ciencias Empresariales la que menos

discusiones creó, con un total 11 discusiones (8 considerando las borradas), seguida de la Facultad de Enfermería con 15 discusiones creadas y ninguna borrada.

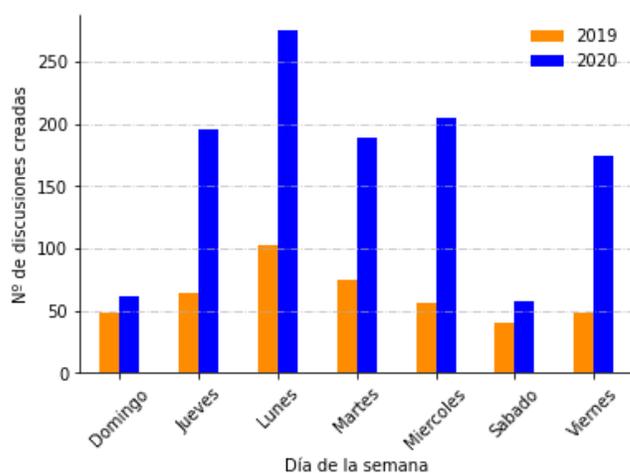
En el año 2020, fue la Facultad de Derecho la que más discusiones creó, siendo también la que más foros creó, con un total de 184 discusiones, sin embargo, se quedaron en 114 discusiones considerando las borradas, pasando a ser la quinta en discusiones netas, lejos de la Facultad de Ciencias de la Educación, con 159 discusiones creadas y ninguna borrada (la cual sería la segunda en discusiones creadas sin tener en cuenta las borradas).

Por otro lado, y llama bastante la atención, la Facultad de Ciencias Experimentales pasó de 85 discusiones a 46, la Facultad de Trabajo Social pasó de nada menos que 133 discusiones a 21 y la ETSI pasó de 114 a 51, considerando las borradas en los tres casos. Tras comprobar las discusiones creadas y las netas de estas tres facultades y de la Facultad de Derecho, no se puede asegurar que fueran los usuarios los que borrarán sus propias discusiones tras publicarlas o que fuera provocado por un borrado masivo realizado por algún profesor o gestor de la base de datos, probablemente para que los alumnos no tuvieran acceso a las discusiones antes de las pruebas de evaluación continua y de los exámenes, siendo la primera opción la más probable, ya que, tras buscar las discusiones creadas netas, así como los foros, con un valor negativo, no aparecen grandes cantidades de discusiones o foros borrados repentinamente.



**Figura XVIII.** Discusiones creadas en 2019 y 2020 por centro

Continuando con las discusiones, si las agrupamos por día de la semana (Figura XIX), podemos apreciar que fue el lunes el día de la semana en que más discusiones se crearon, con 103 en 2019 y 275 en 2020 (96 y 211, respectivamente, si tenemos en cuenta las borradas), aunque la diferencia con el resto de días de la semana no es muy elevada, en torno a 50 discusiones menos, salvo los días del fin de semana, siendo el sábado, en ambos años, el día de la semana en que menos discusiones se crearon, con 40 en 2019 y 57 en 2020 (36 y 42, respectivamente, teniendo en cuenta las borradas). Además, poniendo de relieve la importancia que tuvieron los foros durante el confinamiento, para solventar dudas de los alumnos y facilitar las comunicaciones entre el profesor y los alumnos, se puede apreciar cómo, salvo los días del fin de semana, el número de discusiones creadas se duplicó e incluso triplicó algunos días.

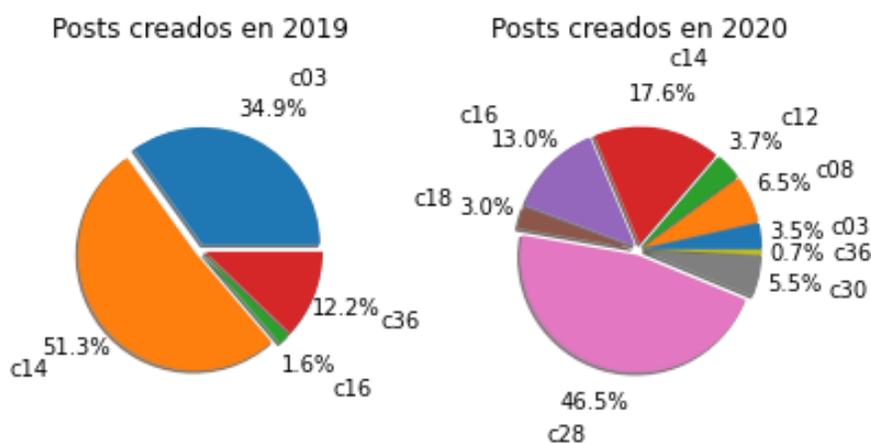


**Figura XIX.** Discusiones creadas en 2019 y 2020 por día de la semana

Para finalizar, en la figura XX se encuentran los posts creados en cada centro, tanto en 2019 como en 2020, considerando los posts como respuestas a discusiones. El hecho de que, en 2019, solo se crearan posts en cuatro facultades, a pesar de que se crearon discusiones en todas, puede indicar que, en 2019, se empleó, fundamentalmente, para realizar notificaciones a los alumnos. La Facultad de Ciencias del Trabajo, que fue la tercera en discusiones creadas (con 69 creadas y 54 netas), es la que más posts creó en 2019, con 97 posts creados y ninguno borrado, más de la mitad de ese año y más de un post por discusión, frente a los tres posts y ninguno borrado de la

Facultad de Enfermería, en la cual se crearon un total de 15 discusiones. Sorprendentemente, la Facultad de Ciencias de la Educación, que fue la que más foros y discusiones creó ese año, es la tercera en posts creados, con un total de 23 y ninguno borrado.

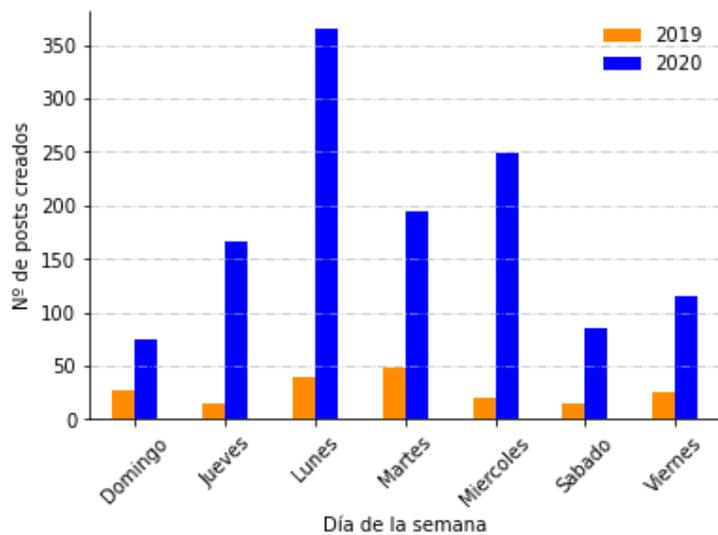
En 2020, la Facultad de Ciencias del Trabajo pasó a ser la segunda en posts creados, con 220 creados y 153 netos, cediendo el primer puesto a la Facultad de Derecho, con un total de 582 posts creados y 460 netos, casi la mitad del total, siendo, también, la que más foros y discusiones creó ese año, lo cual contrasta mucho con los 9 posts y 6 netos de la Facultad de Ciencias de la Educación, a pesar de todas las discusiones creadas en dicha facultad en ambos años.



**Figura XX.** Posts creados en 2019 y 2020 por centro

Si analizamos los datos por día de la semana (Figura XXI), podemos ver que la diferencia de un año a otro es mucho mayor en comparación con las discusiones creadas, ya que, en 2020, se llegan a generar hasta más de 10 veces el número de posts del año anterior. Mientras que, en 2019, casi la mitad de los posts se generaron en martes, con 49 creados y 46 netos, o lunes, con 40 creados y 37 netos, en 2020 casi la mitad se publicaron en lunes, con 365 creados y 286 netos, o en miércoles, con 249 creados y 197 netos. Aunque en el caso de los posts no hay tanta diferencia entre los días del fin de semana y el resto de los días, como ocurría con las discusiones, sí que se publicaron menos posts en fin de semana, incluido el viernes.

Como puede verse en las cifras obtenidas, surge el mismo problema, en cuanto a número de posts borrados, que ocurría con las discusiones.



**Figura XXI.** Posts creados en 2019 y 2020 por día de la semana

## 4.4 Recursos creados

Los códigos relativos al análisis del fichero correspondiente a los recursos creados se encuentran en el anexo XII, el cual contiene el número de recursos didácticos y actividades que se crearon en la plataforma Moodle, dichos recursos y actividades son: archivos, asistencia (herramienta para que el profesor mantenga un registro de los alumnos que asisten a las clases), bases de datos, carpetas, consultas, chats, encuestas, encuestas predefinidas, etiquetas, glosarios, herramientas externas, lecciones, libros, páginas, cuestionarios, sesiones de zoom, talleres, tareas, urls, videos y wikis. El fichero contiene un total de 1.523 registros agrupados, al igual que los datos del fichero del apartado anterior, por usuario, curso, centro, hora (en tramos de una hora, sin minutos), fecha y semana (número de semana del año).

Es necesario aclarar que, aunque se han incluido las carpetas y etiquetas en este fichero, son elementos, puestos a disposición del profesor, para organizar mejor la página de cada asignatura en la Moodle, sin embargo, como veremos en la tabla resumen a continuación, se crearon un gran número de etiquetas y carpetas en 2020, bastantes más que el año anterior, lo que puede indicar que los profesores, dada la situación, decidieran organizar mejor las páginas de Moodle de sus asignaturas. Además, los recursos “Paquete SCORM” y “Paquete de contenido IMS” no han sido incluidos ya que no se han encontrado registros relacionados con su uso.

En la tabla XX se muestran los totales de los recursos y actividades que se crearon en 2019 y 2020, donde puede verse que, mientras algunos recursos no eran empleados en 2019 en los cursos analizados, como los chats, las encuestas, las encuestas predefinidas, los glosarios, las lecciones, las sesiones de zoom (recurso que se incorporó a la plataforma en 2020 para facilitar la docencia online), los talleres y los videos (que también se incorporaron en 2020), se llegaron a emplear todos los recursos existentes durante 2020, e incluso algunos como los chats y las sesiones de zoom, que no se utilizaron en 2019, llegaron a tener bastante peso entre el total de 2020, especialmente las sesiones de zoom.

En esta tabla destaca especialmente el número de archivos creados, con un total de 398 en 2019 y 658 en 2020, cifras a las que solo se acercan las sesiones de zoom, que, aunque no se utilizaron en 2019, se crearon un total de 351 sesiones en 2020. En cuanto a las actividades para la evaluación de los alumnos, parece ser que las más empleadas por los docentes, fueron las tareas, con 45 en 2019 y 168 en 2020, y los cuestionarios, que pasaron de 13 en 2019 a 214 en 2020.

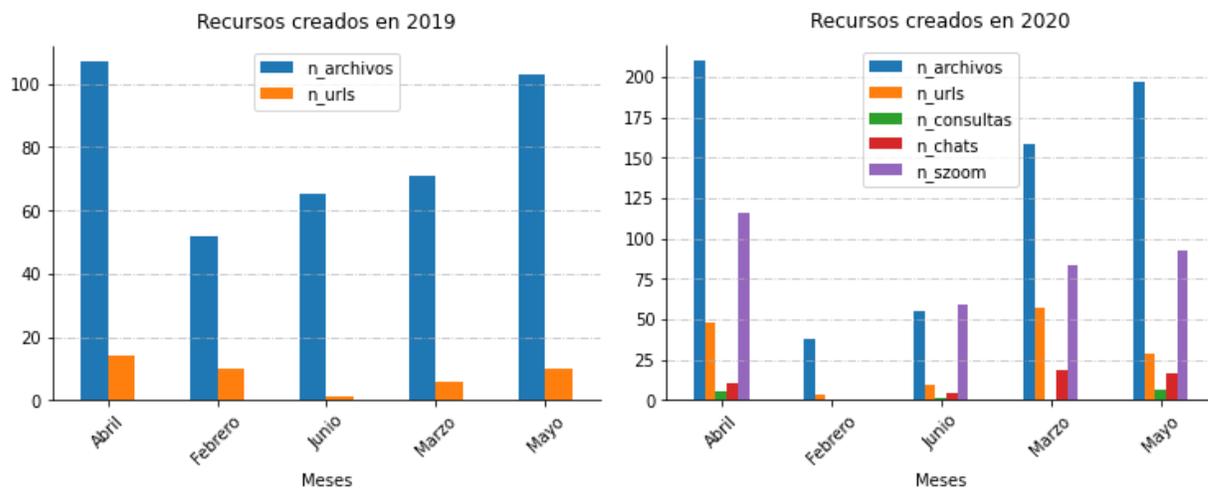
**Tabla XX.** Recursos y actividades totales creados en 2019 y 2020

	<b>2019</b>	<b>2020</b>	<b>Total</b>
n_archivos	398	658	1056
n_asistencia	4	6	10
n_bd	1	2	3
n_carpetas	32	76	108
n_consultas	4	12	16
n_chats	0	48	48
n_encuestas	0	6	6
n_encuestas_pred	0	1	1
n_etiquetas	37	174	211
n_glosarios	0	2	2
n_herr_ext	2	7	9
n_lecciones	0	4	4
n_libros	2	1	3
n_paginas	4	8	12
n_cuestionarios	13	214	227
n_szoom	0	351	351
n_talleres	0	6	6
n_tareas	45	168	213
n_urls	41	146	187
n_videos	0	8	8
n_wikis	1	1	2

Durante el análisis de este fichero, que realizaremos a continuación, se considerarán actividades evaluables las siguientes: bases de datos, cuestionarios, glosarios, lecciones, talleres, tareas, videos y wikis. Se excluyen los chats, las consultas, las encuestas y encuestas predefinidas, los foros y las herramientas externas por considerar que, aunque puedan tener un carácter evaluativo y aparezcan como actividades dentro de la plataforma, no suelen usarse con esa función salvo casos aislados. Las lecciones se han incluido como actividad evaluable ya que, aunque se emplean para exponer materia de las asignaturas, al final de cada sección se suele incorporar un breve cuestionario evaluable sobre el tema expuesto.

En la figura XXII, compuesta por los principales recursos creados en 2019 y 2020, podemos ver que los recursos que destacan sobre el resto, durante todos los meses, son los archivos, que solo se ven superados por las sesiones de Zoom en junio de 2020, mes durante el cual, probablemente, se emplearían para realizar los exámenes online. A pesar de que se han graficado, exclusivamente, aquellos recursos que se hubieran empleado más de una decena de veces a lo largo del curso, si lo desglosamos por meses, tal es el número de veces que se emplearon los archivos y las sesiones de Zoom, que prácticamente hacen imperceptibles gráficamente al resto de recursos, tan solo se acerca el número de urls en 2020. Se puede observar cómo las sesiones de Zoom se volvieron un recurso muy necesario y empleado durante 2020.

En las gráficas también podemos apreciar que el número de archivos que se subían a la plataforma, durante 2019, se iban incrementando hasta abril y luego descendían nuevamente, siendo abril y mayo los meses en que más archivos se publicaron, con más de 100 archivos en ambos meses. Si observamos la de 2020, se puede apreciar cómo, durante los principales meses de docencia (marzo, abril y mayo), el número de archivos subidos a la plataforma se duplicaron respecto al anterior, consecuencia directa del confinamiento decretado en España, mientras que junio se mantiene y febrero tuvo la mitad de archivos que el año anterior. Llama la atención que, mientras en 2019 la diferencia de archivos subidos en marzo respecto a febrero no era muy grande, en 2020 la diferencia es de hasta cuatro veces entre ambos meses y aún más respecto a abril.



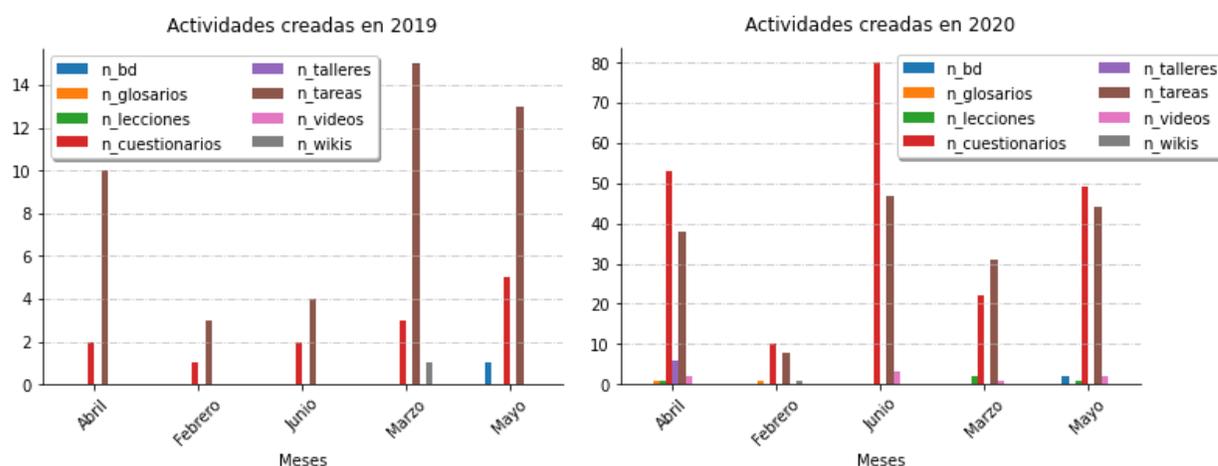
**Figura XXII.** Principales recursos creados en 2019 y 2020 por meses

En la tabla XXI se encuentra, de forma complementaria a la gráfica anterior, el número de recursos empleados por los docentes, agrupado por año y centro.

**Tabla XXI.** Recursos creados por centro y año

fecha	centro	n_archivos	n_asistencia	n_carpetas	n_consultas	n_chats	n_encuestas	n_encuestas_pred	n_etiquetas	n_herr_ext	n_libros	n_paginas	n_szoom	n_urls
2019	c03	122	1	1	0	0	0	0	19	0	0	0	0	18
	c08	54	0	3	0	0	0	0	0	0	0	2	0	0
	c12	25	0	0	0	0	0	0	0	0	0	0	0	0
	c14	39	0	7	3	0	0	0	3	1	1	0	0	12
	c16	70	1	19	0	0	0	0	4	0	1	1	0	1
	c18	11	2	0	1	0	0	0	1	0	0	0	0	1
	c28	0	0	1	0	0	0	0	0	0	0	0	0	2
	c30	16	0	0	0	0	0	0	0	1	0	0	0	0
	c36	61	0	1	0	0	0	0	10	0	0	1	0	7
2020	c03	183	0	10	3	5	0	0	21	1	0	0	50	24
	c08	53	0	5	0	8	0	0	1	0	0	2	40	1
	c12	41	0	0	0	9	0	0	3	0	0	1	40	5
	c14	29	1	0	1	1	2	0	19	1	0	0	55	6
	c16	59	0	9	0	0	0	0	39	0	1	3	35	25
	c18	13	1	1	3	1	0	0	7	1	0	0	71	6
	c28	159	3	23	3	23	0	0	0	0	0	2	26	12
	c30	45	1	13	1	0	2	0	46	2	0	0	26	1
	c36	76	0	15	1	1	2	1	38	2	0	0	8	66

En la figura XXIII tenemos las actividades evaluables empleadas en la plataforma, siendo los cuestionarios y las tareas las más empleadas por los docentes en ambos años. Destaca el hecho de que, mientras en 2019 prácticamente solo se empleaban tareas para evaluar a los alumnos dentro de la plataforma, en 2020, con la docencia online, las actividades más empleadas para evaluar a los alumnos pasaron a ser los cuestionarios, casi a la par que el número de tareas, salvo en el mes de junio, en el que el número de cuestionarios prácticamente dobla el de tareas, debido a la realización de los exámenes de forma online.



**Figura XXIII.** Actividades creadas en 2019 y 2020 por meses

En la tabla XXII, se encuentra, de forma complementaria a la figura anterior, el número de actividades creadas por cada centro y cada año, en la que destaca que, mientras todas las facultades empleaban, en 2020, principalmente cuestionarios y tareas, de forma conjunta, para evaluar a los alumnos, la Facultad de Ciencias Experimentales tan solo empleó tareas, ningún otro recurso, y la Facultad de Ciencias de la Educación, por el contrario, no empleó ni una sola tarea, sino cuestionarios, acompañados de un glosario y dos lecciones.

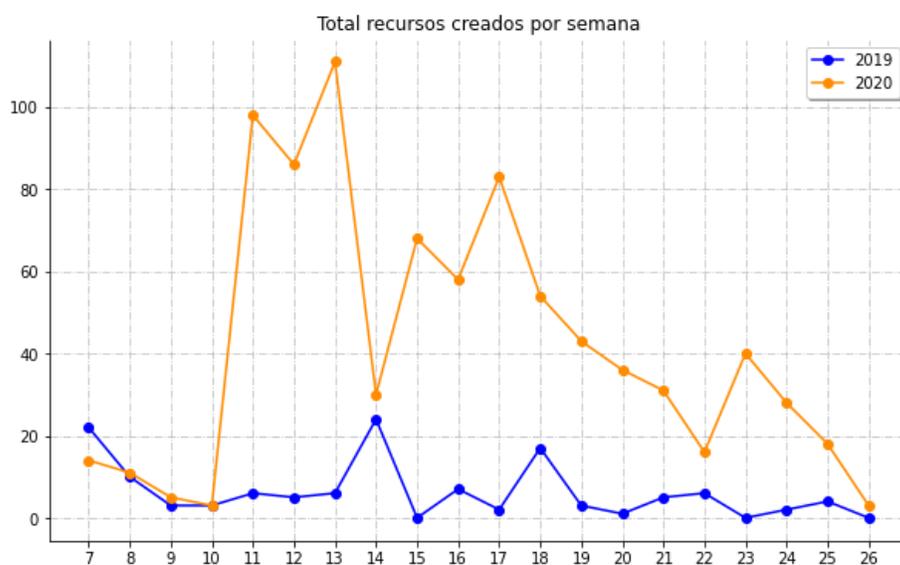
**Tabla XXII.** Actividades creadas por centro y año

fecha	centro	n_bd	n_glosarios	n_lecciones	n_cuestionarios	n_talleres	n_tareas	n_videos	n_wikies
2019	c03	1	0	0	0	0	0	0	0
	c08	0	0	0	0	0	10	0	0
	c12	0	0	0	0	0	2	0	0
	c14	0	0	0	3	0	9	0	0
	c16	0	0	0	3	0	6	0	1
	c18	0	0	0	0	0	15	0	0
	c28	0	0	0	0	0	0	0	0
	c30	0	0	0	7	0	2	0	0
	c36	0	0	0	0	0	1	0	0
2020	c03	0	0	0	0	0	44	0	0
	c08	0	0	0	11	0	21	3	0
	c12	0	0	0	13	0	8	0	0
	c14	0	0	0	21	0	12	2	0
	c16	0	1	0	31	1	17	0	1
	c18	2	0	2	16	5	20	1	0
	c28	0	0	0	41	0	19	0	0
	c30	0	0	0	42	0	27	2	0
	c36	0	1	2	39	0	0	0	0

Para finalizar, tras haber visto en qué meses se crearon en la plataforma el mayor número de recursos y actividades, vamos a analizar, por separado, la evolución temporal que siguió dicho número a lo largo del curso.

En primer lugar, en la figura XXIV, tenemos el total de recursos creados por semana para ambos años. Para el año 2019, podemos ver cómo el número de recursos publicados fue bastante reducido, con dos picos por encima de los 20 recursos, en la semana del 11 al 17 de febrero y en

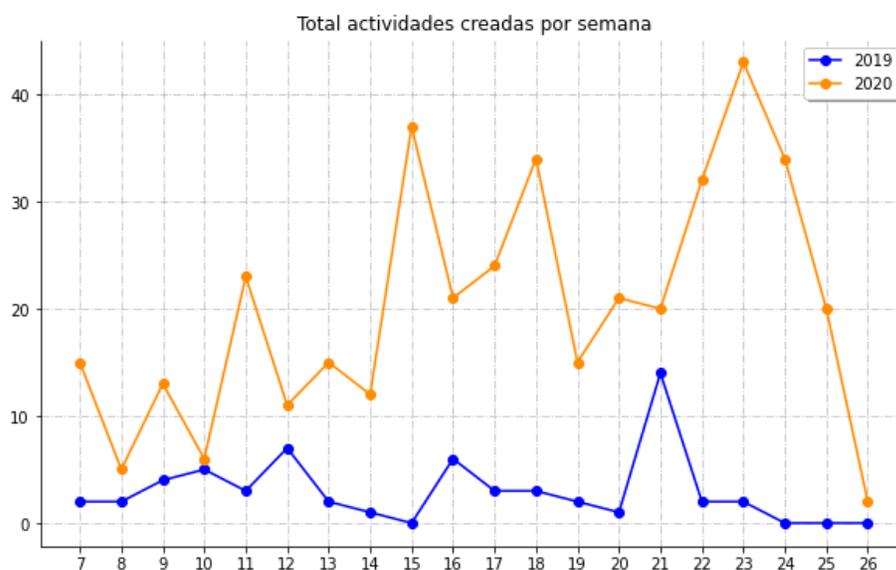
la del 1 al 7 de abril, seguidos por la semana del 29 de abril al 5 de mayo, con poco menos de 20 recursos creados en la plataforma. Sin embargo, la evolución durante 2020 fue mucho más inestable, ya que, aunque empieza descendiendo, en la semana del 9 al 15 de marzo, semana en la que se declara el Estado de Alarma en España, el número de recursos creados crece desde los casi cero de la semana anterior hasta los 100 recursos en una semana. La semana siguiente cayó ligeramente y, la semana del 23 al 29 de marzo, un día después de que se anunciara la primera prórroga del Estado de Alarma, volvió a aumentar por encima de los 100 recursos creados en una semana. A partir de entonces, salvo dos caídas bruscas en las semanas del 30 de marzo al 5 de abril y del 25 al 31 de mayo, con sus posteriores recuperaciones, comenzó a descender el número de recursos creados semana a semana. Curiosamente, el último día de las semanas correspondientes a esas dos caídas bruscas coinciden con el anuncio de la segunda prórroga y del cese de la desescalada a las autonomías, lo que seguramente provocaron los incrementos de los recursos creados en los días posteriores.



**Figura XXIV.** Total de recursos creados por semana en 2019 y 2020

En cuanto a las actividades creadas (Figura XXV), observando la evolución temporal del año 2019 se puede apreciar cómo los picos se generan en momentos muy diferentes a los picos de los recursos de ese mismo año, pero manteniendo la estabilidad el resto del año, mientras que sí se mueve de manera muy similar a la de las actividades de 2020, produciéndose los principales picos de 2019 en las semanas del 18 al 24 de marzo, del 15 al 21 de abril y del 20 al 26 de mayo. Lo cual puede significar que, mientras en 2019 los profesores preparaban las actividades con antelación, en 2020 se fueron adaptando conforme evolucionaba la situación.

Al contrario de lo que sucede con la evolución temporal de 2019, aunque la de 2020 pueda parecer más volátil, tiene los mismos picos que la gráfica anterior, la de los recursos creados, salvo el pico que en la gráfica anterior se producía en la semana 17 (20 a 26 de abril), que en esta se produce una semana después, por lo que los anuncios más trascendentales acerca de la pandemia fueron los que marcaron, sin duda alguna, la actividad que se desarrollaba dentro de la plataforma. Sin embargo, el principal pico en la evolución temporal de las actividades creadas durante 2020 se produjo en la semana del 3 al 9 de junio, aunque las dos semanas posteriores tampoco se quedaron atrás en cuanto a recursos creados, debido a la obligación de tener que realizar los exámenes de forma online.



**Figura XXV.** Total de recursos creados por semana en 2019 y 2020

## 4.5 Participación

Los códigos relativos al análisis del fichero correspondiente a los recursos creados se encuentran en el anexo XIII, el cual contiene las siguientes cantidades: accesos a sesiones de Zoom, asistencias registradas, accesos a chats, participaciones en chats, accesos a glosarios, participaciones en glosarios, envíos de archivos a talleres, envíos de archivos a tareas, videos subidos y consultas, encuestas y cuestionarios respondidos.

El fichero contiene un total de 10.829 registros agrupados, al igual que los datos de ficheros anteriores, por curso, centro, hora (en tramos de una hora, sin minutos), fecha y semana (número de semana del año), excluyendo, en esta ocasión, las ID de los usuarios.

En un primer vistazo al fichero de participación con la tabla XXIII, sorprende el elevado número de accesos y participaciones en chats, con 6.305 accesos y 13.046 mensajes enviados, recurso del que se crearon un total de 48 salas en 2020. Fueron la Facultad de Ciencias Empresariales y la Facultad de Enfermería las únicas que no crearon ninguna sala de chat, todas las demás crearon al menos una, mientras que la Facultad de Derecho creó un total de 23 salas, casi la mitad del total.

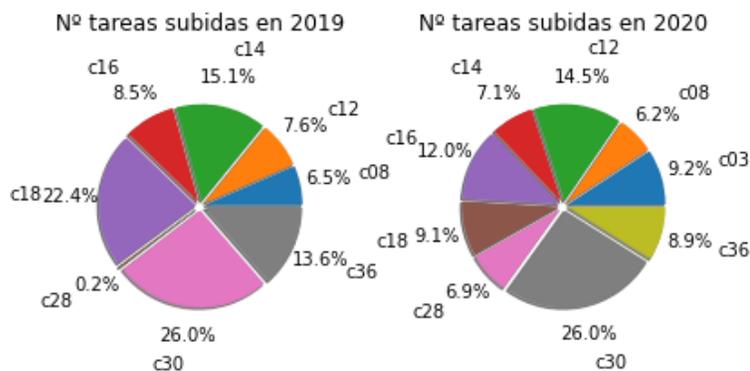
También destaca la gran diferencia que hay entre participaciones en glosarios y accesos a los mismos, con 148 participaciones frente a 1.501 accesos en 2020, de los que se crearon tan solo 2 en 2020, uno en la Facultad de Enfermería y otro en la Facultad de Ciencias de la Educación, lo que puede significar que los alumnos accedieron habitualmente a este recurso para obtener información acerca de algún término o tema de sus asignaturas.

Sin embargo, aunque no se crearon glosarios ni chats en 2019, sí que aparecen registros de que fueran empleados durante ese año.

**Tabla XXIII.** Resumen del fichero de participación

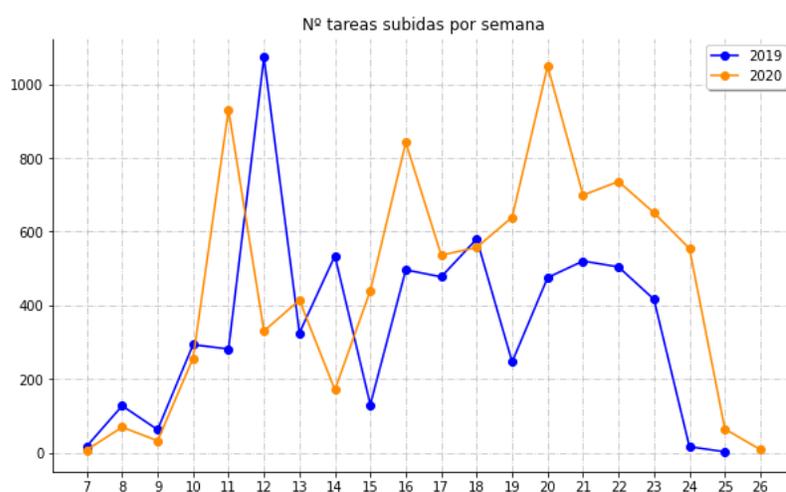
	<b>2019</b>	<b>2020</b>	<b>Total</b>
nar	267	190	457
nac	53	6035	6088
npc	28	13046	13074
nag	19	1501	1520
npg	0	148	148
ntals	0	195	195
ntars	6574	8984	15558
nvs	0	79	79
ncor	0	825	825
ner	0	129	129
ncur	6676	17115	23791

En cuanto al número de tareas subidas por centro y año (Figura XXVI) podemos ver que claramente fue la Facultad de Ciencias Empresariales la que más tareas subió a la plataforma, con un total de 1710 en 2019 y 2334 en 2020, seguida muy de cerca, en 2019, por la ETSI, con 1474 tareas subidas ese año, aunque en 2020 subió la mitad de tareas que en 2019. El hecho de que la ETSI esté tan cerca de la Facultad de Ciencias Empresariales llama aún más la atención si además tenemos en cuenta que, entre las tres asignaturas analizadas de ambas facultades, la ETSI suma en torno a la mitad de usuarios que la Facultad de Ciencias Empresariales. Por el contrario, en la Facultad de Ciencias Experimentales no se subió ninguna tarea y en la Facultad de Derecho se subieron tan solo 14 tareas, estas dos facultades fueron las que tuvieron un mayor aumento en 2020, con 828 y 623 tareas subidas respectivamente, en cuanto a la Facultad de Ciencias del Trabajo, la ETSI y la Facultad de Ciencias de la Educación, subieron menos tareas en 2020 que en 2019. Mientras que en 2019 dos facultades acumulaban casi la mitad de las tareas subidas a la plataforma, en 2020 el gráfico está mucho más repartido entre todas las facultades.



**Figura XXVI.** Número de tareas subidas por centro y año

Si analizamos como evolucionó el número de tareas subidas por semana en la figura XXVII, podemos ver que, mientras en 2019 se produce un único pico en la semana del 18 al 27 de marzo (el cual es la máxima cantidad de tareas subidas en ambos años), siendo relativamente estable el resto del año, en 2020 se producen tres picos, en las semanas del 9 al 15 de marzo, del 13 al 19 de abril y del 11 al 17 de mayo, coincidiendo, tan solo, el primer pico de ambos años con el de las actividades creadas, mientras que el segundo pico de tareas subidas se produce una semana después del de actividades creadas.

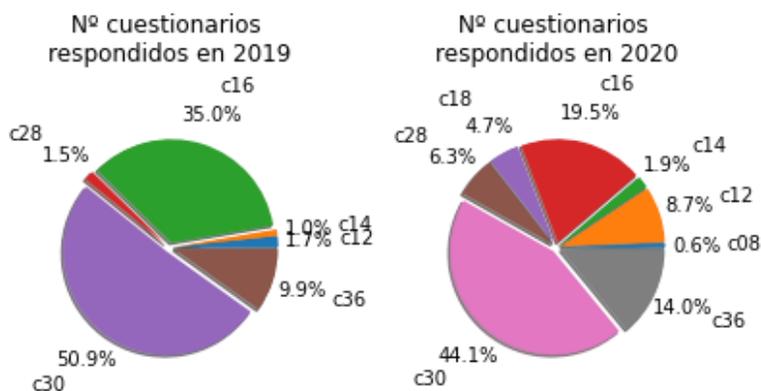


**Figura XXVII.** Número de tareas subidas por semana y año

Respecto a los cuestionarios respondidos (Figura XXVIII), de nuevo destaca la Facultad de Ciencias Empresariales, acumulando la mitad de los cuestionarios respondidos en 2019 y algo menos en 2020, suponiendo el doble en cantidad de cuestionarios respondidos en 2020 respecto a 2019. Aunque dicha facultad fue la que más cuestionarios creó en 2019 (un total de 7, frente a 3 de la Facultad de Ciencias del Trabajo y otros 3 de la Facultad de Enfermería), en 2020 creó uno más que la Facultad de Derecho, la cual creó 41 cuestionarios, y, aunque este hecho pueda parecer que esté influido por el número de alumnos, aun teniendo la mitad de alumnos que la Facultad de Ciencias Empresariales, está en torno a la media en número de usuarios.

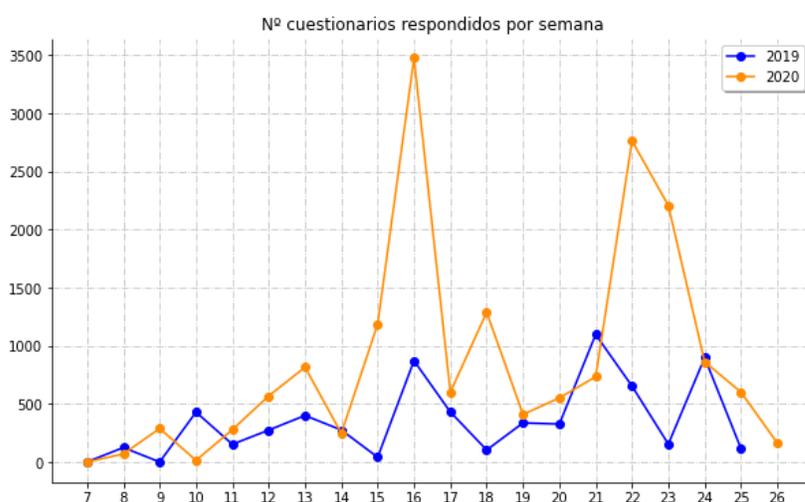
En 2019, los alumnos de la Facultad de Ciencias Experimentales, la Facultad de Humanidades y la ETSI no respondieron a ningún cuestionario, mientras que fueron los usuarios de la Facultad de Ciencias Experimentales los únicos en no responder cuestionarios en 2020. Aparentemente, comparando el número de cuestionarios creados y respondidos, en 2019 la mayoría de las facultades optaban por reutilizar los cuestionarios de años anteriores.

Los aumentos más significativos, de 2020 respecto a 2019, del número de cuestionarios respondidos, los tuvieron los usuarios de la Facultad de Trabajo Social, que pasó de 113 a 1495, y la ETSI, que pasó de 0 a 810.



**Figura XXVIII.** Número de cuestionarios respondidos por centro y año

En cuanto a la evolución temporal del número de cuestionarios respondidos (Figura XXIX), mientras que en 2019 se mantiene muy estable a lo largo del año, con pequeños picos en las semanas del 15 al 21 de abril, del 20 al 26 de mayo y del 10 al 16 de junio, en 2020 se producen dos grandes picos en las semanas del 13 al 19 de abril y del 25 al 31 de mayo, que, al igual que sucedía con las tareas subidas, tampoco se corresponden con los picos de actividades creadas, aunque el primer pico de 2020 se produce una semana después del segundo pico de actividades creadas.



**Figura XXIX.** Número de cuestionarios respondidos por semana y año

De igual modo, es necesario reflexionar acerca de la interacción real de los estudiantes con la plataforma. Como hemos visto hasta ahora la mayoría de las actividades empleadas, que a fin de cuentas son las que llevan al alumno a interactuar con el aula virtual, se centran en cuestionarios y tareas y, de estos, el cuestionario es el único los dos que hace al alumno interactuar con la plataforma de forma activa, mientras que la tarea le hace interactuar con su entorno académico. Lógicamente, es muy necesario que el alumno interactúe con su entorno, sin embargo, si no se le lleva a interactuar con la plataforma, con herramientas activas como pueden ser las wikis o los glosarios, difícilmente se podrá medir el desempeño que hace el alumno y el tiempo que dedica a aprender, mejorando así su rendimiento académico.

## 4.6 Análisis cluster

El análisis cluster es un método de clasificación automática de los datos, es decir, aprendizaje no supervisado, en el que se buscan grupos homogéneos. Durante este apartado se emplearán, y compararán, dos técnicas de clustering, por un lado, el método de K-means, que es un método no jerárquico o de partición por el que se asigna cada observación al grupo con el que guarda mayor similitud, y, por otro lado, el método de Ward, que es un método jerárquico por el que se va buscando formar grupos con la menor varianza posible.

Con este análisis se busca identificar aquellas facultades que tuvieron un comportamiento similar en 2020, para ello, se tomaron los ficheros respectivos a otras acciones, foros, recursos creados y participación, agrupando las observaciones de dicho año por facultad.

Todos los códigos empleados para el análisis cluster, mediante Stata, se encuentran en el anexo XIV.

### 4.6.1 Otras acciones

En primer lugar, comprobaremos que no exista correlación entre las variables. Como puede verse en la tabla XXIV, el total de acciones parece que guarda cierta correlación con el número de inicios de sesión y las visualizaciones de las calificaciones, por lo que vamos a eliminarla a ver si desaparece esa correlación.

**Tabla XXIV.** Tabla de correlaciones de otras acciones (1/2)

	nis	t_acc	vistas~f	num_me~s
nis	1.0000			
t_acc	0.9070	1.0000		
vistas_calif	0.4770	0.7573	1.0000	
num_mensaj~s	0.4296	0.4339	0.3856	1.0000

Si eliminamos el total de acciones, parece que desaparece dicha correlación (Tabla XXV).

**Tabla XXV.** Tabla de correlaciones de otras acciones (2/2)

	nis vistas~f num_me~s		
nis	1.0000		
vistas_calif	0.4770	1.0000	
num_mensaj~s	0.4296	0.3856	1.0000

Tras realizar el análisis estadístico pertinente (Tabla XXVI), estableciendo la regla de duda, el método de Ward nos lleva a quedarnos con 7 grupos, algo sorprendente, ya que eso supondría que prácticamente cada facultad ha tenido un comportamiento diferente.

**Tabla XXVI.** Análisis estadístico para la selección del número de clusters (otras acciones)

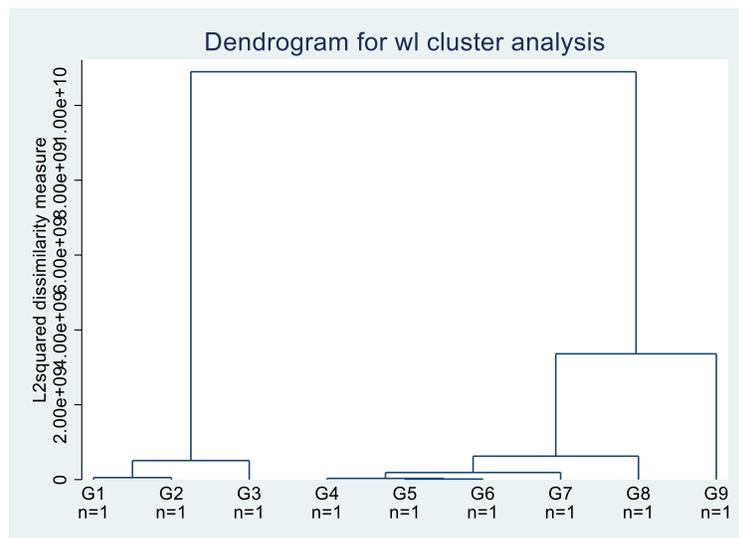
Number of clusters	Duda/Hart	
	Je(2)/Je(1)	pseudo T-squared
1	0.3052	15.94
2	0.2043	15.57
3	0.2749	7.91
4	0.0955	9.47
5	0.1995	8.02
6	0.0000	.
7	0.3182	2.14

Tras generar los clusters, obtenemos la tabla XXVII, en la que podemos ver que tan solo la Facultad de Trabajo Social, la Facultad de Derecho y la Facultad de Ciencias de la Educación habrían tenido un comportamiento similar (grupo 4), asignando al resto a diferentes grupos.

**Tabla XXVII.** Clusters para otras acciones

n_centro	cluster_w1							Total
	1	2	3	4	5	6	7	
c03	1	0	0	0	0	0	0	1
c08	0	1	0	0	0	0	0	1
c12	0	0	0	1	0	0	0	1
c14	0	0	1	0	0	0	0	1
c16	0	0	0	0	1	0	0	1
c18	0	0	0	0	0	1	0	1
c28	0	0	0	1	0	0	0	1
c30	0	0	0	0	0	0	1	1
c36	0	0	0	1	0	0	0	1
Total	1	1	1	3	1	1	1	9

Si observamos el dendrograma generado (Figura XXX) podemos ver que, aunque ha generado un total de 7 grupos, hay facultades con muy baja distancia euclídea entre ellas.



**Figura XXX.** Dendrograma de otras acciones

Si observamos la media de las variables para cada grupo (Tabla XXVIII), podemos ver que el grupo 7 (Facultad de Ciencias Empresariales) es el que tiene mayor de número de inicios de sesión y de mensajes enviados, el grupo 5 (Facultad de Enfermería) es el que tiene mayor número de visualización de calificaciones, el grupo 3 es el que presenta mayor número de inicios de sesión. El grupo 4 sería uno de los más cercanos a la media total en número de inicios de sesión y visualización de calificaciones y de los que tendría menor número de mensajes enviados.

**Tabla XXVIII.** Medias de las variables para cada grupo (otras acciones)

cluster_wl	nis	vistas~f	num_me~s
1	35992	1470	6493
2	39839	938	273
3	18405	209	3062
4	77829.67	1475	1541.333
5	86895	4197	7627
6	60469	1236	5332
7	120674	1842	9522
Total	66195.89	1590.778	4103.667

Por último, en la tabla XXIX podemos ver que, tanto por el método de Ward como por el de K-means, las asignaciones son las mismas.

**Tabla XXIX.** Comparación de K-means y Ward (otras acciones)

cluster_wl	kmeans							Total
	1	2	3	4	5	6	7	
1	1	0	0	0	0	0	0	1
2	0	1	0	0	0	0	0	1
3	0	0	1	0	0	0	0	1
4	0	0	0	3	0	0	0	3
5	0	0	0	0	1	0	0	1
6	0	0	0	0	0	1	0	1
7	0	0	0	0	0	0	1	1
Total	1	1	1	3	1	1	1	9

Si decidieramos quedarnos con solo 3 grupos (Tabla XXX), podemos ver que asignaría al grupo 1 a la Facultad de Ciencias Experimentales, la Facultad de Humanidades y a la Facultad de Ciencias del Trabajo, al grupo 3 asignaría únicamente a la Facultad de Ciencias Empresariales y al grupo 2 asignaría a las restantes, es decir, a la Facultad de Trabajo Social, la Facultad de Enfermería, la ETSI, la Facultad de Derecho y a la Facultad de Ciencias de la Educación.

**Tabla XXX.** Clusters alternativos para otras acciones

n_centro	cluster_w12			Total
	1	2	3	
c03	1	0	0	1
c08	1	0	0	1
c12	0	1	0	1
c14	1	0	0	1
c16	0	1	0	1
c18	0	1	0	1
c28	0	1	0	1
c30	0	0	1	1
c36	0	1	0	1
Total	3	5	1	9

En cuanto a las medias de las variables (Tabla XXXI), vemos que el grupo 2 sería el que tendría mayor número de inicios de sesión y de visualizaciones de calificaciones, el 3 el que tendría mayor número de mensajes y enviados y el grupo 1 el que tendría los datos más bajos.

**Tabla XXXI.** Medias alternativas de las variables para cada grupo (otras acciones)

cluster_w12	nis	vistas~f	num_me~s
1	31412	872.3333	3276
2	76170.6	1971.6	3516.6
3	120674	1842	9522
Total	66195.89	1590.778	4103.667

## 4.6.2 Foros

Para el análisis cluster de este fichero se van a emplear, exclusivamente, los foros, discusiones y posts creados netos, es decir, creados menos borrados, ya que los otros 3 datos están directamente relacionados con estos. Tras analizar la correlación (Tabla XXXII), se observa que no existe correlación entre las variables.

**Tabla XXXII.** Tabla de correlaciones de foros, discusiones y posts creados netos

	foros_~o	discus~o	posts_~o
foros_crea~o	1.0000		
discusione~o	-0.2015	1.0000	
posts_crea~o	0.4619	0.2830	1.0000

Tras aplicarle el correspondiente análisis estadístico (Tabla XXXIII), con regla de parada de duda, para determinar el número de cluster, nos quedamos con 2 grupos.

**Tabla XXXIII.** Análisis estadístico para la selección del número de clusters (foros)

Number of clusters	Duda/Hart	
	Je(2)/Je(1)	pseudo T-squared
1	0.2201	24.80
2	0.4211	8.25
3	0.2513	8.94
4	0.0301	32.26
5	0.1225	7.16
6	0.0000	.
7	0.0000	.

Una vez generados los clusters, podemos ver en la tabla XXXIV que el método de Ward asigna a todas las facultades al mismo grupo, excepto a la Facultad de Derecho que la asigna a un cluster separado.

**Tabla XXXIV.** Clusters alternativos para foros

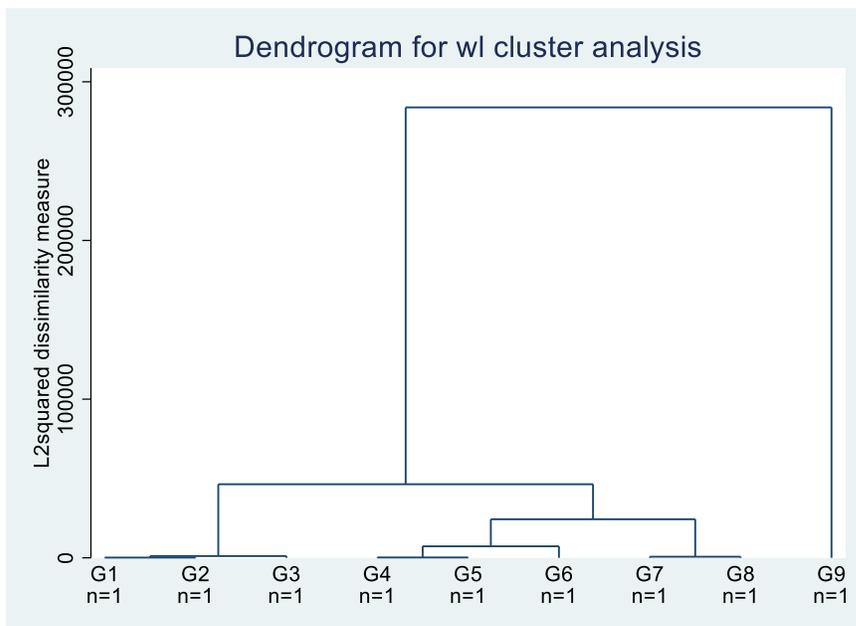
n_centro	cluster_wl		Total
	1	2	
c03	1	0	1
c08	1	0	1
c12	1	0	1
c14	1	0	1
c16	1	0	1
c18	1	0	1
c28	0	1	1
c30	1	0	1
c36	1	0	1
Total	8	1	9

Respecto a las medias de las variables (Tabla XXXV), podemos ver que claramente el grupo 2, al que solo pertenece la Facultad de Derecho, es la que tiene mayor media en todas las variables.

**Tabla XXXV.** Medias de las variables para cada grupo (foros)

cluster_wl	foros_~o	discus~o	posts_~o
1	5.5	93.125	61
2	12	114	460
Total	6.222222	95.44444	105.3333

En el dendrograma generado (Figura XXXI) podemos ver claramente que todas las facultades, salvo una, tienen una distancia euclídea muy pequeña, aunque hay ciertos subgrupos con todavía menos distancia entre sí.



**Figura XXXI.** Dendrograma de foros

Por último, en la tabla XXXVI, podemos ver que por el método de K-means las facultades se asignan a los mismos grupos que con el método de Ward.

**Tabla XXXVI.** Comparación de K-means y Ward (foros)

cluster_w1	kmeans		Total
	1	2	
1	8	0	8
2	0	1	1
Total	8	1	9

### 4.6.3 Recursos creados

Para este fichero, al igual que se hizo en su correspondiente análisis, se va a dividir el clustering en actividades y recursos (excluyendo etiquetas y carpetas)

Observando la tabla de correlaciones de los recursos (Tabla XXXVII), parece que existe una elevada correlación entre algunos recursos, por lo que vamos a eliminar las herramientas externas y las encuestas predefinidas a ver si desaparecen las correlaciones.

**Tabla XXXVII.** Tabla de correlaciones de los recursos (1/2)

	n_archivos	n_asistencia	n_consultas	n_chats	n_encuestas	n_encuesta~d	n_herr_ext
n_archivos	1.0000						
n_asistencia	0.2675	1.0000					
n_consultas	0.5447	0.5669	1.0000				
n_chats	0.5641	0.6529	0.3164	1.0000			
n_encuestas	-0.2959	0.0000	-0.1890	-0.4688	1.0000		
n_encuesta~d	0.0185	-0.2500	-0.0945	-0.2176	0.5000	1.0000	
n_herr_ext	-0.1147	-0.1000	0.1890	-0.5692	0.8000	0.5500	1.0000
n_libros	-0.0903	-0.2500	-0.3780	-0.2679	-0.2500	-0.1250	-0.3500
n_paginas	0.1045	0.0714	-0.3780	0.4066	-0.5714	-0.2857	-0.8000
n_szoom	-0.2613	-0.0271	0.3178	-0.1907	-0.3798	-0.6307	-0.2360
n_urls	0.2753	-0.2917	0.0198	-0.2034	0.2937	0.9013	0.4088

	n_libros	n_paginas	n_szoom	n_urls
n_libros	1.0000			
n_paginas	0.6786	1.0000		
n_szoom	-0.0814	-0.2034	1.0000	
n_urls	0.1589	-0.1230	-0.5753	1.0000

Tras eliminar las encuestas predefinidas y las herramientas externas ya no quedan recursos correlacionados (Tabla XXXVIII)

**Tabla XXXVIII.** Tabla de correlaciones de los recursos (2/2)

	n_archivos	n_asistencia	n_consultas	n_chats	n_encuestas	n_libros	n_paginas
n_archivos	1.0000						
n_asistencia	0.2675	1.0000					
n_consultas	0.5447	0.5669	1.0000				
n_chats	0.5641	0.6529	0.3164	1.0000			
n_encuestas	-0.2959	0.0000	-0.1890	-0.4688	1.0000		
n_libros	-0.0903	-0.2500	-0.3780	-0.2679	-0.2500	1.0000	
n_paginas	0.1045	0.0714	-0.3780	0.4066	-0.5714	0.6786	1.0000
n_szoom	-0.2613	-0.0271	0.3178	-0.1907	-0.3798	-0.0814	-0.2034
n_urls	0.2753	-0.2917	0.0198	-0.2034	0.2937	0.1589	-0.1230
		n_szoom	n_urls				
n_szoom		1.0000					
n_urls		-0.5753	1.0000				

El análisis estadístico con la regla de parada de duda (Tabla XXXIX) nos lleva a quedarnos con dos grupos en lo que respecta a recursos creados.

**Tabla XXXIX.** Análisis estadístico para la selección del número de clusters (recursos)

Number of clusters	Duda/Hart	
	$J_e(2)/J_e(1)$	pseudo T-squared
1	0.2708	18.85
2	0.3807	8.13
3	0.3325	8.03
4	0.0000	.
5	0.3363	3.95
6	0.0000	.
7	0.2993	2.34

En la tabla XL podemos ver que el método de Ward asigna a la Facultad de Ciencias Experimentales y a la Facultad de Derecho al grupo 1 y al resto de facultades al grupo 2.

**Tabla XL** Clusters para recursos creados

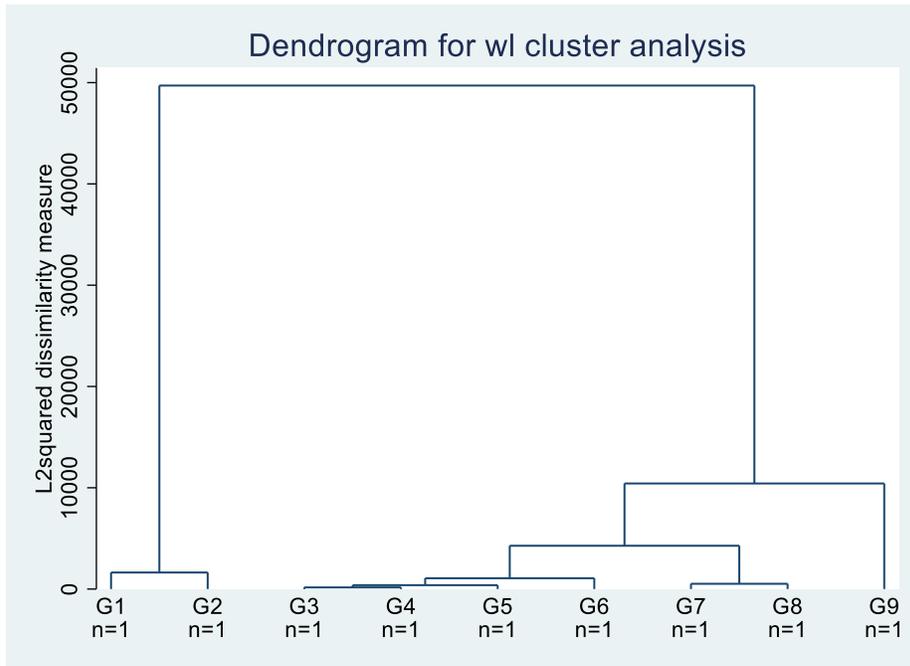
n_centro	cluster_w1		Total
	1	2	
c03	1	0	1
c08	0	1	1
c12	0	1	1
c14	0	1	1
c16	0	1	1
c18	0	1	1
c28	1	0	1
c30	0	1	1
c36	0	1	1
Total	2	7	9

En cuanto a la media de las variables de recursos creados, en la tabla XLI podemos ver que el grupo 1 es el que tiene mayor media en archivos, asistencia, consultas, chats, páginas y urls, mientras que el grupo 2 tiene mayor media en el resto (encuestas y libros), salvo en el número de sesiones de zoom, que, aunque la media del grupo 2 es mayor, están muy igualados ambos grupos.

**Tabla XLI.** Medias de las variables para cada grupo (recursos)

cluster_w1	n_arch~s	n_asis~a	n_cons~s	n_chats	n_encu~s	n_libros	n_pagi~s	n_szoom	n_urls
1	171	1.5	3	14	0	0	1	38	18
2	45.14286	.4285714	.8571429	2.857143	.8571429	.1428571	.8571429	39.28571	15.71429
Total	73.11111	.6666667	1.333333	5.333333	.6666667	.1111111	.8888889	39	16.22222

En el dendrograma (Figura XXXII) podemos ver claramente los dos grupos que ha formado, aunque la distancia euclídea en algunos subgrupos es muy reducida.



**Figura XXXII.** Dendrograma de recursos creados

Si comparamos los resultados obtenidos por el método de K-means y de Ward, podemos ver en la tabla XLII que las asignaciones a grupos son las mismas por ambos métodos

**Tabla XLII.** Comparación de K-means y Ward (recursos creados)

cluster_wl	kmeans		Total
	1	2	
1	2	0	2
2	0	7	7
Total	2	7	9



Aunque el análisis estadístico por la regla de duda (Tabla XLV) nos lleva a quedarnos con un solo grupo, vamos a quedarnos con cuatro grupos, ya que, aunque nos dice que todas las facultades tuvieron el mismo comportamiento, vamos a ver como se comportaron dentro de ese único grupo.

**Tabla XLV.** Análisis estadístico para la selección del número de clusters (actividades)

Number of clusters	Duda/Hart	
	Je(2)/Je(1)	pseudo T-squared
1	0.4920	7.23
2	0.1921	12.62
3	0.2865	4.98
4	0.3535	3.66
5	0.2562	2.90
6	0.0000	.
7	0.0000	.

En la tabla XLVI podemos ver que el método Ward asigna al grupo 1 a la Facultad de Ciencias Experimentales, la Facultad de Humanidades, la Facultad de Trabajo Social, la Facultad de Ciencias del Trabajo y la ETSI, mientras que al grupo 2 asigna a la Facultad de Enfermería, la Facultad de Derecho, la Facultad de Ciencias Empresariales y a la Facultad de Ciencias de la Educación.

**Tabla XLVI.** Clusters para actividades creadas

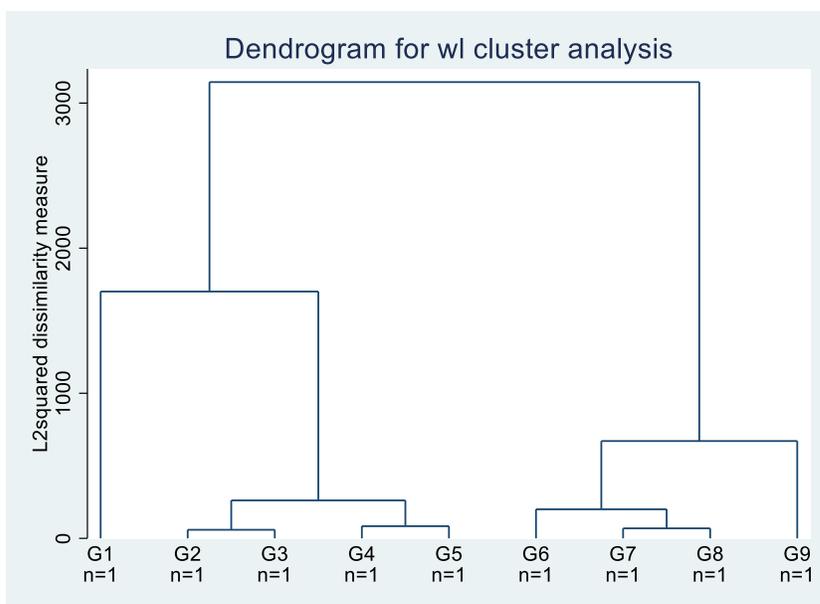
n_centro	cluster_wl		Total
	1	2	
c03	1	0	1
c08	1	0	1
c12	1	0	1
c14	1	0	1
c16	0	1	1
c18	1	0	1
c28	0	1	1
c30	0	1	1
c36	0	1	1
Total	5	4	9

Si observamos la media de las variables por grupo (Tabla XLVII), podemos ver que el grupo 1 es el que tiene mayor número de talleres, tareas y videos creados, mientras que el grupo 2 es el que tiene mayor media en número de glosarios, lecciones, cuestionarios y wikis creadas.

**Tabla XLVII.** Medias de las variables para cada grupo (actividades)

cluster_wl	n_glos~s	n_lecc~s	n_cues~s	n_tall~s	n_tareas	n_videos	n_wikis
1	0	.4	12.2	1	21	1.2	0
2	.5	.5	38.25	.25	15.75	.5	.25
Total	.2222222	.4444444	23.77778	.6666667	18.66667	.8888889	.1111111

En el dendrograma de las actividades creadas (Figura XXXIII) podemos ver que sucede lo mismo que con los recursos creados, con una distancia euclídea muy pequeña entre los subgrupos.



**Figura XXXIII.** Dendrograma de actividades creadas

Por último, si comparamos la asignación que realiza el método de Ward con el de K-means (Tabla XLVIII), podemos ver que ambos asignan las facultades a los mismos grupos.

**Tabla XLVIII.** Comparación de K-means y Ward (actividades creadas)

cluster_wl	kmeans		Total
	1	2	
1	5	0	5
2	0	4	4
Total	5	4	9

#### 4.6.4 Participación

En último lugar, vamos a analizar el fichero de participación. Si observamos la tabla de correlaciones (Tabla XLIX) podemos ver que existe una elevada correlación entre varias variables.

**Tabla XLIX.** Tabla de correlaciones de las participaciones (1/2)

	naz	nar	nac	npc	nag	npg	ntals
naz	1.0000						
nar	0.0370	1.0000					
nac	-0.0665	-0.2738	1.0000				
npc	0.0259	-0.2047	0.9889	1.0000			
nag	-0.4472	0.0107	-0.2723	-0.3064	1.0000		
npg	-0.4521	-0.0026	-0.2653	-0.3039	0.9990	1.0000	
ntals	0.1055	0.9410	-0.2802	-0.2304	-0.1870	-0.1889	1.0000
ntars	0.8704	-0.1436	-0.2727	-0.1965	-0.0581	-0.0635	-0.1241
nvs	-0.3874	-0.2469	0.0279	-0.0362	-0.2586	-0.2581	-0.1708
ncor	0.3723	0.7503	-0.1415	-0.0855	-0.3762	-0.3726	0.8309
ner	0.5951	-0.2966	-0.3416	-0.3418	0.1146	0.1318	-0.2014
ncur	0.7208	-0.1166	-0.2921	-0.2319	0.2316	0.2274	-0.1714
	ntars	nvs	ncor	ner	ncur		
ntars	1.0000						
nvs	-0.3853	1.0000					
ncor	0.0143	-0.3169	1.0000				
ner	0.7528	-0.2026	0.0649	1.0000			
ncur	0.9106	-0.3715	0.0050	0.8417	1.0000		

Tras probar diversas combinaciones, dado el elevado número de variables correlacionadas, finalmente emplearemos las participaciones en chats y glosarios y las tareas, talleres y videos subidos, dando lugar a la tabla L con las correlaciones entre ellas.

**Tabla L.** Tabla de correlaciones de las participaciones (2/2)

	npc	npg	ntals	ntars	nvs
npc	1.0000				
npg	-0.3039	1.0000			
ntals	-0.2304	-0.1889	1.0000		
ntars	-0.1965	-0.0635	-0.1241	1.0000	
nvs	-0.0362	-0.2581	-0.1708	-0.3853	1.0000

El análisis estadístico por la regla de duda (Tabla LI) nos indica que nos quedemos con 5 clusters.

**Tabla LI.** Análisis estadístico para la selección del número de clusters (participación)

Number of clusters	Duda/Hart	
	Je(2)/Je(1)	pseudo T-squared
1	0.1657	35.26
2	0.0000	.
3	0.4389	6.39
4	0.1083	32.92
5	0.5838	2.14
6	0.4382	2.56
7	0.3931	1.54

En la tabla LII tenemos la asignación que realiza el método de Ward, el cual asigna al grupo 5 a la Facultad de Ciencias Experimentales, la Facultad de Ciencias del Trabajo, la Facultad de Enfermería, la ETSI y la Facultad de Ciencias de la Educación, mientras que a las restantes (Facultad de Ciencias Empresariales, Facultad de Humanidades, Facultad de Trabajo Social y Facultad de Derecho) las asigna a grupos diferentes.

**Tabla LII.** Clusters para participación

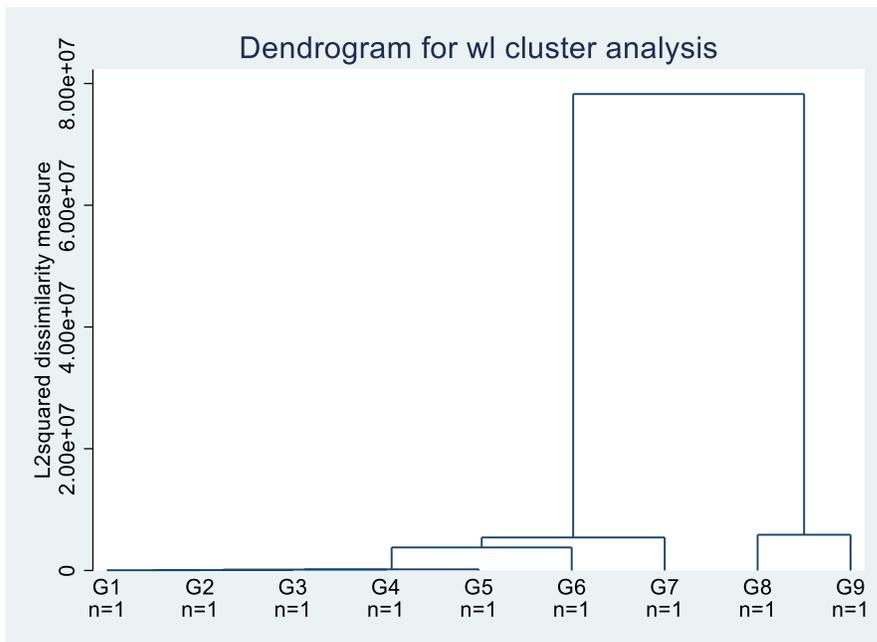
n_centro	cluster_wl					Total
	1	2	3	4	5	
c03	1	0	0	0	0	1
c08	0	0	1	0	0	1
c12	0	0	0	1	0	1
c14	1	0	0	0	0	1
c16	1	0	0	0	0	1
c18	1	0	0	0	0	1
c28	0	0	0	0	1	1
c30	0	1	0	0	0	1
c36	1	0	0	0	0	1
Total	5	1	1	1	1	9

En cuanto a las medias de las variables por grupo (Tabla LIII), podemos ver que el grupo 1 es el único con participación en glosarios y talleres subidos, el grupo 2 tan solo tiene tareas subidas, el grupo 3 es el que más videos subió (de los dos grupos en los que se subieron videos), el grupo 4 es en el que se subieron más tareas y el grupo 5 es en el que se participó más en los chats.

**Tabla LIII.** Medias de las variables para cada grupo (participación)

cluster_wl	npc	npg	ntals	ntars	nvs
1	110.8	29.6	39	832.2	4.6
2	0	0	0	2334	0
3	1791	0	0	559	56
4	4188	0	0	1307	0
5	6513	0	0	623	0
Total	1449.556	16.44444	21.66667	998.2222	8.777778

En el dendrograma (Figura XXXIV) podemos ver que las distancias euclídeas son muy superiores a los casos anteriores y que el grupo con mayor número de facultades es muy homogéneo.



**Figura XXXIV.** Dendrograma de participación

Por último, en la tabla LIV podemos ver que tanto el método de Ward como el de K-means realizan la misma asignación

**Tabla LIV.** Comparación de K-means y Ward (participación)

cluster_wl	kmeans					Total
	1	2	3	4	5	
1	5	0	0	0	0	5
2	0	1	0	0	0	1
3	0	0	1	0	0	1
4	0	0	0	1	0	1
5	0	0	0	0	1	1
Total	5	1	1	1	1	9

## 5 Conclusiones

1. Los usuarios pasaron más tiempo conectados e interactuando en 2020 como consecuencia de la docencia online.
2. Se realizó un mayor uso del servicio de aulas virtuales conforme avanzaba el curso, especialmente en 2020, además, dicho incremento parece estar relacionado con los anuncios que realizaba el Gobierno de España sobre el Estado de Alarma y la evolución de la pandemia.
3. Elevado número de conexiones y de acciones durante la noche, aunque se desconoce si se debe a usuarios en el extranjero.
4. Se realiza un mayor uso del servicio de aulas virtuales los lunes, decayendo conforme avanza la semana.
5. Se producen más interacciones interpersonales dentro de la plataforma a principios de la semana que a finales.
6. Los cuestionarios y las tareas fueron las actividades evaluables más empleadas por los docentes en ambos años.
7. Las sesiones de Zoom y los chats jugaron un papel fundamental en 2020 para el desarrollo de la docencia online.
8. Comparando los resultados con los datos ofrecidos por Blackboard, aunque el aumento en el uso de las aulas virtuales no fue tan elevado como el del uso de los servicios de Blackboard, si que se duplicó, en incluso se triplicó en algunos casos, el uso de las aulas virtuales y sus herramientas.
9. Comparando los resultados con el caso expuesto de la Universidad de Goce Delcev, si que se encuentran cifras muy similares en cuanto al incremento en el número de actividades realizadas a través de la plataforma.
10. En base a los análisis cluster realizados, no se puede determinar que haya dos o más facultades con un comportamiento similar.

## Referencias

- Blackboard. (1 de Abril de 2020). Infographic: The Impact of COVID-19 on Education. *Blackboard Blog*. Obtenido de <https://blog.blackboard.com/infographic-march-covid-19-impact-on-education/>
- Dascalu, M.-D., Ruseti, S., Dascalu, M., McNamara, D., Carabas, M., Rebedea, T., & Trausan-Matu, S. (Agosto de 2021). Before and during COVID-19: A Cohesion Network Analysis of students'. *Computers in Human Behavior*, 121.
- Hernández-García, Á., Acquila-Natale, E., Iglesias-Pradas, S., & Chaparro-Peláez, J. (2018). *Design of an Extraction, Transform and Load Process for Calculation of Teamwork Indicators in Moodle*, 62-73. Madrid, España: LASI-SPAIN.
- Lapevska, D., Velinov, A., & Zdravev, Z. (2021). Analysis of Moodle Activities Before and After the Covid-19 Pandemic—Case Study at Goce Delchev University. *Balkan Journal of Applied Mathematics and Informatics (BJAMI)*, 51-58.
- Moodle. (Agosto de 2021). Obtenido de Moodle.org: <https://docs.moodle.org/all/es/Historia>
- Portillo Peñuelas, S. A., Castellanos Pierra, L. I., Reynoso González, O. U., & Gavotto Nogales, O. I. (Septiembre de 2020). Enseñanza remota de emergencia ante la pandemia Covid-19 en Educación Media Superior y Educación. *Propósitos y Representaciones*, 8 SPE(3), 589.
- Ruiz-de-Azcárate, C., Hernández-García, Á., Iglesias-Pradas, S., & Acquila-Natale, E. (2017). *Proposal of a system of indicators to assess teamwork using log-based learning analytics*, 78-92. Madrid, España: LASI-SPAIN.

## Anexos

### Anexo I

Script SQL empleado para la extracción de datos de las diferentes tablas de la base de datos.

Las líneas de código mostradas a continuación fueron las empleadas para la tabla denominada como “c36\_19”, la cual se corresponde con el centro 36 durante el año 2019, para el resto de las tablas se procedía a sustituir el nombre de la tabla y las fechas según correspondieran.

Las fechas empleadas para las tablas correspondientes a 2020 son: '2020-02-17 00:00:00' y '2020-07-20 23:59:59'.

```
-- Número de inicios de sesión
select userid, count(*) as nis, hour(from_unixtime(timecreated)) as nis_h,
day(from_unixtime(timecreated)) as nis_d,
month(from_unixtime(timecreated)) as nis_m, week(from_unixtime(timecreated)) as nis_w
from c36_19
where eventname like '%user_loggedin' and from_unixtime(timecreated) between '2019-02-18 00:00:00'
and '2019-07-19 23:59:59'
group by userid, nis_w, nis_m, nis_d, nis_h
order by userid, nis_w, nis_m, nis_d, nis_h;

-- Vistas de inicios y cierres de sesión
create view inicios_sesion_c36_19 as (
select userid as usuario, (timecreated) as fecha, origin as plataforma, ip
from c36_19
where (eventname like '%user_loggedin') and from_unixtime(timecreated) between '2019-02-18 00:00:00'
and '2019-07-19 23:59:59' )
order by usuario asc, fecha asc)

create view cierres_sesion_c36_19 as (
select userid as usuario, (timecreated) as fecha, origin as plataforma, ip
from c36_19
where (eventname like '%user_loggedout') and from_unixtime(timecreated) between '2019-02-18
00:00:00' and '2019-07-19 23:59:59' )
order by usuario asc, fecha asc)

-- Mensajes
select userid as usuario, count(*) as num_mensajes_enviados, hour(from_unixtime(timecreated)) as
n_men_env_h,
day(from_unixtime(timecreated)) as n_men_env_d, month(from_unixtime(timecreated)) as
n_men_env_m,
week(from_unixtime(timecreated)) as n_men_env_w
```

```

from c36_19
where eventname like '%core%message_sent' and
from_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by userid, n_men_env_w, n_men_env_m, n_men_env_d, n_men_env_h
order by userid, n_men_env_w, n_men_env_m, n_men_env_d, n_men_env_h;

-- Ver calificaciones
select userid as usuario, count(*) as vistas_calif, hour(from_unixtime(timecreated)) as vistas_calif_h,
day(from_unixtime(timecreated)) as vistas_calif_d, month(from_unixtime(timecreated)) as
vistas_calif_m,
week(from_unixtime(timecreated)) as vistas_calif_w
from c36_19
where eventname like '%gradereport_user%grade_report_viewed' and
from_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by userid, vistas_calif_w, vistas_calif_m, vistas_calif_d, vistas_calif_h
order by userid, vistas_calif_w, vistas_calif_m, vistas_calif_d, vistas_calif_h;

-- Acciones
select userid as usuario, count(*) as t_acc, hour(from_unixtime(timecreated)) as t_acc_h,
day(from_unixtime(timecreated)) as t_acc_d, month(from_unixtime(timecreated)) as t_acc_m,
week(from_unixtime(timecreated)) as t_acc_w
from c36_19
where eventname not like '%user_loggedin' and eventname not like '%user_loggedout' and
from_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by userid, t_acc_w, t_acc_m, t_acc_d, t_acc_h
order by userid, t_acc_w, t_acc_m, t_acc_d, t_acc_h;

-- Vistas de foros creados y borrados
create view F_Creados_c36_19 as (
select userid as usuario, courseid as curso, count(*) as foros_creados, hour(from_unixtime(timecreated))
as f_c_h, day(from_unixtime(timecreated)) as f_c_d, month(from_unixtime(timecreated)) as f_c_m,
week(from_unixtime(timecreated)) as f_c_w
from c36_19
where (eventname like '%course_module_created') and (other like '%forum%') and
from_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by courseid, userid, f_c_w, f_c_m, f_c_d, f_c_h
order by courseid, userid, f_c_w, f_c_m, f_c_d, f_c_h);

create view F_Borrados_c36_19 as (
select userid as usuario, courseid as curso, count(*) as foros_borrados, hour(from_unixtime(timecreated))
as f_b_h, day(from_unixtime(timecreated)) as f_b_d, month(from_unixtime(timecreated)) as f_b_m,
week(from_unixtime(timecreated)) as f_b_w
from c36_19
where (eventname like '%course_module_deleted') and (other like '%forum%') and
from_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by courseid, userid, f_b_w, f_b_m, f_b_d, f_b_h
by courseid, userid, f_b_w, f_b_m, f_b_d, f_b_h);

-- Foros creados (bruto)
select *
from F_creados_c36_19

```

```

-- Foros creados (neto)
select C.usuario, C.curso, (case when isnull(C.foros_creados-B.foros_borrados) then C.foros_creados else
C.foros_creados-B.foros_borrados end) as foros_creados_netos, C.f_c_h as f_c_n_h, C.f_c_d as f_c_n_d,
C.f_c_m as f_c_n_m, C.f_c_w as f_c_n_w
from F_Creados_c36_19 C left join F_Borrados_c36_19 B on (C.usuario=B.usuario and
C.curso=B.curso)
group by C.curso, C.usuario, f_c_n_w, f_c_n_m, f_c_n_d, f_c_n_h
order by C.curso, C.usuario, f_c_n_w, f_c_n_m, f_c_n_d, f_c_n_h;

-- Vistas de discusiones creadas y borradas
create view D_Creadas_c36_19 as (
select userid as usuario, courseid as curso, count(*) as discusiones_creadas,
hour(from_unixtime(timecreated)) as d_c_h, day(from_unixtime(timecreated)) as d_c_d,
month(from_unixtime(timecreated)) as d_c_m, week(from_unixtime(timecreated)) as d_c_w
from c36_19
where eventname like '%discussion_created' and from_unixtime(timecreated) between '2019-02-18
00:00:00' and '2019-07-19 23:59:59'
group by courseid, userid, d_c_w, d_c_m, d_c_d, d_c_h
order by courseid, userid, d_c_w, d_c_m, d_c_d, d_c_h);

create view D_Borradas_c36_19 as (
select userid as usuario, courseid as curso, count(*) as discusiones_borradas,
hour(from_unixtime(timecreated)) as d_b_h, day(from_unixtime(timecreated)) as d_b_d,
month(from_unixtime(timecreated)) as d_b_m, week(from_unixtime(timecreated)) as d_b_w
from c36_19
where eventname like '%discussion_deleted' and from_unixtime(timecreated) between '2019-02-18
00:00:00' and '2019-07-19 23:59:59'
group by courseid, userid, d_b_w, d_b_m, d_b_d, d_b_h
order by courseid, userid, d_b_w, d_b_m, d_b_d, d_b_h);

-- Discusiones creadas (bruto)
select *
from D_creadas_c36_19

-- Discusiones creadas (neto)
select C.usuario, C.curso, (case when isnull(C.discusiones_creadas-B.discusiones_borradas) then
C.discusiones_creadas else C.discusiones_creadas-B.discusiones_borradas end) as
discusiones_creadas_netos, C.d_c_h as d_c_n_h, C.d_c_d as d_c_n_d, C.d_c_m as d_c_n_m, C.d_c_w as
d_c_n_w
from D_Creadas_c36_19 C left join D_Borradas_c36_19 B on (C.usuario=B.usuario and
C.curso=B.curso)
group by C.curso, C.usuario, d_c_n_w, d_c_n_m, d_c_n_d, d_c_n_h
order by C.curso, C.usuario, d_c_n_w, d_c_n_m, d_c_n_d, d_c_n_h;

-- Vistas de posts creados y borrados
create view P_Creados_c36_19 as (
select userid as usuario, courseid as curso, count(*) as posts_creados, hour(from_unixtime(timecreated))
as p_c_h, day(from_unixtime(timecreated)) as p_c_d, month(from_unixtime(timecreated)) as p_c_m,
week(from_unixtime(timecreated)) as p_c_w
from c36_19
where eventname like '%post_created' and from_unixtime(timecreated) between '2019-02-18 00:00:00'
and '2019-07-19 23:59:59'

```

```
group by courseid, userid, p_c_w, p_c_m, p_c_d, p_c_h
order by courseid, userid, p_c_w, p_c_m, p_c_d, p_c_h);
```

```
create view P_Borrados_c36_19 as (
select userid as usuario, courseid as curso, count(*) as posts_borrados, hour(from_unixtime(timecreated))
as p_b_h, day(from_unixtime(timecreated)) as p_b_d, month(from_unixtime(timecreated)) as p_b_m,
week(from_unixtime(timecreated)) as p_b_w
from c36_19
where (eventname like '%post_deleted') and (objectid not in (select objectid from c36_19 where
eventname like '%discussion_deleted') ) and from_unixtime(timecreated) between '2019-02-18 00:00:00'
and '2019-07-19 23:59:59'
group by courseid, userid, p_b_w, p_b_m, p_b_d, p_b_h
order by courseid, userid, p_b_w, p_b_m, p_b_d, p_b_h);
```

```
-- Posts creados (bruto)
select *
from P_creados_c36_19
```

```
-- Posts creados (neto)
select C.usuario, C.curso, (case when isnull(C.posts_creados-B.posts_borrados) then C.posts_creados else
C.posts_creados-B.posts_borrados end) as posts_creados_netos, C.p_c_h as p_c_n_h, C.p_c_d as
p_c_n_d, C.p_c_m as p_c_n_m, C.p_c_w as p_c_n_w
from P_Creados_c36_19 C left join P_Borrados_c36_19 B on (C.usuario=B.usuario and
C.curso=B.curso)
group by C.curso, C.usuario, p_c_n_w, p_c_n_m, p_c_n_d, p_c_n_h
order by C.curso, C.usuario, p_c_n_w, p_c_n_m, p_c_n_d, p_c_n_h;
```

```
-- Base de datos
select userid as usuario, courseid as curso, count(*) as n_bd, hour(from_unixtime(timecreated)) as
n_bd_h, day(from_unixtime(timecreated)) as n_bd_d, month(from_unixtime(timecreated)) as n_bd_m,
week(from_unixtime(timecreated)) as n_bd_w
from c36_19
where (eventname like '%course_module_created') and (other like '%data%') and
from_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by usuario, curso, n_bd_w, n_bd_m, n_bd_d, n_bd_h
order by usuario, curso, n_bd_w, n_bd_m, n_bd_d, n_bd_h;
```

```
-- Chat
select userid as usuario, courseid as curso, count(*) as n_chats, hour(from_unixtime(timecreated)) as
n_chat_h, day(from_unixtime(timecreated)) as n_chat_d, month(from_unixtime(timecreated)) as
n_chat_m, week(from_unixtime(timecreated)) as n_chat_w
from c36_19
where (eventname like '%course_module_created') and (other like '%chat%') and
from_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by usuario, curso, n_chat_w, n_chat_m, n_chat_d, n_chat_h
order by usuario, curso, n_chat_w, n_chat_m, n_chat_d, n_chat_h;
```

```
-- Accesos chat
select courseid as curso, count(*) as nac, hour(from_unixtime(timecreated)) as nac_h,
day(from_unixtime(timecreated)) as nac_d, month(from_unixtime(timecreated)) as nac_m,
week(from_unixtime(timecreated)) as nac_w
from c36_19
```

```

where (eventname like '%mod_chat%course_module_viewed') and from_unixtime(timecreated) between
'2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by courseid, nac_w, nac_m, nac_d, nac_h
order by courseid, nac_w, nac_m, nac_d, nac_h;

```

-- Participación chat

```

select courseid as curso, count(*) as npc, hour(from_unixtime(timecreated)) as npc_h,
day(from_unixtime(timecreated)) as npc_d, month(from_unixtime(timecreated)) as npc_m,
week(from_unixtime(timecreated)) as npc_w
from c36_19
where (eventname like '%mod_chat%message_sent') and from_unixtime(timecreated) between '2019-02-
18 00:00:00' and '2019-07-19 23:59:59'
group by courseid, npc_w, npc_m, npc_d, npc_h
order by courseid, npc_w, npc_m, npc_d, npc_h;

```

-- Consulta

```

select userid as usuario, courseid as curso, count(*) as n_consultas, hour(from_unixtime(timecreated)) as
n_consultas_h, day(from_unixtime(timecreated)) as n_consultas_d, month(from_unixtime(timecreated))
as n_consultas_m, week(from_unixtime(timecreated)) as n_consultas_w
from c36_19
where (eventname like '%course_module_created') and (other like '%choice%') and
from_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by usuario, curso, n_consultas_w, n_consultas_m, n_consultas_d, n_consultas_h
order by usuario, curso, n_consultas_w, n_consultas_m, n_consultas_d, n_consultas_h;

```

-- Consultas respondidas

```

select courseid as curso, count(*) as ncor, hour(from_unixtime(timecreated)) as ncor_h,
day(from_unixtime(timecreated)) as ncor_d, month(from_unixtime(timecreated)) as ncor_m,
week(from_unixtime(timecreated)) as ncor_w
from c36_19
where eventname like '%mod_choice%answer_created' and from_unixtime(timecreated) between '2019-
02-18 00:00:00' and '2019-07-19 23:59:59'
group by courseid, ncor_w, ncor_m, ncor_d, ncor_h
order by courseid, ncor_w, ncor_m, ncor_d, ncor_h;

```

-- Cuestionario

```

select userid as usuario, courseid as curso, count(*) as n_cuestionarios,
hour(from_unixtime(timecreated)) as n_cuestionarios_h, day(from_unixtime(timecreated)) as
n_cuestionarios_d, month(from_unixtime(timecreated)) as n_cuestionarios_m,
week(from_unixtime(timecreated)) as n_cuestionarios_w
from c36_19
where (eventname like '%course_module_created') and (other like '%quiz%') and
from_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by usuario, curso, n_cuestionarios_w, n_cuestionarios_m, n_cuestionarios_d, n_cuestionarios_h
order by usuario, curso, n_cuestionarios_w, n_cuestionarios_m, n_cuestionarios_d, n_cuestionarios_h;

```

-- Cuestionarios realizados

```

select courseid as curso, count(*) as ncur, hour(from_unixtime(timecreated)) as ncur_h,
day(from_unixtime(timecreated)) as ncur_d, month(from_unixtime(timecreated)) as ncur_m,
week(from_unixtime(timecreated)) as ncur_w
from c36_19

```

where eventname like '%mod\_quiz%attempt\_submitted' and from\_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'

group by courseid, ncur\_w, ncur\_m, ncur\_d, ncur\_h  
order by courseid, ncur\_w, ncur\_m, ncur\_d, ncur\_h;

-- Encuesta

select userid as usuario, courseid as curso, count(\*) as n\_encuestas, hour(from\_unixtime(timecreated)) as n\_encuestas\_h, day(from\_unixtime(timecreated)) as n\_encuestas\_d, month(from\_unixtime(timecreated)) as n\_encuestas\_m, week(from\_unixtime(timecreated)) as n\_encuestas\_w  
from c36\_19

where (eventname like '%course\_module\_created') and (other like '%feedback%') and from\_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'  
group by usuario, curso, n\_encuestas\_w, n\_encuestas\_m, n\_encuestas\_d, n\_encuestas\_h  
order by usuario, curso, n\_encuestas\_w, n\_encuestas\_m, n\_encuestas\_d, n\_encuestas\_h;

-- Encuestas respondidas

select courseid as curso, count(\*) as ner, hour(from\_unixtime(timecreated)) as ner\_h, day(from\_unixtime(timecreated)) as ner\_d, month(from\_unixtime(timecreated)) as ner\_m, week(from\_unixtime(timecreated)) as ner\_w  
from c36\_19

where eventname like '%mod\_feedback%response\_submitted' and from\_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'

group by courseid, ner\_w, ner\_m, ner\_d, ner\_h  
order by courseid, ner\_w, ner\_m, ner\_d, ner\_h;

-- Encuesta predefinida

select userid as usuario, courseid as curso, count(\*) as n\_encuestas\_pred, hour(from\_unixtime(timecreated)) as n\_encuestas\_pred\_h, day(from\_unixtime(timecreated)) as n\_encuestas\_pred\_d, month(from\_unixtime(timecreated)) as n\_encuestas\_pred\_m, week(from\_unixtime(timecreated)) as n\_encuestas\_pred\_w  
from c36\_19

where (eventname like '%course\_module\_created') and (other like '%survey%') and from\_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'  
group by usuario, curso, n\_encuestas\_pred\_w, n\_encuestas\_pred\_m, n\_encuestas\_pred\_d, n\_encuestas\_pred\_h  
order by usuario, curso, n\_encuestas\_pred\_w, n\_encuestas\_pred\_m, n\_encuestas\_pred\_d, n\_encuestas\_pred\_h;

-- Glosario

select userid as usuario, courseid as curso, count(\*) as n\_glosarios, hour(from\_unixtime(timecreated)) as n\_glosarios\_h, day(from\_unixtime(timecreated)) as n\_glosarios\_d, month(from\_unixtime(timecreated)) as n\_glosarios\_m, week(from\_unixtime(timecreated)) as n\_glosarios\_w  
from c36\_19

where (eventname like '%course\_module\_created') and (other like '%glossary%') and from\_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'  
group by usuario, curso, n\_glosarios\_w, n\_glosarios\_m, n\_glosarios\_d, n\_glosarios\_h  
order by usuario, curso, n\_glosarios\_w, n\_glosarios\_m, n\_glosarios\_d, n\_glosarios\_h;

-- Accesos glosario

```
select courseid as curso, count(*) as nag, hour(from_unixtime(timecreated)) as nag_h,
day(from_unixtime(timecreated)) as nag_d, month(from_unixtime(timecreated)) as nag_m,
week(from_unixtime(timecreated)) as nag_w
from c36_19
where (eventname like '%mod_glossary%course_module_viewed') and from_unixtime(timecreated)
between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by courseid, nag_w, nag_m, nag_d, nag_h
order by courseid, nag_w, nag_m, nag_d, nag_h;
```

-- Participación glosario

```
select courseid as curso, count(*) as npg, hour(from_unixtime(timecreated)) as npg_h,
day(from_unixtime(timecreated)) as npg_d, month(from_unixtime(timecreated)) as npg_m,
week(from_unixtime(timecreated)) as npg_w
from c36_19
where (eventname like '%mod_glossary%entry_created') and
from_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by courseid, npg_w, npg_m, npg_d, npg_h
order by courseid, npg_w, npg_m, npg_d, npg_h;
```

-- Herramienta externa

```
select userid as usuario, courseid as curso, count(*) as n_herr_ext, hour(from_unixtime(timecreated)) as
n_herr_ext_h, day(from_unixtime(timecreated)) as n_herr_ext_d, month(from_unixtime(timecreated)) as
n_herr_ext_m, week(from_unixtime(timecreated)) as n_herr_ext_w
from c36_19
where (eventname like '%course_module_created') and (other like '%lti%') and
from_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by usuario, curso, n_herr_ext_w, n_herr_ext_m, n_herr_ext_d, n_herr_ext_h
order by usuario, curso, n_herr_ext_w, n_herr_ext_m, n_herr_ext_d, n_herr_ext_h;
```

-- Leccion

```
select userid as usuario, courseid as curso, count(*) as n_lecciones, hour(from_unixtime(timecreated)) as
n_lecciones_h, day(from_unixtime(timecreated)) as n_lecciones_d, month(from_unixtime(timecreated))
as n_lecciones_m, week(from_unixtime(timecreated)) as n_lecciones_w
from c36_19
where (eventname like '%course_module_created') and (other like '%lesson%') and
from_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by usuario, curso, n_lecciones_w, n_lecciones_m, n_lecciones_d, n_lecciones_h
order by usuario, curso, n_lecciones_w, n_lecciones_m, n_lecciones_d, n_lecciones_h;
```

-- Taller

```
select userid as usuario, courseid as curso, count(*) as n_talleres, hour(from_unixtime(timecreated)) as
n_talleres_h, day(from_unixtime(timecreated)) as n_talleres_d, month(from_unixtime(timecreated)) as
n_talleres_m, week(from_unixtime(timecreated)) as n_talleres_w
from c36_19
where (eventname like '%course_module_created') and (other like '%workshop%') and
from_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by usuario, curso, n_talleres_w, n_talleres_m, n_talleres_d, n_talleres_h
order by usuario, curso, n_talleres_w, n_talleres_m, n_talleres_d, n_talleres_h;
```

-- Talleres enviados

```
select courseid as curso, count(*) as ntals, hour(from_unixtime(timecreated)) as ntals_h,
day(from_unixtime(timecreated)) as ntals_d, month(from_unixtime(timecreated)) as ntals_m,
week(from_unixtime(timecreated)) as ntals_w
from c36_19
where eventname like '%mod_workshop%assessable_uploaded' and from_unixtime(timecreated) between
'2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by courseid, ntals_w, ntals_m, ntals_d, ntals_h
order by courseid, ntals_w, ntals_m, ntals_d, ntals_h;
```

-- Tarea

```
select userid as usuario, courseid as curso, count(*) as n_tareas, hour(from_unixtime(timecreated)) as
n_tareas_h, day(from_unixtime(timecreated)) as n_tareas_d, month(from_unixtime(timecreated)) as
n_tareas_m, week(from_unixtime(timecreated)) as n_tareas_w
from c36_19
where (eventname like '%course_module_created') and (other like '%assign%') and
from_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by usuario, curso, n_tareas_w, n_tareas_m, n_tareas_d, n_tareas_h
order by usuario, curso, n_tareas_w, n_tareas_m, n_tareas_d, n_tareas_h;
```

-- Tareas enviadas

```
select courseid as curso, count(*) as ntars, hour(from_unixtime(timecreated)) as ntars_h,
day(from_unixtime(timecreated)) as ntars_d, month(from_unixtime(timecreated)) as ntars_m,
week(from_unixtime(timecreated)) as ntars_w
from c36_19
where eventname like '%mod_assign%assessable_submitted' and from_unixtime(timecreated) between
'2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by courseid, ntars_w, ntars_m, ntars_d, ntars_h
order by courseid, ntars_w, ntars_m, ntars_d, ntars_h;
```

-- Wiki

```
select userid as usuario, courseid as curso, count(*) as n_wikis, hour(from_unixtime(timecreated)) as
n_wikis_h, day(from_unixtime(timecreated)) as n_wikis_d, month(from_unixtime(timecreated)) as
n_wikis_m, week(from_unixtime(timecreated)) as n_wikis_w
from c36_19
where (eventname like '%course_module_created') and (other like '%wiki%') and
from_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by usuario, curso, n_wikis_w, n_wikis_m, n_wikis_d, n_wikis_h
order by usuario, curso, n_wikis_w, n_wikis_m, n_wikis_d, n_wikis_h;
```

-- Archivo

```
select userid as usuario, courseid as curso, count(*) as n_archivos, hour(from_unixtime(timecreated)) as
n_archivos_h, day(from_unixtime(timecreated)) as n_archivos_d, month(from_unixtime(timecreated)) as
n_archivos_m, week(from_unixtime(timecreated)) as n_archivos_w
from c36_19
where (eventname like '%course_module_created') and (other like '%resource%') and
from_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by usuario, curso, n_archivos_w, n_archivos_m, n_archivos_d, n_archivos_h
order by usuario, curso, n_archivos_w, n_archivos_m, n_archivos_d, n_archivos_h;
```

-- Carpeta

```
select userid as usuario, courseid as curso, count(*) as n_carpetas, hour(from_unixtime(timecreated)) as
n_carpetas_h, day(from_unixtime(timecreated)) as n_carpetas_d, month(from_unixtime(timecreated)) as
n_carpetas_m, week(from_unixtime(timecreated)) as n_carpetas_w
from c36_19
where (eventname like '%course_module_created') and (other like '%folder%') and
from_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by usuario, curso, n_carpetas_w, n_carpetas_m, n_carpetas_d, n_carpetas_h
order by usuario, curso, n_carpetas_w, n_carpetas_m, n_carpetas_d, n_carpetas_h;
```

-- Etiqueta

```
select userid as usuario, courseid as curso, count(*) as n_etiquetas, hour(from_unixtime(timecreated)) as
n_etiquetas_h, day(from_unixtime(timecreated)) as n_etiquetas_d, month(from_unixtime(timecreated)) as
n_etiquetas_m, week(from_unixtime(timecreated)) as n_etiquetas_w
from c36_19
where (eventname like '%course_module_created') and (other like '%label%') and
from_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by usuario, curso, n_etiquetas_w, n_etiquetas_m, n_etiquetas_d, n_etiquetas_h
order by usuario, curso, n_etiquetas_w, n_etiquetas_m, n_etiquetas_d, n_etiquetas_h;
```

-- Libro

```
select userid as usuario, courseid as curso, count(*) as n_libros, hour(from_unixtime(timecreated)) as
n_libros_h, day(from_unixtime(timecreated)) as n_libros_d, month(from_unixtime(timecreated)) as
n_libros_m, week(from_unixtime(timecreated)) as n_libros_w
from c36_19
where (eventname like '%course_module_created') and (other like '%book%') and
from_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by usuario, curso, n_libros_w, n_libros_m, n_libros_d, n_libros_h
order by usuario, curso, n_libros_w, n_libros_m, n_libros_d, n_libros_h;
```

-- Pagina

```
select userid as usuario, courseid as curso, count(*) as n_paginas, hour(from_unixtime(timecreated)) as
n_paginas_h, day(from_unixtime(timecreated)) as n_paginas_d, month(from_unixtime(timecreated)) as
n_paginas_m, week(from_unixtime(timecreated)) as n_paginas_w
from c36_19
where (eventname like '%course_module_created') and (other like '%page%') and
from_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by usuario, curso, n_paginas_w, n_paginas_m, n_paginas_d, n_paginas_h
order by usuario, curso, n_paginas_w, n_paginas_m, n_paginas_d, n_paginas_h;
```

-- URL

```
select userid as usuario, courseid as curso, count(*) as n_urls, hour(from_unixtime(timecreated)) as
n_urls_h, day(from_unixtime(timecreated)) as n_urls_d, month(from_unixtime(timecreated)) as
n_urls_m, week(from_unixtime(timecreated)) as n_urls_w
from c36_19
where (eventname like '%course_module_created') and (other like '%url%') and
from_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by usuario, curso, n_urls_w, n_urls_m, n_urls_d, n_urls_h
order by usuario, curso, n_urls_w, n_urls_m, n_urls_d, n_urls_h;
```

-- Asistencia

```
select userid as usuario, courseid as curso, count(*) as n_asistencia, hour(from_unixtime(timecreated)) as
n_asistencia_h, day(from_unixtime(timecreated)) as n_asistencia_d, month(from_unixtime(timecreated))
as n_asistencia_m, week(from_unixtime(timecreated)) as n_asistencia_w
from c36_19
where (eventname like '%course_module_created') and (other like '%attendance%') and
from_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by usuario, curso, n_asistencia_w, n_asistencia_m, n_asistencia_d, n_asistencia_h
order by usuario, curso, n_asistencia_w, n_asistencia_m, n_asistencia_d, n_asistencia_h;
```

-- Registros asistencia

```
select courseid as curso, count(*) as nar, hour(from_unixtime(timecreated)) as nar_h,
day(from_unixtime(timecreated)) as nar_d, month(from_unixtime(timecreated)) as nar_m,
week(from_unixtime(timecreated)) as nar_w
from c36_19
where eventname like '%mod_attendance%session_added' and from_unixtime(timecreated) between
'2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by courseid, nar_w, nar_m, nar_d, nar_h
order by courseid, nar_w, nar_m, nar_d, nar_h;
```

-- Videos

```
select userid as usuario, courseid as curso, count(*) as n_videos, hour(from_unixtime(timecreated)) as
n_videos_h, day(from_unixtime(timecreated)) as n_videos_d, month(from_unixtime(timecreated)) as
n_videos_m, week(from_unixtime(timecreated)) as n_videos_w
from c36_19
where (eventname like '%course_module_created') and (other like '%kalvidas%') and
from_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by usuario, curso, n_videos_w, n_videos_m, n_videos_d, n_videos_h
order by usuario, curso, n_videos_w, n_videos_m, n_videos_d, n_videos_h;
```

-- Videos subidos

```
select courseid as curso, count(*) as nvs, hour(from_unixtime(timecreated)) as nvs_h,
day(from_unixtime(timecreated)) as nvs_d, month(from_unixtime(timecreated)) as nvs_m,
week(from_unixtime(timecreated)) as nvs_w
from c36_19
where eventname like '%mod_kalvidassign%assignment_submitted' and from_unixtime(timecreated)
between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by courseid, nvs_w, nvs_m, nvs_d, nvs_h
order by courseid, nvs_w, nvs_m, nvs_d, nvs_h;
```

-- Zoom

```
select userid as usuario, courseid as curso, count(*) as n_szoom, hour(from_unixtime(timecreated)) as
n_szoom_h, day(from_unixtime(timecreated)) as n_szoom_d, month(from_unixtime(timecreated)) as
n_szoom_m, week(from_unixtime(timecreated)) as n_szoom_w
from c36_19
where (eventname like '%course_module_created') and (other like '%zoom%') and
from_unixtime(timecreated) between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by usuario, curso, n_szoom_w, n_szoom_m, n_szoom_d, n_szoom_h
order by usuario, curso, n_szoom_w, n_szoom_m, n_szoom_d, n_szoom_h;
```

```
-- Accesos Zoom
select courseid as curso, count(*) as naz, hour(from_unixtime(timecreated)) as naz_h,
day(from_unixtime(timecreated)) as naz_d, month(from_unixtime(timecreated)) as naz_m,
week(from_unixtime(timecreated)) as naz_w
from c36_19
where eventname like '%mod_zoom%join_meeting_button_clicked' and from_unixtime(timecreated)
between '2019-02-18 00:00:00' and '2019-07-19 23:59:59'
group by courseid, naz_w, naz_m, naz_d, naz_h
order by courseid, naz_w, naz_m, naz_d, naz_h;
```

## Anexo II

Librerías y variables empleadas en Python.

Las variables contienen las listas de profesores por año y centro, la codificación de los centros, los dos años analizados y los nombres de los ficheros generados.

- Librerías:

```
import mariadb, sys, copy, csv
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from datetime import datetime, timedelta
from operator import itemgetter
```

- Variables:

```
c03_19 = [542, 642, 1083, 18309, 12317, 37989, 38620]
c03_20 = [1083, 16684, 60270, 12317, 37989, 46065, 38620, 47445, 11971, 35412]
c08_19 = [1350, 1417, 15545, 16630, 47076, 5343, 7732, 12017, 14109, 14624, 14797, 557, 14431]
c08_20 = [1350, 15545, 46609, 14624, 14797, 14109, 12017, 7732, 5343, 557, 14431, 47076]
c12_19 = [539, 31028, 39374, 54835, 55204, 686, 3488, 46627, 54391]
c12_20 = [539, 31028, 39374, 54835, 55204, 32107, 46054, 686, 3488, 54391, 46026, 71568]
c14_19 = [14555, 23524, 18233]
c14_20 = [23524, 14555, 18233, 1779]
c16_19 = [634, 1584, 32112, 37009, 56373, 311, 1739, 12021, 59001, 552, 3367, 13555, 18635, 57684]
c16_20 = [32112, 56373, 1584, 32123, 1746, 311, 1739, 12021, 3367, 552, 18635, 13555, 44676, 69861]
c18_19 = [4190, 234, 2263, 55189, 1796, 1046, 35, 566, 32998, 16628, 542, 44008, 45756]
c18_20 = [4190, 234, 2263, 55189, 1796, 1046, 35, 566, 32998, 16628, 542, 44008, 45756]
c28_19 = [475, 14557, 14454, 14743, 35422, 1090, 47091]
c28_20 = [475, 14557, 14454, 14743, 1090]
c30_19 = [498, 568, 569, 594, 595, 631, 1747, 1800, 24210, 58914, 58920, 615, 1766, 3950, 4242,
55072]
c30_20 = [1800, 631, 498, 24210, 594, 615, 1747, 595, 1766, 4242, 3950]
c36_19 = [14610, 490, 1743, 4747, 4361, 15708, 55770, 57837, 14577,4361, 43852, 35238]
c36_20 = [14610, 490, 1743, 4747, 4361, 15708, 55770, 57837, 14577,4361, 43852, 35238]
```

```
centros = ['c03', 'c08', 'c12', 'c14', 'c16', 'c18', 'c28', 'c30', 'c36']
años = ['19', '20']
```

```
nom_arc_1 = ['tc']
nom_arc_2 = ['ta']
nom_arc_3 = ['nis', 'tacc', 'vcalif', 'nmenenv']
nom_arc_4 = ['fc', 'fcn', 'dc', 'dcn', 'pc', 'pcn']
nom_arc_5 = ['acczoom', 'regasist', 'accchats', 'partchats', 'accglos', 'partglos', 'envtall', 'envtar', \
'envvid', 'nconsr', 'nencr', 'nquizr']
nom_arc_6 = ['narc', 'nasist', 'nbd', 'ncarp', 'ncons', 'nchats', 'nenc', 'nencpredef', 'netiq', 'nglos', \
'nherrext', 'nlecc', 'nlibros', 'npag', 'nquiz', 'nszoom', 'ntall', 'ntar', 'nurl', 'nvid', 'nwiki']
nom_arc_7 = ['lal']
```

## Anexo III

Funciones y bucle de Python para la obtención del tiempo activo y del tiempo conectado a la plataforma de cada uno de los usuarios.

- Función para obtener todas las acciones de un usuario:

```
def acciones_usuario(tabla, ID):
    acciones = []
    cur.execute("select eventname, userid, timecreated, origin, ip \
        from %s \
        where userid=%s" % (tabla, ID) )
    for row in cur.fetchall():
        acciones.append(row)
    return acciones
```

- Función para obtener el tiempo conectado de todos los usuarios de un centro en un año:

```
def tiempo_conectado(LU, tabla):
    from datetime import datetime, timedelta
    from operator import itemgetter

    TC = [['id', 'tiempo_conectado', 'plataforma', 'fecha']]

    for user in range(len(LU)):
        acciones = sorted(acciones_usuario(tabla, LU[user][0]), key=itemgetter(2))
        pacc, tact, acca = 0, 0, 0
        origen, fecha = "", ""
        for acc in range(len(acciones)):
            if (acciones[acc][0] == "\\core\\event\\user_loggedin' or pacc == 0) and acca == 0:
                pacc = acciones[acc][2]
                origen = acciones[acc][3]
                fecha = str(datetime.strptime(str(datetime.fromtimestamp(acciones[acc][2])), "%Y-%m-%d
%H:%M:%S").date()))

            elif acciones[acc][0] == "\\core\\event\\user_loggedin' and pacc != 0 and acca != 0 :
                tact = str(datetime.fromtimestamp(acca)-datetime.fromtimestamp(pacc))
                TC.append([LU[user][0], tact, origen, fecha])
                pacc = acciones[acc][2]
                origen = acciones[acc][3]
                fecha = str(datetime.strptime(str(datetime.fromtimestamp(acciones[acc][2])), "%Y-%m-%d
%H:%M:%S").date()))
                tact, acca = 0, 0

        else:
            if acca == 0 and acciones[acc][0] != "\\core\\event\\user_loggedout' and \
                ((datetime.fromtimestamp(acciones[acc][2]) - datetime.fromtimestamp(pacc)) <=
                timedelta(hours=2)):
```

```

    acca = acciones[acc][2]

    elif acca != 0 and acciones[acc][0] != '\\core\\event\\user_loggedout' and \
    ((datetime.fromtimestamp(acciones[acc][2]) - datetime.fromtimestamp(acca)) <=
timedelta(hours=2)):
        acca = acciones[acc][2]

    elif acca == 0 and acciones[acc][0] != '\\core\\event\\user_loggedout' and \
    ((datetime.fromtimestamp(acciones[acc][2]) - datetime.fromtimestamp(pacc)) >
timedelta(hours=2)):
        pacc = acciones[acc][2]
        origen = acciones[acc][3]
        fecha = str(datetime.strptime(str(datetime.fromtimestamp(acciones[acc][2])), "%Y-%m-%d
%H:%M:%S").date())

    elif acca != 0 and acciones[acc][0] != '\\core\\event\\user_loggedout' and \
    ((datetime.fromtimestamp(acciones[acc][2]) - datetime.fromtimestamp(acca)) >
timedelta(hours=2)):
        tact = str(datetime.fromtimestamp(acca)-datetime.fromtimestamp(pacc))
        TC.append([LU[user][0], tact, origen, fecha])
        pacc = acciones[acc][2]
        origen = acciones[acc][3]
        fecha = str(datetime.strptime(str(datetime.fromtimestamp(acciones[acc][2])), "%Y-%m-%d
%H:%M:%S").date())
        tact, acca = 0, 0

    if acciones[acc][0] == '\\core\\event\\user_loggedout':
        if acc == 0 or acciones[acc-1][0] == '\\core\\event\\user_loggedout':
            pass

    elif acca != 0 and \
    ((datetime.fromtimestamp(acciones[acc][2]) - datetime.fromtimestamp(acca)) <=
timedelta(hours=4)):
        acca = acciones[acc][2]
        tact = str(datetime.fromtimestamp(acca)-datetime.fromtimestamp(pacc))
        TC.append([LU[user][0], tact, origen, fecha])
        pacc, tact, acca = 0, 0, 0
        origen, fecha = ", "

    elif acca == 0 and \
    ((datetime.fromtimestamp(acciones[acc][2]) - datetime.fromtimestamp(pacc)) <=
timedelta(hours=4)):
        acca = acciones[acc][2]
        tact = str(datetime.fromtimestamp(acca)-datetime.fromtimestamp(pacc))
        TC.append([LU[user][0], tact, origen, fecha])
        pacc, tact, acca = 0, 0, 0
        origen, fecha = ", "

    elif (acc+1) == len(acciones) and acciones[acc][0] != '\\core\\event\\user_loggedin' and \
    ((datetime.fromtimestamp(acciones[acc][2]) - datetime.fromtimestamp(acca)) <=
timedelta(hours=2)):
        acca = acciones[acc][2]

```

```
tact = str(datetime.fromtimestamp(acca)-datetime.fromtimestamp(pacc))
TC.append([LU[user][0], tact, origen, fecha])
pacc, tact, acca = 0, 0, 0
origen, fecha = "", "
```

```
return TC
```

- Función para obtener el tiempo activo de todos los usuarios de un centro en un año:

```
def tiempo_activo(LU, tabla):
    from datetime import datetime, timedelta
    from operator import itemgetter

    TA = [['id', 'tiempo_activo', 'plataforma', 'fecha']]
    for user in range(len(LU)):
        acciones = sorted(acciones_usuario(tabla, LU[user][0]), key=itemgetter(2))
        pacc, tact, acca = 0, 0, 0
        origen, fecha = "", ""
        for acc in range(len(acciones)):
            if (acciones[acc][0] == "\\core\\event\\user_loggedin' or pacc == 0) and acca == 0:
                pacc = acciones[acc][2]
                origen = acciones[acc][3]
                fecha = str(datetime.strptime(str(datetime.fromtimestamp(acciones[acc][2])), "%Y-%m-%d
%H:%M:%S").date()))

                elif acciones[acc][0] == "\\core\\event\\user_loggedin' and pacc != 0 and acca != 0 :
                    tact = str(datetime.fromtimestamp(acca)-datetime.fromtimestamp(pacc))
                    TA.append([LU[user][0], tact, origen, fecha])
                    pacc = acciones[acc][2]
                    origen = acciones[acc][3]
                    fecha = str(datetime.strptime(str(datetime.fromtimestamp(acciones[acc][2])), "%Y-%m-%d
%H:%M:%S").date()))
                    tact, acca = 0, 0

            else:
                if acca == 0 and acciones[acc][0] != "\\core\\event\\user_loggedout' and \
                    ((datetime.fromtimestamp(acciones[acc][2]) - datetime.fromtimestamp(pacc)) <=
                    timedelta(minutes=15)):
                        acca = acciones[acc][2]

                elif acca != 0 and acciones[acc][0] != "\\core\\event\\user_loggedout' and \
                    ((datetime.fromtimestamp(acciones[acc][2]) - datetime.fromtimestamp(acca)) <=
                    timedelta(minutes=15)):
                        acca = acciones[acc][2]

                elif acca == 0 and acciones[acc][0] != "\\core\\event\\user_loggedout' and \
                    ((datetime.fromtimestamp(acciones[acc][2]) - datetime.fromtimestamp(pacc)) >
                    timedelta(minutes=15)):
                        pacc = acciones[acc][2]
                        origen = acciones[acc][3]
```

```

        fecha = str(datetime.strptime(str(datetime.fromtimestamp(acciones[acc][2])), "%Y-%m-%d
%H:%M:%S").date())

        elif acca != 0 and acciones[acc][0] != '\\core\\event\\user_loggedout' and \
        ((datetime.fromtimestamp(acciones[acc][2]) - datetime.fromtimestamp(pacc)) >
timedelta(minutes=15)):
            tact = str(datetime.fromtimestamp(acca)-datetime.fromtimestamp(pacc))
            TA.append([LU[user][0], tact, origen, fecha])
            pacc = acciones[acc][2]
            origen = acciones[acc][3]
            fecha = str(datetime.strptime(str(datetime.fromtimestamp(acciones[acc][2])), "%Y-%m-%d
%H:%M:%S").date())
            tact, acca = 0, 0

        if acciones[acc][0] == '\\core\\event\\user_loggedout':
            if acc == 0 or acciones[acc-1][0] == '\\core\\event\\user_loggedout':
                pass

            elif acca != 0 and \
            ((datetime.fromtimestamp(acciones[acc][2]) - datetime.fromtimestamp(acca)) <=
timedelta(hours=1)):
                acca = acciones[acc][2]
                tact = str(datetime.fromtimestamp(acca)-datetime.fromtimestamp(pacc))
                TA.append([LU[user][0], tact, origen, fecha])
                pacc, tact, acca = 0, 0, 0
                origen, fecha = ", "

            elif acca == 0 and \
            ((datetime.fromtimestamp(acciones[acc][2]) - datetime.fromtimestamp(pacc)) <=
timedelta(hours=1)):
                acca = acciones[acc][2]
                tact = str(datetime.fromtimestamp(acca)-datetime.fromtimestamp(pacc))
                TA.append([LU[user][0], tact, origen, fecha])
                pacc, tact, acca = 0, 0, 0
                origen, fecha = ", "

            elif (acc+1) == len(acciones) and acciones[acc][0] != '\\core\\event\\user_loggedin' and \
            ((datetime.fromtimestamp(acciones[acc][2]) - datetime.fromtimestamp(acca)) <=
timedelta(minutes=15)):
                acca = acciones[acc][2]
                tact = str(datetime.fromtimestamp(acca)-datetime.fromtimestamp(pacc))
                TA.append([LU[user][0], tact, origen, fecha])
                pacc, tact, acca = 0, 0, 0
                origen, fecha = ", "

    return TA

```

- Bucle para la ejecución de las funciones anteriores:

```

for año in años:
    if año == '19':
        db = 'curso18' + año
    if año == '20':
        db = 'curso19' + año

    try:
        con = mariadb.connect(
            user = "root",
            password = "",
            host = "localhost",
            port = 3306,
            database = db)

    except mariadb.Error as e:
        print(f"Error connecting to MariaDB Platform: {e}")
        sys.exit(1)

    cur = con.cursor()

    for centro in centros:
        cya = centro + '_' + año

        cur.execute(("select distinct userid\
            from %s") % cya)
        LU = []
        for row in cur.fetchall():
            LU.append(list(row))

        TC = tiempo_conectado(LU, cya)
        csv.writer(open("TFM/%s" % ('20' + año + '/' + centro + '/' + cya + '_tc.csv'), 'w',
            newline=")).writerows(TC)

        TA = tiempo_activo(LU, cya)
        csv.writer(open("TFM/%s" % ('20' + año + '/' + centro + '/' + cya + '_ta.csv'), 'w',
            newline=")).writerows(TA)

    con.close()

```

## Anexo IV

Código para la obtención de las ID correspondientes a los alumnos de cada centro y año.

```

for año in años:
    if año == '19':
        db = 'curso18' + año
    if año == '20':
        db = 'curso19' + año

    try:
        con = mariadb.connect(
            user = "root",
            password = "",
            host = "localhost",
            port = 3306,
            database = db)

    except mariadb.Error as e:
        print(f"Error connecting to MariaDB Platform: {e}")
        sys.exit(1)

    cur = con.cursor()

    for centro in centros:
        cya = centro + '_' + año

        cur.execute(("select distinct userid as usuario \
            from %s \
            where userid in (select distinct userid \
                from %s \
                where courseid not in (0)) \
            and userid in (select distinct userid \
                from %s \
                where eventname like %s and other like %s) \
            order by userid") % (cya, cya, cya, "%user_loggedin", "%alu@%"))

        LAL = [['IDs']]
        for row in cur.fetchall():
            LAL.append(list(row))

        csv.writer(open("TFM/%s" % ('20' + año + '/' + centro + '/' + cya + '_lal.csv'), 'w',
            newline=")).writerows(LAL)

    con.close()

```

## Anexo V

Código para la adecuación de los ficheros CSV generados, renombrando correctamente las columnas, poniendo al final la columna con el recuento, además de juntar las columnas de día y mes en una nueva con el año incorporado y añadir una columna con la codificación del centro.

```

for año in años:
    for centro in centros:
        cya = centro + '_' + año
        for num in range(3,7):
            arc_iterado = 'nom_arc_' + str(num)
            for archivo in vars()[arc_iterado]:

                datos = list(csv.reader(open('TFM/%s' % ('20' + año + '/' + centro + '/' + cya + '_' + archivo +
'.csv'), 'r')))
                count = 0

                if len(datos[0]) == 7:

                    for fila in datos:

                        if count == 0:
                            fila.append(fila.pop(fila.index(fila[2])))
                            fila.insert(fila.index(fila[2]), 'hora')
                            fila.pop(fila.index(fila[3]))
                            fila.insert(fila.index(fila[5]), 'semana')
                            fila.pop(fila.index(fila[6]))
                            fila[3:5] = ['fecha']
                            fila.insert(2, 'centro')

                            count += 1

                        else:
                            fila.append(fila.pop(fila.index(fila[2])))
                            fila[3:5] = ['20' + año + '-' + fila[4] + '-' + fila[3]]
                            fila.insert(2, centro)

                csv.writer(open("TFM/corregidos/%s" % ('20' + año + '/' + centro + '/' + cya + '_' + archivo +
'.csv'), 'w', newline=")).writerows(datos)

                elif len(datos[0]) == 6:

                    for fila in datos:

                        if count == 0:

                            fila.append(fila.pop(fila.index(fila[1])))

```

```

        fila.insert(fila.index(fila[1]), 'hora')
        fila.pop(fila.index(fila[2]))
        fila.insert(fila.index(fila[4]), 'semana')
        fila.pop(fila.index(fila[5]))
        fila[2:4] = ['fecha']
        fila.insert(1, 'centro')
        count += 1

    if fila[0] == 'userid':
        fila.insert(fila.index(fila[0]), 'usuario')
        fila.pop(fila.index(fila[1]))

    else:

        fila.append(fila.pop(fila.index(fila[1])))
        fila[2:4] = ['20' + año + '-' + fila[3] + '-' + fila[2]]
        fila.insert(1, centro)

    csv.writer(open("TFM/corregidos/%s" % ('20' + año + '/' + centro + '/' + cya + '_' + archivo +
'.csv'), 'w', newline=")).writerows(datos)

    for num in [1,2,7]:

        arc_iterado = 'nom_arc_' + str(num)

        for archivo in vars()[arc_iterado]:

            datos = list(csv.reader(open("TFM/%s" % ('20' + año + '/' + centro + '/' + cya + '_' + archivo +
'.csv'), 'r')))

            if num == 7:

                datos[0].insert(datos[0].index(datos[0][0]), 'usuario')
                datos[0].pop(datos[0].index(datos[0][1]))

            elif num == 1 or num == 2:
                count = 0

            for fila in datos:

                if count == 0:
                    datos[0].insert(datos[0].index(datos[0][0]), 'usuario')
                    datos[0].pop(datos[0].index(datos[0][1]))
                    fila.insert(1, 'centro')
                    count += 1

                else:
                    fila.insert(1, centro)

    csv.writer(open("TFM/corregidos/%s" % ('20' + año + '/' + centro + '/' + cya + '_' + archivo +
'.csv'), 'w', newline=")).writerows(datos)

```

## Anexo VI

Código para incorporar las listas de profesores a las de los alumnos.

```

for año in años:
    for centro in centros:
        cya = centro + '_' + año

        lista = []
        fichero = list(csv.reader(open('TFM/%s' % ('20' + año + '/' + centro + '/' + cya + '_lal.csv'), 'r')))
        count = 0
        for row in fichero:
            if count == 0:
                count += 1
            else:
                lista.append([int(row[0])])
        for ID in vars()[cya]:
            if [ID] not in lista:
                lista.append([ID])

        for num in [4,6]:
            arc_iterado = 'nom_arc_' + str(num)

            for archivo in vars()[arc_iterado]:
                fichero = list(csv.reader(open('TFM/%s' % ('20' + año + '/' + centro + '/' + cya + '_' + archivo +
                '.csv'), 'r')))
                count = 0
                for row in fichero:
                    if count == 0:
                        count += 1
                    else:
                        if [int(row[0])] not in lista:
                            lista.append([int(row[0])])

        lista = sorted(lista)
        lista.insert(0, ['usuario'])
        csv.writer(open("TFM/corregidos/%s" % ('20' + año + '/' + centro + '/' + cya + '_lpya.csv'), 'w',
        newline=")).writerows(lista)

```

## Anexo VII

Código para unificar todos los ficheros generados en seis y eliminar los registros innecesarios (aquellos que no pertenecen a alumnos o docentes de cada curso) en los casos en los que procede.

```

for num in range(1,7):
    arc_iterado = 'nom_arc_' + str(num)
    if num in [4,6]:
        arc_combinar = []
        for archivo in vars()[arc_iterado]:
            for año in años:
                for centro in centros:
                    cya = centro + '_' + año
                    fichero = pd.read_csv('TFM/corregidos/%s' % ('20' + año + '/' + centro + '/' + cya + '_' +
archivo + '.csv'), parse_dates = ['fecha'])
                    fichero = fichero.set_index(fichero.columns.values[0:6].tolist())
                    arc_combinar.append(fichero)
                df = pd.concat(arc_combinar, ignore_index = False, axis=0)
                df = df.reset_index(drop = False)
                df = df.fillna(0)
                df = df.groupby(df.columns.values[0:6].tolist()).sum()
                df = df.reset_index(drop = False)
                df[df.columns[0:2]] = df[df.columns[0:2]].astype(int)
                df[df.columns[3]] = df[df.columns[3]].astype(int)
                df[df.columns[5:]] = df[df.columns[5:]].astype(int)
            if num == 4:
                df.to_csv('TFM/combinados/foros.csv')
            if num == 6:
                df.to_csv('TFM/combinados/recursos_creados.csv')
    if num == 5:
        arc_combinar = []
        for archivo in vars()[arc_iterado]:
            for año in años:
                for centro in centros:
                    cya = centro + '_' + año
                    fichero = pd.read_csv('TFM/corregidos/%s' % ('20' + año + '/' + centro + '/' + cya + '_' +
archivo + '.csv'), parse_dates = ['fecha'])
                    fichero = fichero.set_index(fichero.columns.values[0:5].tolist())
                    arc_combinar.append(fichero)
                df = pd.concat(arc_combinar, ignore_index = False, axis=0)
                df = df.reset_index(drop = False)
                df = df.fillna(0)
                df = df.groupby(df.columns.values[0:5].tolist()).sum()
                df = df.reset_index(drop = False)
                df[df.columns[0]] = df[df.columns[0]].astype(int)
                df[df.columns[2]] = df[df.columns[2]].astype(int)

```

```

df[df.columns[4:]] = df[df.columns[4:]].astype(int)
df.to_csv('TFM/combinados/participacion.csv')
if num == 3:
    arc_combinar = []
    for archivo in vars()[arc_iterado]:
        for año in años:
            for centro in centros:
                cya = centro + '_' + año
                fichero = pd.read_csv('TFM/corregidos/%s' % ('20' + año + '/' + centro + '/' + cya + '_' +
archivo + '.csv'), parse_dates = ['fecha'])
                lista = list(csv.reader(open('TFM/corregidos/%s' % ('20' + año + '/' + centro + '/' + cya +
'_lpya.csv'), 'r')))
                usuarios = []
                for i in lista[1:]:
                    usuarios.append(i[0])
                fichero = fichero[fichero.usuario.isin(usuarios)]
                fichero = fichero.set_index(fichero.columns.values[0:5].tolist())
                arc_combinar.append(fichero)
df = pd.concat(arc_combinar, ignore_index = False, axis=0)
df = df.reset_index(drop = False)
df = df.fillna(0)
df = df.groupby(df.columns.values[0:5].tolist()).sum()
df = df.reset_index(drop = False)
df[df.columns[0]] = df[df.columns[0]].astype(int)
df[df.columns[2]] = df[df.columns[2]].astype(int)
df[df.columns[4:]] = df[df.columns[4:]].astype(int)
df.to_csv('TFM/combinados/otras_acciones.csv')
if num in [1,2]:
    arc_combinar = []
    for archivo in vars()[arc_iterado]:
        for año in años:
            for centro in centros:
                cya = centro + '_' + año
                fichero = pd.read_csv('TFM/corregidos/%s' % ('20' + año + '/' + centro + '/' + cya + '_' +
archivo + '.csv'), parse_dates = ['fecha'])
                lista = list(csv.reader(open('TFM/corregidos/%s' % ('20' + año + '/' + centro + '/' + cya +
'_lpya.csv'), 'r')))
                usuarios = []
                for i in lista[1:]:
                    usuarios.append(i[0])
                fichero = fichero[fichero.usuario.isin(usuarios)]
                arc_combinar.append(fichero)
df = pd.concat(arc_combinar, ignore_index = False, axis=0)
df[df.columns[0]] = df[df.columns[0]].astype(int)
df[df.columns[3]] = df[df.columns[3]].astype(str)
if num == 1:
    df.to_csv('TFM/combinados/tiempo_conectado.csv')
if num == 2:
    df.to_csv('TFM/combinados/tiempo_activo.csv')

```

## Anexo VIII

Códigos empleados para el análisis del tiempo conectado.

```
df1 = pd.read_csv('TFM/combinados/tiempo_conectado.csv', parse_dates = ['fecha'])
df1.tiempo_conectado = pd.to_timedelta(df1.tiempo_conectado)
df1
```

```
media = df1.groupby(df1['fecha'].dt.to_period('Y'))['tiempo_conectado'].apply(lambda x:
x.mean()).rename("Tiempo medio conectado")
maximo = df1.groupby(df1['fecha'].dt.to_period('Y'))['tiempo_conectado'].apply(lambda x:
x.max()).rename("Tiempo máximo conectado")
minimo = df1.groupby(df1['fecha'].dt.to_period('Y'))['tiempo_conectado'].apply(lambda x:
x.min()).rename("Tiempo mínimo conectado")
desv = df1.groupby(df1['fecha'].dt.to_period('Y'))['tiempo_conectado'].apply(lambda x:
x.std()).rename("Desviación típica")
df11 = pd.concat([media, maximo, minimo, desv],axis = 1)
df11.index.name = 'Año'
df11
```

```
print(df1['tiempo_conectado'].where(df1.fecha.dt.year==2019).count())
print(df1['tiempo_conectado'].where(df1.fecha.dt.year==2020).count())
```

```
plt.figure(figsize=(10,6))
df1['tiempo_conectado'].where(df1.fecha.dt.year==2019).astype('timedelta64[m]').plot.hist(bins=range(0,
800,10), edgecolor = 'black')
plt.title("Tiempo conectado en 2019")
plt.xlabel('Minutos')
plt.ylabel('Nº de conexiones')
plt.grid(linestyle = '-.', alpha = 0.75)
plt.gca().spines['right'].set_color('white')
plt.gca().spines['top'].set_color('white')
plt.axis([0,180,0,110000])
plt.show()
```

```
plt.figure(figsize=(10,6))
df1['tiempo_conectado'].where(df1.fecha.dt.year==2019).astype('timedelta64[m]').plot.hist(bins=range(0,
800,10), edgecolor = 'black')
plt.title("Tiempo conectado en 2019")
plt.xlabel('Minutos')
plt.ylabel('Nº de conexiones')
plt.grid(linestyle = '-.', alpha = 0.75)
```

```
plt.gca().spines['right'].set_color('white')
plt.gca().spines['top'].set_color('white')
plt.axis([120,800,0,1500])
plt.show()
```

```
plt.figure(figsize=(10,6))
df1['tiempo_conectado'].where(df1.fecha.dt.year==2020).astype('timedelta64[m]').plot.hist(bins=range(0,
800,10), edgecolor = 'black')
plt.title("Tiempo conectado en 2020")
plt.xlabel('Minutos')
plt.ylabel('N° de conexiones')
plt.grid(linestyle = '-.', alpha = 0.75)
plt.gca().spines['right'].set_color('white')
plt.gca().spines['top'].set_color('white')
plt.axis([0,180,0,200000])
plt.show()
```

```
plt.figure(figsize=(10,6))
df1['tiempo_conectado'].where(df1.fecha.dt.year==2020).astype('timedelta64[m]').plot.hist(bins=range(0,
800,10), edgecolor = 'black')
plt.title("Tiempo conectado en 2020")
plt.xlabel('Minutos')
plt.ylabel('N° de conexiones')
plt.grid(linestyle = '-.', alpha = 0.75)
plt.gca().spines['right'].set_color('white')
plt.gca().spines['top'].set_color('white')
plt.axis([120,800,0,3000])
plt.show()
```

```
df12 = df1.set_index([df1['fecha'].dt.to_period('Y'), 'centro'], inplace = False)[['tiempo_conectado']]
df12 = df12.reindex(df12.index.rename(['Año', 'Centro']))
media = df12.groupby(['Centro', 'Año']).apply(lambda x: x.mean())
media.columns = ['Tiempo medio conectado']
maximo = df12.groupby(['Centro', 'Año']).apply(lambda x: x.max())
maximo.columns = ['Tiempo máximo conectado']
minimo = df12.groupby(['Centro', 'Año']).apply(lambda x: x.min())
minimo.columns = ['Tiempo mínimo conectado']
desv = df12.groupby(['Centro', 'Año']).apply(lambda x: x.std())
desv.columns = ['Desviación típica']
df13 = pd.concat([media, maximo, minimo, desv], axis = 1)
df13
```

```
Meses = {1: 'Enero', 2: 'Febrero', 3: 'Marzo', 4: 'Abril', 5: 'Mayo', 6: 'Junio', 7: 'Julio', 8: 'Agosto', \
9: 'Septiembre', 10: 'Octubre', 11: 'Noviembre', 14: 'Diciembre'}
df14 = df1[['tiempo_conectado', 'fecha']]
df14 = df14.set_index([df14['fecha'].dt.to_period('Y')], inplace = False)
```

```

df14.index.name = 'Año'
df14['Mes'] = df14['fecha'].dt.month
df14['Mes'] = df14['Mes'].apply(lambda x: Meses[x])
df14 = df14.set_index(['Mes'], inplace = False, append = True)[['tiempo_conectado']]
media = df14.groupby(['Año', 'Mes']).apply(lambda x: x.mean())
media.columns = ['Tiempo medio conectado']
maximo = df14.groupby(['Año', 'Mes']).apply(lambda x: x.max())
maximo.columns = ['Tiempo máximo conectado']
minimo = df14.groupby(['Año', 'Mes']).apply(lambda x: x.min())
minimo.columns = ['Tiempo mínimo conectado']
desv = df14.groupby(['Año', 'Mes']).apply(lambda x: x.std())
desv.columns = ['Desviación típica']
df15 = pd.concat([media, maximo, minimo, desv], axis = 1)
df15 = df15.sort_values(by=['Año', 'Tiempo medio conectado'], ascending = True)
df15

```

```

dias_semana = {0:'Lunes', 1:'Martes', 2:'Miercoles', 3:'Jueves', 4:'Viernes', 5:'Sabado', 6:'Domingo'}
df16 = df1[['tiempo_conectado', 'fecha']]
df16 = df16.set_index([df1['fecha'].dt.to_period('Y')], inplace = False)
df16.index.name = 'Año'
df16['Día'] = df16['fecha'].dt.dayofweek
df16['Día'] = df16['Día'].apply(lambda x: dias_semana[x])
df16 = df16.set_index(['Día'], inplace = False, append = True)[['tiempo_conectado']]
media = df16.groupby(['Año', 'Día']).apply(lambda x: x.mean())
media.columns = ['Tiempo medio conectado']
maximo = df16.groupby(['Año', 'Día']).apply(lambda x: x.max())
maximo.columns = ['Tiempo máximo conectado']
minimo = df16.groupby(['Año', 'Día']).apply(lambda x: x.min())
minimo.columns = ['Tiempo mínimo conectado']
desv = df16.groupby(['Año', 'Día']).apply(lambda x: x.std())
desv.columns = ['Desviación típica']
df17 = pd.concat([media, maximo, minimo, desv], axis = 1)
df17 = df17.sort_values(by=['Año', 'Tiempo medio conectado'], ascending = True)
df17

```

## Anexo IX

Códigos empleados para el análisis del tiempo activo.

```
df2 = pd.read_csv('TFM/combinados/tiempo_activo.csv', parse_dates=['fecha'])
df2.tiempo_activo = pd.to_timedelta(df2.tiempo_activo)
df2
```

```
media = df2.groupby(df2['fecha'].dt.to_period('Y'))['tiempo_activo'].apply(lambda x:
x.mean()).rename('Tiempo medio activo')
maximo = df2.groupby(df2['fecha'].dt.to_period('Y'))['tiempo_activo'].apply(lambda x:
x.max()).rename('Tiempo máximo activo')
minimo = df2.groupby(df2['fecha'].dt.to_period('Y'))['tiempo_activo'].apply(lambda x:
x.min()).rename('Tiempo mínimo activo')
desv = df2.groupby(df2['fecha'].dt.to_period('Y'))['tiempo_activo'].apply(lambda x:
x.std()).rename('Desviación típica')
df21 = pd.concat([media, maximo, minimo, desv],axis=1)
df21.index.name = 'Año'
df21
```

```
print(df2['tiempo_activo'].where(df2.fecha.dt.year==2019).count())
print(df2['tiempo_activo'].where(df2.fecha.dt.year==2020).count())
```

```
plt.figure(figsize=(10,6))
df2['tiempo_activo'].where(df2.fecha.dt.year==2019).astype('timedelta64[m]').plot.hist(bins=range(0,250,
10), edgecolor = 'black')
plt.title('Tiempo activo en 2019')
plt.xlabel('Minutos')
plt.ylabel('Nº de conexiones')
plt.grid(linestyle = '-.', alpha = 0.75)
plt.gca().spines['right'].set_color('white')
plt.gca().spines['top'].set_color('white')
plt.axis([0,60,0,170000])
plt.show()
```

```
plt.figure(figsize=(10,6))
df2['tiempo_activo'].where(df2.fecha.dt.year==2019).astype('timedelta64[m]').plot.hist(bins=range(0,250,
10), edgecolor = 'black')
plt.title('Tiempo activo en 2019')
plt.xlabel('Minutos')
plt.ylabel('Nº de conexiones')
plt.grid(linestyle = '-.', alpha = 0.75)
```

```
plt.gca().spines['right'].set_color('white')
plt.gca().spines['top'].set_color('white')
plt.axis([60,250,0,350])
plt.show()
```

```
plt.figure(figsize=(10,6))
df2['tiempo_activo'].where(df2.fecha.dt.year==2020).astype('timedelta64[m]').plot.hist(bins=range(0,350,
10), edgecolor = 'black')
plt.title("Tiempo activo en 2020")
plt.xlabel('Minutos')
plt.ylabel('N° de conexiones')
plt.grid(linestyle = '-.', alpha = 0.75)
plt.gca().spines['right'].set_color('white')
plt.gca().spines['top'].set_color('white')
plt.axis([0,60,0,300000])
plt.show()
```

```
plt.figure(figsize=(10,6))
df2['tiempo_activo'].where(df2.fecha.dt.year==2020).astype('timedelta64[m]').plot.hist(bins=range(0,350,
10), edgecolor = 'black')
plt.title("Tiempo activo en 2020")
plt.xlabel('Minutos')
plt.ylabel('N° de conexiones')
plt.grid(linestyle = '-.', alpha = 0.75)
plt.gca().spines['right'].set_color('white')
plt.gca().spines['top'].set_color('white')
plt.axis([60,320,0,1200])
plt.show()
```

```
df22 = df2.set_index(['centro', df2['fecha'].dt.to_period('Y')], inplace = False)[['tiempo_activo']]
df22 = df22.reindex(df22.index.rename(['Centro', 'Año']))
media = df22.groupby(['Centro', 'Año']).apply(lambda x: x.mean())
media.columns = ['Tiempo medio activo']
maximo = df22.groupby(['Centro', 'Año']).apply(lambda x: x.max())
maximo.columns = ['Tiempo máximo activo']
minimo = df22.groupby(['Centro', 'Año']).apply(lambda x: x.min())
minimo.columns = ['Tiempo mínimo activo']
desv = df22.groupby(['Centro', 'Año']).apply(lambda x: x.std())
desv.columns = ['Desviación típica']
df23 = pd.concat([media, maximo, minimo, desv],axis=1)
df23
```

```
Meses = { 1: 'Enero', 2: 'Febrero', 3: 'Marzo', 4: 'Abril', 5: 'Mayo', 6: 'Junio', 7: 'Julio', 8: 'Agosto', \
          9: 'Septiembre', 10: 'Octubre', 11: 'Noviembre', 12: 'Diciembre' }
df24 = df2[['tiempo_activo', 'fecha']]
df24 = df24.set_index([df2['fecha'].dt.to_period('Y')], inplace = False)
```

```

df24.index.name = 'Año'
df24['Mes'] = df24['fecha'].dt.month
df24['Mes'] = df24['Mes'].apply(lambda x: Meses[x])
df24 = df24.set_index(['Mes'], inplace = False, append = True)[['tiempo_activo']]
media = df24.groupby(['Año', 'Mes']).apply(lambda x: x.mean())
media.columns = ['Tiempo medio activo']
maximo = df24.groupby(['Año', 'Mes']).apply(lambda x: x.max())
maximo.columns = ['Tiempo máximo activo']
minimo = df24.groupby(['Año', 'Mes']).apply(lambda x: x.min())
minimo.columns = ['Tiempo mínimo activo']
desv = df24.groupby(['Año', 'Mes']).apply(lambda x: x.std())
desv.columns = ['Desviación típica']
df25 = pd.concat([media, maximo, minimo, desv], axis = 1)
df25 = df25.sort_values(by=['Año', 'Tiempo medio activo'], ascending = True)
df25

```

```

dias_semana = {0:'Lunes', 1:'Martes', 2:'Miercoles', 3:'Jueves', 4:'Viernes', 5:'Sabado', 6:'Domingo'}
df26 = df2[['tiempo_activo', 'fecha']]
df26 = df26.set_index([df2['fecha'].dt.to_period('Y')], inplace = False)
df26.index.name = 'Año'
df26['Día'] = df26['fecha'].dt.dayofweek
df26['Día'] = df26['Día'].apply(lambda x: dias_semana[x])
df26 = df26.set_index(['Día'], inplace = False, append = True)[['tiempo_activo']]
media = df26.groupby(['Año', 'Día']).apply(lambda x: x.mean())
media.columns = ['Tiempo medio activo']
maximo = df26.groupby(['Año', 'Día']).apply(lambda x: x.max())
maximo.columns = ['Tiempo máximo activo']
minimo = df26.groupby(['Año', 'Día']).apply(lambda x: x.min())
minimo.columns = ['Tiempo mínimo activo']
desv = df26.groupby(['Año', 'Día']).apply(lambda x: x.std())
desv.columns = ['Desviación típica']
df27 = pd.concat([media, maximo, minimo, desv], axis = 1)
df27 = df27.sort_values(by=['Año', 'Tiempo medio activo'], ascending = True)
df27

```

## Anexo X

Códigos empleados para el análisis del fichero con otras acciones.

```
df3 = pd.read_csv('TFM/combinados/otras_acciones.csv', parse_dates=['fecha'])
df3
```

```
df31 = df3[df3.columns[5:]].where(df3.fecha.dt.year == 2019).sum().astype(int).to_frame()
df31.columns = ['2019']
df31['2020'] = df3[df3.columns[5:]].where(df3.fecha.dt.year == 2020).sum().astype(int).to_frame()
df31['Total'] = df3[df3.columns[5:]].where(df3.fecha.dt.year == 2019).sum().astype(int) + \
df3[df3.columns[5:]].where(df3.fecha.dt.year == 2020).sum().astype(int)
df31
```

```
plt.subplot(1,2,1)
df321 = df3['nis'].where(df3.fecha.dt.year == 2019).groupby(df3['centro']).sum().to_frame()
labels = df321.where(df321.nis != 0).dropna().index
explode = (0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05)
plt.pie(np.array(df321.get(df321['nis']!=0)).ravel(), labels = labels, explode = explode, autopct =
'%1.1f%%', \
    pctdistance = 1.3, shadow = True, labeldistance = 1.65)
plt.title('Nº de inicios de sesión en 2019',pad = 30.0)
plt.subplot(1,2,2)
df322 = df3['nis'].where(df3.fecha.dt.year == 2020).groupby(df3['centro']).sum().to_frame()
labels = df322.where(df322.nis != 0).dropna().index
explode = (0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05)
plt.pie(np.array(df322.get(df322['nis']!=0)).ravel(), labels = labels, explode = explode, autopct =
'%1.1f%%', \
    pctdistance = 1.3, shadow = True, labeldistance = 1.65)
plt.title('Nº de inicios de sesión en 2020', pad=30.0)
plt.subplots_adjust(wspace=0.6)
plt.show()
```

```
plt.subplot(1,2,1)
df331 = df3['t_acc'].where(df3.fecha.dt.year == 2019).groupby(df3['centro']).sum().to_frame()
labels = df331.where(df331.t_acc != 0).dropna().index
explode = (0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05)
plt.pie(np.array(df331.get(df331['t_acc']!=0)).ravel(), labels = labels, explode = explode, autopct =
'%1.1f%%', \
    pctdistance = 1.3, shadow = True, labeldistance = 1.65)
plt.title('Total de acciones en 2019',pad = 30.0)
plt.subplot(1,2,2)
df332 = df3['t_acc'].where(df3.fecha.dt.year == 2020).groupby(df3['centro']).sum().to_frame()
labels = df332.where(df332.t_acc != 0).dropna().index
```

```

explode = (0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05)
plt.pie(np.array(df332.get(df332['t_acc']!=0)).ravel(), labels = labels, explode = explode, autopct =
'%1.1f%%', \
    pctdistance = 1.3, shadow = True, labeldistance = 1.65)
plt.title('Total de acciones en 2020', pad=30.0)
plt.subplots_adjust(wspace=0.6)
plt.show()

```

```

Meses = { 1: 'Enero', 2: 'Febrero', 3: 'Marzo', 4: 'Abril', 5: 'Mayo', 6: 'Junio', 7: 'Julio', 8: 'Agosto', \
    9: 'Septiembre', 10: 'Octubre', 11: 'Noviembre', 12: 'Diciembre'}
df34 = df3[['fecha', 'nis', 't_acc']]
df34 = df34.set_index([df34['fecha'].dt.to_period('Y')], inplace = False)
df34.index.name = 'Año'
df34['Mes'] = df34['fecha'].dt.month
df34['Mes'] = df34['Mes'].apply(lambda x: Meses[x])
df341 = df34.loc['2019']
df341 = df341.groupby(['Mes', 'Año']).sum().unstack().fillna(0)
df341.plot(kind='bar', rot=45)
plt.grid(linestyle = '-.', alpha = 0.75, axis = 'y')
plt.gca().spines['right'].set_color('white')
plt.gca().spines['top'].set_color('white')
plt.legend(title=False, frameon=False, labels=['nis','t_acc'])
plt.xlabel('Mes')
plt.title('2019')
plt.show()

```

```

df342 = df34.loc['2020']
df342 = df342.groupby(['Mes', 'Año']).sum().unstack().fillna(0)
df342.plot(kind='bar', rot=45)
plt.grid(linestyle = '-.', alpha = 0.75, axis = 'y')
plt.gca().spines['right'].set_color('white')
plt.gca().spines['top'].set_color('white')
plt.legend(title=False, frameon=False, labels=['nis','t_acc'])
plt.xlabel('Mes')
plt.title('2020')
plt.show()

```

```

Meses = { 1: 'Enero', 2: 'Febrero', 3: 'Marzo', 4: 'Abril', 5: 'Mayo', 6: 'Junio', 7: 'Julio', 8: 'Agosto', \
    9: 'Septiembre', 10: 'Octubre', 11: 'Noviembre', 12: 'Diciembre'}
df35 = df3[['fecha', 'vistas_calif', 'num_mensajes_enviados']]
df35 = df35.set_index([df35['fecha'].dt.to_period('Y')], inplace = False)
df35.index.name = 'Año'
df35['Mes'] = df35['fecha'].dt.month
df35['Mes'] = df35['Mes'].apply(lambda x: Meses[x])
df351 = df35.loc['2019']
df351 = df351.groupby(['Mes', 'Año']).sum().unstack().fillna(0)
df351.plot(kind='bar', rot=45)

```

```

plt.grid(linestyle = '-.', alpha = 0.75, axis = 'y')
plt.gca().spines['right'].set_color('white')
plt.gca().spines['top'].set_color('white')
plt.legend(title=False, frameon=False, labels=['vistas_calif','num_mensajes_enviados'])
plt.xlabel('Mes')
plt.title('2019')
plt.show()

```

```

df342 = df34.loc['2020']
df342 = df342.groupby(['Mes', 'Año']).sum().unstack().fillna(0)
df342.plot(kind='bar', rot=45)
plt.grid(linestyle = '-.', alpha = 0.75, axis = 'y')
plt.gca().spines['right'].set_color('white')
plt.gca().spines['top'].set_color('white')
plt.legend(title=False, frameon=False, labels=['vistas_calif','num_mensajes_enviados'])
plt.xlabel('Mes')
plt.title('2020')
plt.show()

```

```

dias_semana = {0:'Lunes', 1:'Martes', 2:'Miercoles', 3:'Jueves', 4:'Viernes', 5:'Sabado', 6:'Domingo'}
df36 = df3[['nis', 'fecha']]
df36 = df36.set_index([df3['fecha'].dt.to_period('Y')], inplace = False)
df36.index.name = 'Año'
df36['Día'] = df36['fecha'].dt.dayofweek
df36['Día'] = df36['Día'].apply(lambda x: dias_semana[x])
df36 = df36.groupby(['Día', 'Año']).sum().unstack().fillna(0)
df36.plot(kind='bar', color=['darkorange','blue'], rot=45)
plt.grid(linestyle = '-.', alpha = 0.75, axis = 'y')
plt.gca().spines['right'].set_color('white')
plt.gca().spines['top'].set_color('white')
plt.legend(title=False, frameon=False, labels=['2019', '2020'])
plt.ylabel('Nº de inicios de sesión')
plt.xlabel('Día de la semana')
plt.show()

```

```

dias_semana = {0:'Lunes', 1:'Martes', 2:'Miercoles', 3:'Jueves', 4:'Viernes', 5:'Sabado', 6:'Domingo'}
df37 = df3[['t_acc', 'fecha']]
df37 = df37.set_index([df3['fecha'].dt.to_period('Y')], inplace = False)
df37.index.name = 'Año'
df37['Día'] = df37['fecha'].dt.dayofweek
df37['Día'] = df37['Día'].apply(lambda x: dias_semana[x])
df37 = df37.groupby(['Día', 'Año']).sum().unstack().fillna(0)
df37.plot(kind='bar', color=['darkorange','blue'], rot=45)
plt.grid(linestyle = '-.', alpha = 0.75, axis = 'y')
plt.gca().spines['right'].set_color('white')
plt.gca().spines['top'].set_color('white')
plt.legend(title=False, frameon=False, labels=['2019', '2020'])

```

```
plt.ylabel('Total de acciones')
plt.xlabel('Día de la semana')
plt.show()
```

```
df38 = df3.drop(['usuario', 'centro', 'semana', 't_acc', 'vistas_calif', 'num_mensajes_enviados'], axis=1)
df381 = df38.where(df38.fecha.dt.year == 2019)
df381["Nº inicios de sesión '19'"] = df381.sum(axis=1)
df381 = df381.groupby('hora').sum().astype('int')[['nis']]
df382 = df38.where(df38.fecha.dt.year == 2020)
df382["Nº inicios de sesión '20'"] = df382.sum(axis=1)
df382 = df382.groupby('hora').sum().astype('int')[['nis']]
plt.figure(figsize=(10,6))
plt.plot(df381.index, df381['nis'], color='b',linestyle='-', marker='o', label="2019")
plt.plot(df382.index, df382['nis'], color='darkorange', linestyle='-', marker='o', label="2020")
plt.xticks(list(df382.index), list(range(0,24)))
plt.gca().spines['right'].set_color('white')
plt.gca().spines['top'].set_color('white')
plt.grid(linestyle = '-.', alpha = 0.75)
plt.title('Nº inicios de sesión por hora')
plt.legend(title=False, frameon=True, fancybox=True, shadow=True)
plt.show()
```

```
df39 = df3.drop(['usuario', 'centro', 'semana', 'nis', 'vistas_calif', 'num_mensajes_enviados'], axis=1)
df391 = df39.where(df39.fecha.dt.year == 2019)
df391["Total de acciones '19'"] = df391.sum(axis=1)
df391 = df391.groupby('hora').sum().astype('int')[['t_acc']]
df392 = df39.where(df39.fecha.dt.year == 2020)
df392["Total de accionesn '20'"] = df392.sum(axis=1)
df392 = df392.groupby('hora').sum().astype('int')[['t_acc']]
plt.figure(figsize=(10,6))
plt.plot(df391.index, df391['t_acc'], color='b',linestyle='-', marker='o', label="2019")
plt.plot(df392.index, df392['t_acc'], color='darkorange', linestyle='-', marker='o', label="2020")
plt.xticks(list(df392.index), list(range(0,24)))
plt.gca().spines['right'].set_color('white')
plt.gca().spines['top'].set_color('white')
plt.grid(linestyle = '-.', alpha = 0.75)
plt.title('Total de acciones por hora')
plt.legend(title=False, frameon=True, fancybox=True, shadow=True)
plt.show()
```

## Anexo XI

Códigos empleados para el análisis del fichero de foros.

```
df4 = pd.read_csv('TFM/combinados/foros.csv', parse_dates=['fecha'])
df4
```

```
df41 = df4[df4.columns[6:]].where(df4.fecha.dt.year == 2019).sum().astype(int).to_frame()
df41.columns = ['2019']
df41['2020'] = df4[df4.columns[6:]].where(df4.fecha.dt.year == 2020).sum().astype(int).to_frame()
df41['Total'] = df4[df4.columns[6:]].where(df4.fecha.dt.year == 2019).sum().astype(int) + \
df4[df4.columns[6:]].where(df4.fecha.dt.year == 2020).sum().astype(int)
df41
```

```
plt.subplot(1,2,1)
df42 = df4['foros_creados'].where(df4.fecha.dt.year == 2019).groupby(df4['centro']).sum().to_frame()
labels = df42.where(df42.foros_creados != 0).dropna().index
explode = (0.05, 0.05)
plt.pie(np.array(df42.get(df42['foros_creados']!=0)).ravel(), labels = labels, explode = explode, autopct =
'%1.1f%%', \
    pctdistance = 1.3, shadow = True, labeldistance = 1.55)
plt.title('Foros creados en 2019',pad = 30.0)
plt.subplot(1,2,2)
df43 = df4['foros_creados'].where(df4.fecha.dt.year == 2020).groupby(df4['centro']).sum().to_frame()
labels = df43.where(df43.foros_creados != 0).dropna().index
explode = (0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05)
plt.pie(np.array(df43.get(df43['foros_creados']!=0)).ravel(), labels = labels, explode = explode, autopct =
'%1.1f%%', \
    pctdistance = 1.3, shadow = True, labeldistance = 1.55)
plt.title('Foros creados en 2020', pad=30.0)
plt.show()
```

```
plt.subplot(1,2,1)
df44 = df4['discusiones_creadas'].where(df4.fecha.dt.year ==
2019).groupby(df4['centro']).sum().to_frame()
labels = df44.index
explode = (0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05)
plt.pie(np.array(df44.get(df44['discusiones_creadas']!=0)).ravel(), labels = labels, explode = explode,
autopct = '%1.1f%%', \
    pctdistance = 1.3, shadow = True, labeldistance = 1.6)
plt.title('Discusiones creadas en 2019',pad = 30.0)
plt.subplot(1,2,2)
df45 = df4['discusiones_creadas'].where(df4.fecha.dt.year ==
2020).groupby(df4['centro']).sum().to_frame()
```

```

labels = df45.index
explode = (0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05)
plt.pie(np.array(df45.get(df45['discusiones_creadas']!=0)).ravel(), labels = labels, explode = explode,
autopct = '%1.1f%%', \
    pctdistance = 1.3, shadow = True, labeldistance = 1.6)
plt.title('Discusiones creadas en 2020', pad=30.0)
plt.subplots_adjust(wspace=0.5)
plt.show()

```

```

dias_semana = {0:'Lunes', 1:'Martes', 2:'Miercoles', 3:'Jueves', 4:'Viernes', 5:'Sabado', 6:'Domingo'}
df46 = df4[['discusiones_creadas', 'fecha']]
df46 = df46.set_index([df4['fecha'].dt.to_period('Y')], inplace = False)
df46.index.name = 'Año'
df46['Día'] = df46['fecha'].dt.dayofweek
df46['Día'] = df46['Día'].apply(lambda x: dias_semana[x])
df46 = df46.groupby(['Día', 'Año']).sum().unstack().fillna(0)
df46.plot(kind='bar', color=['darkorange', 'blue'], rot=45)
plt.grid(linestyle = '-.', alpha = 0.75, axis = 'y')
plt.gca().spines['right'].set_color('white')
plt.gca().spines['top'].set_color('white')
plt.legend(title=False, frameon=False, labels=['2019', '2020'])
plt.ylabel('N° de discusiones creadas')
plt.xlabel('Día de la semana')
plt.show()

```

```

plt.subplot(1,2,1)
df47 = df4['posts_creados'].where(df4.fecha.dt.year == 2019).groupby(df4['centro']).sum().to_frame()
labels = df47.where(df47.posts_creados != 0).dropna().index
explode = (0.05, 0.05, 0.05, 0.05)
plt.pie(np.array(df47.get(df47['posts_creados']!=0)).ravel(), labels = labels, explode = explode, autopct =
'%1.1f%%', \
    pctdistance = 1.3, shadow = True, labeldistance = 1.6)
plt.title('Posts creados en 2019', pad = 30.0)
plt.subplot(1,2,2)
df48 = df4['posts_creados'].where(df4.fecha.dt.year == 2020).groupby(df4['centro']).sum().to_frame()
labels = df48.where(df48.posts_creados != 0).dropna().index
explode = (0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05)
plt.pie(np.array(df48.get(df48['posts_creados']!=0)).ravel(), labels = labels, explode = explode, autopct =
'%1.1f%%', \
    pctdistance = 1.3, shadow = True, labeldistance = 1.6)
plt.title('Posts creados en 2020', pad=30.0)
plt.subplots_adjust(wspace=0.5)
plt.show()

```

```

dias_semana = {0:'Lunes', 1:'Martes', 2:'Miercoles', 3:'Jueves', 4:'Viernes', 5:'Sabado', 6:'Domingo'}
df49 = df4[['posts_creados', 'fecha']]
df49 = df49.set_index([df4['fecha'].dt.to_period('Y')], inplace = False)

```

```
df49.index.name = 'Año'
df49['Día'] = df49['fecha'].dt.dayofweek
df49['Día'] = df49['Día'].apply(lambda x: dias_semana[x])
df49 = df49.groupby(['Día', 'Año']).sum().unstack().fillna(0)
df49.plot(kind='bar', color=['darkorange', 'blue'], rot=45)
plt.grid(linestyle = '-.', alpha = 0.75, axis = 'y')
plt.gca().spines['right'].set_color('white')
plt.gca().spines['top'].set_color('white')
plt.legend(title=False, frameon=False, labels=['2019', '2020'])
plt.ylabel('Nº de posts creados')
plt.xlabel('Día de la semana')
plt.show()
```

## Anexo XII

Códigos empleados para el análisis del fichero con los recursos creados.

```
df5 = pd.read_csv('TFM/combinados/recursos_creados.csv', parse_dates=['fecha'])
df5
```

```
df51 = df5[df5.columns[6:]].where(df5.fecha.dt.year == 2019).sum().astype(int).to_frame()
df51.columns = ['2019']
df51['2020'] = df5[df5.columns[6:]].where(df5.fecha.dt.year == 2020).sum().astype(int).to_frame()
df51['Total'] = df5[df5.columns[6:]].where(df5.fecha.dt.year == 2019).sum().astype(int) + \
df5[df5.columns[6:]].where(df5.fecha.dt.year == 2020).sum().astype(int)
df51
```

```
Meses = {1: 'Enero', 2: 'Febrero', 3: 'Marzo', 4: 'Abril', 5: 'Mayo', 6: 'Junio', 7: 'Julio', 8: 'Agosto', \
9: 'Septiembre', 10: 'Octubre', 11: 'Noviembre', 12: 'Diciembre'}
df52 = df5.drop(columns=['n_bd', 'n_cuestionarios', 'n_glosarios', 'n_lecciones', 'n_talleres', 'n_tareas', \
'n_videos', 'n_wikis', 'n_etiquetas', 'n_carpetas'], inplace = False)
df52 = df52.drop(columns=['usuario', 'curso', 'hora', 'semana'])
df52['Año'] = df52.fecha.dt.year
df52['Mes'] = df52.fecha.dt.month
df52 = df52.sort_values('Mes')
df52['Mes'] = df52['Mes'].apply(lambda x: Meses[x])
df521 = df52.where(df52.fecha.dt.year == 2019).groupby(['Año', 'Mes']).sum()
df521[['n_archivos', 'n_urls']].plot(kind='bar', rot=45, tick_label=orden_meses)
plt.xticks([0,1,2,3,4], ['Abril', 'Febrero', 'Junio', 'Marzo', 'Mayo'])
plt.grid(linestyle = '-.', alpha = 0.75, axis = 'y')
plt.gca().spines['right'].set_color('white')
plt.gca().spines['top'].set_color('white')
plt.legend(title=False, frameon=True)
plt.xlabel('Meses')
plt.title('Recursos creados en 2019', pad=10.0)
plt.show()
```

```
df522 = df52.where(df52.fecha.dt.year == 2020).groupby(['Año', 'Mes']).sum()
df522[['n_archivos', 'n_urls', 'n_consultas', 'n_chats', 'n_szoom']].plot(kind='bar', rot=45,
tick_label=orden_meses)
plt.xticks([0,1,2,3,4], ['Abril', 'Febrero', 'Junio', 'Marzo', 'Mayo'])
plt.grid(linestyle = '-.', alpha = 0.75, axis = 'y')
plt.gca().spines['right'].set_color('white')
plt.gca().spines['top'].set_color('white')
plt.legend(title=False, frameon=True)
plt.xlabel('Meses')
plt.title('Recursos creados en 2020', pad=10.0)
```

```
plt.show()
```

```
Meses = { 1: 'Enero', 2: 'Febrero', 3: 'Marzo', 4: 'Abril', 5: 'Mayo', 6: 'Junio', 7: 'Julio', 8: 'Agosto', \
          9: 'Septiembre', 10: 'Octubre', 11: 'Noviembre', 12: 'Diciembre'}
df53 = df5.drop(['usuario', 'curso', 'centro', 'hora', 'semana', 'n_archivos', 'n_asistencia', 'n_carpetas', \
               'n_consultas', 'n_chats', 'n_encuestas', 'n_encuestas_pred', 'n_etiquetas', 'n_herr_ext', \
               'n_libros', 'n_paginas', 'n_szoom', 'n_urls'], axis=1)
df53['Año'] = df53.fecha.dt.year
df53['Mes'] = df53.fecha.dt.month
df53['Mes'] = df53['Mes'].apply(lambda x: Meses[x])
df531 = df53.where(df53.fecha.dt.year == 2019).groupby(['Año', 'Mes']).sum()
df531.plot(kind='bar', rot=45)
plt.xticks([0,1,2,3,4], ['Abril', 'Febrero', 'Junio', 'Marzo', 'Mayo'])
plt.grid(linestyle = '-.', alpha = 0.75, axis = 'y')
plt.gca().spines['right'].set_color('white')
plt.gca().spines['top'].set_color('white')
plt.xlabel('Meses')
plt.title('Actividades creadas en 2019', pad=10.0)
plt.legend(title=False, frameon=True, loc='upper left', ncol=2, fancybox=True, shadow=True)
plt.show()
```

```
df532 = df53.where(df53.fecha.dt.year == 2020).groupby(['Año', 'Mes']).sum()
df532.plot(kind='bar', rot=45)
plt.xticks([0,1,2,3,4], ['Abril', 'Febrero', 'Junio', 'Marzo', 'Mayo'])
plt.grid(linestyle = '-.', alpha = 0.75, axis = 'y')
plt.gca().spines['right'].set_color('white')
plt.gca().spines['top'].set_color('white')
plt.xlabel('Meses')
plt.title('Actividades creadas en 2020', pad=10.0)
plt.legend(title=False, frameon=True, loc='upper right', bbox_to_anchor=(1.175, 1), ncol=2,
fancybox=True, shadow=True)
plt.show()
```

```
df54 = df5.drop(columns=['usuario', 'curso', 'hora', 'semana', 'n_bd', 'n_cuestionarios', 'n_glosarios', \
                       'n_lecciones', 'n_talleres', 'n_tareas', 'n_videos', 'n_wikis'], inplace = False)
df54.groupby([(df54.fecha.dt.year, 'centro')]).sum()
```

```
df55 = df5.drop(['usuario', 'curso', 'hora', 'semana', 'n_archivos', 'n_asistencia', 'n_carpetas', \
               'n_consultas', 'n_chats', 'n_encuestas', 'n_encuestas_pred', 'n_etiquetas', 'n_herr_ext', \
               'n_libros', 'n_paginas', 'n_szoom', 'n_urls'], axis=1)
df55.groupby([(df55.fecha.dt.year, 'centro')]).sum()
```

```
df56 = df5.drop(['usuario', 'curso', 'centro', 'hora'], axis=1)
df561 = df56.drop(['n_bd', 'n_cuestionarios', 'n_glosarios', 'n_lecciones', 'n_talleres', 'n_tareas', \
```

```

        'n_videos', 'n_wikis'], axis=1).dropna()
df5611 = df561.where(df561.fecha.dt.year == 2019)
df5611["Total recursos '19'"] = df5611.iloc[:, 3:].sum(axis=1)
df5611 = df5611.groupby('semana').sum().astype('int')[["Total recursos '19'"]]
df5612 = df561.where(df561.fecha.dt.year == 2020)
df5612["Total recursos '20'"] = df5612.iloc[:, 3:].sum(axis=1)
df5612 = df5612.groupby('semana').sum().astype('int')[["Total recursos '20'"]]
plt.figure(figsize=(10,6))
plt.plot(df5611.index, df5611["Total recursos '19'"], color='b',linestyle='-', marker='o', label="2019")
plt.plot(df5612.index, df5612["Total recursos '20'"], color='darkorange', linestyle='-', marker='o',
label="2020")
plt.xticks(list(df5612.index), list(range(7,27)))
plt.gca().spines['right'].set_color('white')
plt.gca().spines['top'].set_color('white')
plt.grid(linestyle = '-.', alpha = 0.75)
plt.title("Total recursos creados por semana")
plt.legend(title=False, frameon=True, fancybox=True, shadow=True)
plt.show()

df55 = df5.drop(['usuario', 'curso', 'centro', 'hora'], axis=1)
df551 = df55[['fecha', 'semana', 'n_bd', 'n_cuestionarios', 'n_glosarios', 'n_lecciones', 'n_talleres', 'n_tareas',
\
        'n_videos', 'n_wikis']].dropna()
df5511 = df551.where(df551.fecha.dt.year == 2019)
df5511["Total actividades '19'"] = df5511.iloc[:, 3:].sum(axis=1)
df5511 = df5511.groupby('semana').sum().astype('int')[["Total actividades '19'"]]
df5512 = df551.where(df551.fecha.dt.year == 2020)
df5512["Total actividades '20'"] = df5512.iloc[:, 3:].sum(axis=1)
df5512 = df5512.groupby('semana').sum().astype('int')[["Total actividades '20'"]]
plt.figure(figsize=(10,6))
plt.plot(df5511.index, df5511["Total actividades '19'"], color='b',linestyle='-', marker='o', label="2019")
plt.plot(df5512.index, df5512["Total actividades '20'"], color='darkorange', linestyle='-', marker='o',
label="2020")
plt.xticks(list(df5512.index), list(range(7,27)))
plt.gca().spines['right'].set_color('white')
plt.gca().spines['top'].set_color('white')
plt.grid(linestyle = '-.', alpha = 0.75)
plt.title("Total actividades creadas por semana")
plt.legend(title=False, frameon=True, fancybox=True, shadow=True)
plt.show()

```

## Anexo XIII

Códigos empleados para el análisis del fichero de participación.

```
df6 = pd.read_csv('TFM/combinados/participacion.csv', parse_dates=['fecha'])
df6

df61 = df6[df6.columns[6:]].where(df6.fecha.dt.year == 2019).sum().astype(int).to_frame()
df61.columns = ['2019']
df61['2020'] = df6[df6.columns[6:]].where(df6.fecha.dt.year == 2020).sum().astype(int).to_frame()
df61['Total'] = df6[df6.columns[6:]].where(df6.fecha.dt.year == 2019).sum().astype(int) + \
df6[df6.columns[6:]].where(df6.fecha.dt.year == 2020).sum().astype(int)
df61

plt.subplot(1,2,1)
df621 = df6['ntars'].where(df6.fecha.dt.year == 2019).groupby(df6['centro']).sum().to_frame()
labels = df621.where(df621.ntars != 0).dropna().index
explode = (0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05)
plt.pie(np.array(df621.get(df621['ntars']!=0)).ravel(), labels = labels, explode = explode, autopct =
'%1.1f%%', \
    pctdistance = 1.3, shadow = True, labeldistance = 1.6)
plt.title('Nº tareas subidas en 2019',pad = 30.0)
plt.subplot(1,2,2)
df622 = df6['ntars'].where(df6.fecha.dt.year == 2020).groupby(df6['centro']).sum().to_frame()
labels = df622.where(df622.ntars != 0).dropna().index
explode = (0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05)
plt.pie(np.array(df622.get(df622['ntars']!=0)).ravel(), labels = labels, explode = explode, autopct =
'%1.1f%%', \
    pctdistance = 1.3, shadow = True, labeldistance = 1.6)
plt.title('Nº tareas subidas en 2020', pad=30.0)
plt.subplots_adjust(wspace=0.5)
plt.show()

plt.subplot(1,2,1)
df631 = df6['ncur'].where(df6.fecha.dt.year == 2019).groupby(df6['centro']).sum().to_frame()
labels = df631.where(df631.ncur != 0).dropna().index
explode = (0.05, 0.05, 0.05, 0.05, 0.05, 0.05)
plt.pie(np.array(df631.get(df631['ncur']!=0)).ravel(), labels = labels, explode = explode, autopct =
'%1.1f%%', \
    pctdistance = 1.3, shadow = True, labeldistance = 1.6)
plt.title('Nº cuestionarios \nrespondidos en 2019',pad = 30.0)
plt.subplot(1,2,2)
df632 = df6['ncur'].where(df6.fecha.dt.year == 2020).groupby(df6['centro']).sum().to_frame()
labels = df632.where(df632.ncur != 0).dropna().index
```

```

explode = (0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05)
plt.pie(np.array(df632.get(df632['ncur']!=0)).ravel(), labels = labels, explode = explode, autopct =
'%1.1f%%', \
    pctdistance = 1.3, shadow = True, labeldistance = 1.6)
plt.title('Nº cuestionarios \nrespondidos en 2020', pad=30.0)
plt.subplots_adjust(wspace=0.5)
plt.show()

```

```

df64 = df6.drop(['centro', 'hora', 'naz', 'nar', 'nac', 'npc', 'nag', 'npg', 'ntals', 'nvs', \
    'ncor', 'ner', 'ncur'], axis=1)
df641 = df64.where(df64.fecha.dt.year == 2019)
df641["Nº tareas subidas '19'"] = df641.sum(axis=1)
df641 = df641.groupby('semana').sum().astype('int')[['ntars']]
df642 = df64.where(df64.fecha.dt.year == 2020)
df642["Nº tareas subidas '20'"] = df642.sum(axis=1)
df642 = df642.groupby('semana').sum().astype('int')[['ntars']]
plt.figure(figsize=(10,6))
plt.plot(df641.index, df641['ntars'], color='b', linestyle='-', marker='o', label="2019")
plt.plot(df642.index, df642['ntars'], color='darkorange', linestyle='-', marker='o', label="2020")
plt.xticks(list(df642.index), list(range(7,27)))
plt.gca().spines['right'].set_color('white')
plt.gca().spines['top'].set_color('white')
plt.grid(linestyle = '-.', alpha = 0.75)
plt.title('Nº tareas subidas por semana')
plt.legend(title=False, frameon=True, fancybox=True, shadow=True)
plt.show()

```

```

df64 = df6.drop(['centro', 'hora', 'naz', 'nar', 'nac', 'npc', 'nag', 'npg', 'ntals', 'nvs', \
    'ncor', 'ner', 'ntars'], axis=1)
df641 = df64.where(df64.fecha.dt.year == 2019)
df641["Nº cuestionarios respondidos '19'"] = df641.sum(axis=1)
df641 = df641.groupby('semana').sum().astype('int')[['ncur']]
df642 = df64.where(df64.fecha.dt.year == 2020)
df642["Nº cuestionarios respondidos '20'"] = df642.sum(axis=1)
df642 = df642.groupby('semana').sum().astype('int')[['ncur']]
plt.figure(figsize=(10,6))
plt.plot(df641.index, df641['ncur'], color='b', linestyle='-', marker='o', label="2019")
plt.plot(df642.index, df642['ncur'], color='darkorange', linestyle='-', marker='o', label="2020")
plt.xticks(list(df642.index), list(range(7,27)))
plt.gca().spines['right'].set_color('white')
plt.gca().spines['top'].set_color('white')
plt.grid(linestyle = '-.', alpha = 0.75)
plt.title('Nº cuestionarios respondidos por semana')
plt.legend(title=False, frameon=True, fancybox=True, shadow=True)
plt.show()

```

## Anexo XIV

Códigos empleados para el análisis cluster en Stata.

- Código para agrupar los ficheros (Python):

```
for archivo in ['otras_acciones', 'foros', 'recursos_creados', 'participacion']:
    fichero = pd.read_csv('TFM/combinados/%s' % (archivo + '.csv'), parse_dates=['fecha'])
    if archivo == 'otras_acciones' or archivo == 'participacion':
        fichero_1 = fichero.where(fichero.fecha.dt.year ==
2020).groupby(fichero['centro']).sum().astype('int')
        fichero_1 = fichero_1.drop(fichero_1.columns[0:3], axis=1)
        fichero_1.to_csv('TFM/combinados/%s' % (archivo + '_agrupado_20.csv'), index=True)

    if archivo == 'foros' or archivo == 'recursos_creados':
        fichero_1 = fichero.where(fichero.fecha.dt.year ==
2020).groupby(fichero['centro']).sum().astype('int')
        fichero_1 = fichero_1.drop(fichero_1.columns[0:4], axis=1)
        fichero_1.to_csv('TFM/combinados/%s' % (archivo + '_agrupado_20.csv'), index=True)
```

- Código para otras acciones:

```
clear all
import delimited "C:\Users\Javier\TFM\combinados\otras_acciones_agrupado_20.csv"
encode centro, generate(n_centro)
pwcorr nis vistas_calif num_mensajes_enviados
cluster wardslinkage nis vistas_calif num_mensajes_enviados, measure(L2squared) name(wl)
cluster dendrogram wl, cutnumber(9) showcount
cluster stop wl, rule(duda)
cluster generate cluster_wl = groups(7), name(wl) ties(error)
tabulate cluster_wl, missing
tab n_centro cluster_wl
tabstat nis vistas_calif num_mensajes_enviados, statistics (mean) by (cluster_wl)
cluster kmeans nis vistas_calif num_mensajes_enviados, k(7) measure(L2squared) name(kmeans)
start(group(cluster_wl))
tabulate cluster_wl kmeans
tabstat n_centro, statistics (mean) by (cluster_wl)

cluster generate cluster_wl2 = groups(3), name(wl) ties(error)
tabulate cluster_wl2, missing
tab n_centro cluster_wl2
tabstat nis vistas_calif num_mensajes_enviados, statistics (mean) by (cluster_wl2)
cluster kmeans nis vistas_calif num_mensajes_enviados, k(3) measure(L2squared) name(kmeans2)
start(group(cluster_wl2))
tabulate cluster_wl2 kmeans2
tabstat n_centro, statistics (mean) by (cluster_wl2)
```

- Código para foros:

```
clear all
import delimited "C:\Users\Javier\TFM\combinados\foros_agrupado_20.csv"
encode centro, generate(n_centro)
pworth foros_creados_netos discusiones_creadas_netos posts_creados_netos
cluster wardslinkage foros_creados_netos discusiones_creadas_netos posts_creados_netos,
measure(L2squared) name(wl)
cluster dendrogram wl, cutnumber(9) showcount
cluster stop wl, rule(duda)
cluster generate cluster_wl = groups(2), name(wl) ties(error)
tabulate cluster_wl, missing
tab n_centro cluster_wl
tabstat foros_creados_netos discusiones_creadas_netos posts_creados_netos, statistics (mean) by
(cluster_wl)
cluster kmeans foros_creados_netos discusiones_creadas_netos posts_creados_netos, k(2)
measure(L2squared) name(kmeans) start(group(cluster_wl))
tabulate cluster_wl kmeans
tabstat n_centro, statistics (mean) by (cluster_wl)
cluster dendrogram wl, cutnumber(9) showcount
```

#### Código para recursos:

```
clear all
import delimited "C:\Users\Javier\TFM\combinados\recursos_creados_agrupado_20.csv"
encode centro, generate(n_centro)
pworth n_archivos n_asistencia n_consultas n_chats n_encuestas n_libros n_paginas n_szoom n_urls
cluster wardslinkage n_archivos n_asistencia n_consultas n_chats n_encuestas n_libros n_paginas
n_szoom n_urls, measure(L2squared) name(wl)
cluster dendrogram wl, cutnumber(9) showcount
cluster stop wl, rule(duda)
cluster generate cluster_wl = groups(2), name(wl) ties(error)
tabulate cluster_wl, missing
tab n_centro cluster_wl
tabstat n_archivos n_asistencia n_consultas n_chats n_encuestas n_libros n_paginas n_szoom n_urls,
statistics (mean) by (cluster_wl)
cluster kmeans n_archivos n_asistencia n_consultas n_chats n_encuestas n_libros n_paginas n_szoom
n_urls, k(2) measure(L2squared) name(kmeans) start(group(cluster_wl))
tabulate cluster_wl kmeans
tabstat n_centro, statistics (mean) by (cluster_wl)
```

- Código para actividades:

```

clear all
import delimited "C:\Users\Javier\TFM\combinados\recursos_creados_agrupado_20.csv"
encode centro, generate(n_centro)
pworth n_glosarios n_lecciones n_cuestionarios n_talleres n_tareas n_videos n_wikis
cluster wardslinkage n_glosarios n_lecciones n_cuestionarios n_talleres n_tareas n_videos n_wikis,
measure(L2squared) name(wl)
cluster dendrogram wl, cutnumber(9) showcount
cluster stop wl, rule(duda)
cluster generate cluster_wl = groups(2), name(wl) ties(error)
tabulate cluster_wl, missing
tab n_centro cluster_wl
tabstat n_glosarios n_lecciones n_cuestionarios n_talleres n_tareas n_videos n_wikis, statistics (mean) by
(cluster_wl)
cluster kmeans n_glosarios n_lecciones n_cuestionarios n_talleres n_tareas n_videos n_wikis, k(2)
measure(L2squared) name(kmeans) start(group(cluster_wl))
tabulate cluster_wl kmeans
tabstat n_centro, statistics (mean) by (cluster_wl)

```

- Código para participación:

```

clear all
import delimited "C:\Users\Javier\TFM\combinados\participacion_agrupado_20.csv"
encode centro, generate(n_centro)
pworth npc npg ntals ntars nvs
cluster wardslinkage npc npg ntals ntars nvs, measure(L2squared) name(wl)
cluster dendrogram wl, cutnumber(9) showcount
cluster stop wl, rule(duda)
cluster generate cluster_wl = groups(5), name(wl) ties(error)
tabulate cluster_wl, missing
tab n_centro cluster_wl
tabstat npc npg ntals ntars nvs, statistics (mean) by (cluster_wl)
cluster kmeans npc npg ntals ntars nvs, k(5) measure(L2squared) name(kmeans) start(group(cluster_wl))
tabulate cluster_wl kmeans
tabstat n_centro, statistics (mean) by (cluster_wl)

```