UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf
Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Desarrollo de una herramienta para el análisis de rendimiento del alumnado del Grado en Ingeniería Informática de la ETSINF.

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Artur de Osset Greño

Tutores: César Ferri Ramírez y Antonio Molina Marco

Curso: 2018-2019

Agradecimientos

A mi familia, por todo su apoyo y paciencia durante estos años.

A mis tutores, César y Antonio, por su guía en este proyecto.

Pedro Pablo del Servicio de Evaluación, Planificación y Calidad de la UPV por los datos sobre los que se trabaja.

Gracias.

Resumen

Con el paso de los años se han matriculado cientos de estudiantes de toda índole, con no menos variedad de resultados, en el grado de ingeniería informática de la ETSINF y muchos más lo harán en el futuro. De todos se guardan en bases de datos cierta información en el momento de su ingreso y toda su trayectoria dentro de la universidad.

En este proyecto se ha desarrollado una herramienta web pensada para facilitar el análisis de toda esa información centrándose en el rendimiento, permitiendo focalizarse en las asignaturas o el alumnado, el filtrado de datos, incluso realizar ciertas predicciones.

Se ha documentado todo el proceso, desde la elección del lenguaje de programación utilizado hasta un análisis posterior de los resultados facilitados por la aplicación, pasando por las diferentes fases de desarrollo.

Palabras clave: r, análisis de datos, rendimiento.

Abstract

Over the years hundreds of students of all kinds have enrolled, with no less variety of results, in the computer engineering degree of the ETSINF and many more will do so in the future. Of all of them, certain information is kept in databases at the time of admission and their entire trajectory within the university.

In this project, a Web tool has been developed, designed to facilitate the analysis of all this information, focusing on performance, allowing targeting on the subjects or students, filtering data, and even making certain predictions.

The entire process has been documented, from the choice of the programming language used to a subsequent analysis of the results provided by the application, going by the different phases of development.

Keywords: r, data analysis, performance.

Tabla de contenidos

Capítulo 1: Introducción

Por La Escola Tècnica Superior d'Enginyeria Informàtica de la UPV pasan cada año unos pocos cientos de nuevos estudiantes. Unos provienen de bachillerato, otros de cursos formativos. Unos son hombres, otras mujeres. Unos acabarán la carrera en cuatro años mientras que otros tardarán más y probablemente muchos no lo harán nunca. Y todos rendirán de forma diferente.

Todo esto puede llevar a realizarse algunas preguntas ¿Hay una diferencia en notas entre un alumno que haya cursado un ciclo formativo y uno que provenga directamente de bachillerato? No hace falta más que observar las aulas para ver lo evidente que es la mayoría de hombres que hay frente a las mujeres en este grado, ¿Qué porcentaje exacto es? ¿Disminuye o aumenta con el paso de los años? ¿Hay una diferencia entre el rendimiento de estos dos grupos? ¿Cuáles asignaturas son las que más se suspenden? ¿Con que proporción exacta? ¿Los alumnos matriculados un año son, en términos generales, mejores estudiantes que otro? Un alumno que destaque en ciertas asignaturas en los primeros cursos, ¿a qué rama o ramas podría decantarse? Y se podría seguir así eternamente.

Los datos estaban ahí, con toda la información necesaria para responder a muchas más que a estas preguntas. Este TFG no se encarga de responder a esas cuestiones, pero sí de crear una herramienta que permita, al menos, facilitar encontrar las respuestas.

Motivaciones

Las motivaciones son diversas. Por un lado, algunas alternativas, proyectos sin duda interesantes y prácticos, consistían en elaborar módulos o mejoras sobre aplicaciones ya existentes, una tarea nada desdeñable. Sin embargo, la posibilidad de la creación de un proyecto completo, desde su ideación hasta su fin, resultaba un desafío más acorde con lo que se buscaba, ya que fomentaría y mejoraría las habilidades de gestión y planificación.

Por otro lado, la ciencia de datos, que se define como "un concepto para unificar estadísticas, análisis de datos, aprendizaje automático y sus métodos relacionados para comprender y analizar los fenómenos reales", está en auge en estos últimos años, siendo buscados cada vez más expertos informáticos en este campo. Este proyecto era una oportunidad perfecta para comenzar a profundizar en ese mundo, obteniendo pericia en herramientas y técnicas propias del campo.

Por último, hablando de herramientas, las "armas" de un programador son los lenguajes que conoce y aunque no había en origen un lenguaje escogido, las alternativas eran principalmente uno desconocido y uno con poca experiencia en su

manejo. Aprender uno o aumentar el dominio del otro aumentaría las expectativas laborales en un futuro.

Objetivos

El objetivo principal y final de este TFG es, como su propio nombre indica, desarrollar una herramienta que permita a los gestores de una titulación un análisis del rendimiento de su alumnado. Este objetivo general se puede desglosar en los siguientes objetivos específicos:

- Conocer la evolución de los egresados de una titulación en relación a su rendimiento a lo largo de los estudios según distintos parámetros configurables (sexo, edad, cohorte, etc.).
- Realizar agrupamientos y correlaciones bajo diversos parámetros.
- Realizar predicciones sobre rendimientos futuros o posibles itinerarios de los estudiantes.

Estructura de la memoria

Para finalizar la introducción se va a hacer un repaso al resto de secciones que forman esta memoria.

Análisis tecnológico

Para realizar este proyecto era necesario elegir primero un lenguaje de programación adecuado para sus necesidades concretas. En este aparatado se procede a documentar el proceso de comparación y elección del lenguaje utilizado, así como un breve repaso por el proceso de aprendizaje seguido.

Análisis de los datos

Tras haber seleccionado un lenguaje se recibió los datos, en un formato adecuado, sobre los que se realizaría toda la aplicación. En este apartado se procede a hacer un análisis de estos datos, detallando su composición y características.

Alcance de la aplicación

Una vez se controla el entorno de desarrollo y se tiene conocimiento sobre los datos con los que trabajar, en esta sección se procede a especificar una lista de funcionalidades deseadas para la aplicación y demás ideas que, aunque tal vez no se piensen implementar en un principio, puedan ser opciones viables de expansión.

Desarrollar la aplicación

Habiendo realizado los puntos anteriores llega el momento de poner en práctica lo aprendido y utilizando el diseño, comenzar a programar. En esta sección se documentó todo el proceso de desarrollo de la aplicación.

Analizar los resultados de la aplicación

Una vez desarrollada la aplicación se procede a utilizarla para analizar los datos de los alumnos. En esta sección se pretende comprobar su correcto funcionamiento y utilidad, estudiando casos concretos con datos reales.

Conclusiones

Tras el completo proceso de desarrollo y análisis llega el momento de recapitular y comprobar los resultados. Ese es el objetivo de esta sección.

Capítulo 2: Análisis tecnológico

Este proyecto en concreto ofrece una cierta flexibilidad pues consiste en desarrollar una herramienta de análisis estadístico, habiendo flexibilidad en los detalles, tales como el lenguaje utilizado. Actualmente existen varias alternativas de lenguajes potentes en el campo del proyecto, así que se realizó un breve estudio de las opciones disponibles seleccionando la que resultó más adecuada.

Python

El primer lenguaje de programación planteado para la realización del proyecto fue Python[1], una sólida herramienta con la que se tiene cierta experiencia debido a que se utilizaba en la asignatura Sistemas de almacenamiento y recuperación de información (SAR) y Algorítmica (ALG), ambas de la rama Computación. Llamado así por el famoso grupo británico humorístico "Monty Python", este lenguaje de código abierto es flexible, permitiendo utilizar varios paradigmas de programación, multiplataforma por lo que un código escrito en Windows funcionara sin problemas en Mac o Linux y es altamente legible comparado con otros lenguajes como C++. La idea bajo la que nació entre finales de los años ochenta y a principio de los noventa fue crear un lenguaje fácil de aprender y de usar manteniendo una gran capacidad de procesado.

Si bien es cierto que Python como tal no esta específicamente desarrollado para el uso estadístico y el manejo de datos como otros, existen varias librerías ideadas para este fin. Algunas de las más conocidas son:

- **Numpy**: abreviatura de Numerical Python, es un paquete fundamental ya que incluye una gran variedad de herramientas para facilitar los cálculos científicos, tales como matrices más eficientes y operaciones sobre ellas.
- **Matplotlib**: introduce una gran variedad de gráficos a Python permitiendo una visualización y exploración de datos mayor.
- **IPython**: una poderosa herramienta que permite una mayor interacción, sobre todo en su vertiente web.
- **Pandas**: integra unas estructuras de datos (llamados *data frames* en inglés) fáciles de utilizar, con un manejo parecido al de bases de datos como SQL, optimizadas para un alto rendimiento.

Gracias a estas extensiones Python se convierte en una alternativa muy a tener en cuenta cuando lo que se desea es realizar análisis de datos y estadísticas.

[1] https://www.python.org/

R

R fue lanzado en el año 1995, como una herramienta para estadísticas y modelos gráficos. Si Python es un lenguaje de uso general que puede ser empleado en análisis de datos, R[2] fue creado desde el principio pensando totalmente en ese aspecto. Así como Python necesita de módulos externos para realizar los cálculos pertinentes de un modo destacable, R incluye todas esas características, necesitando módulos externos en pocos casos, solamente para operaciones muy concretas. Pese a todo, R no es un lenguaje tan popular como otros, mucho menos tanto como Python, el más buscado del año 2018 (según PYPL)[3] pero es algo lógico puesto que el campo de trabajo de R es exclusivo de la ciencia de datos y Python es multipropósito.

Además, R posee también librerías especializadas en campos concretos de la ciencia de datos que permiten realizar de un modo muy sencillo algunas tareas arduas. Algunos ejemplos son:

- **Caret**: librería que permite de un modo sencillo y accesible realizar predicciones de modelos.
- **PerformanceAnalytics**: introduce herramientas para realizar investigaciones en datos como correlaciones.
- **Shiny**: añade toda una capa web de manejo sencillo traduciendo ordenes concretas a lenguaje HTML.

R más estas extensiones permiten una gran capacidad de análisis, reduciendo complicados algoritmos a escasas líneas de código.

SAS

Otro lenguaje digno de mención es SAS. Desarrollado en los años sesenta por *SAS Insitute*[4] fue concebido para computarizar los datos estadísticos del entorno agrícola, ya que por aquel entonces no había ninguna herramienta informática enfocada en ese campo. Desde su modesto comienzo el lenguaje ha ido creciendo y evolucionando hasta convertirse en un líder de la industria analítica llegando a ocupar una cuota de mercado del 30,8% en analítica avanzada y predictiva en 2017[5].

Especializado en toda clase de operaciones sobre tablas de datos: leerlas, modificarlas, combinarlas, filtrarlas, obtener informes en base a ellas… se adapta perfectamente a los requisitos solicitados a un lenguaje en un proyecto de estas características.

[2] https://www.r-project.org/
[3] https://adtmag.com/articles/2019/01/08/tiobe-jan-2019.aspx
[4] https://www.sas.com/es_es/home.html
[5] https://www.lavanguardia.com/vida/20181001/452117057581/economiaempresas--sas-logra-una-cuota-de-mercado-del-308-en-analitica-avanzada-y-predictiva-en-2017.html

Sin embargo, el lenguaje SAS tiene una pega que lo ha descartado como una de las opciones para este TFG, no es de código abierto, por tanto, hubiese sido necesario comprar una licencia de uso.

R VS Python

Después de analizar cada uno de los lenguajes se ha obtenido suficiente información para hacer una comparación de ambos, centrada en unos campos de especial interés.

Aprendizaje

Python es más parecido a otros lenguajes de programación como Java, además el hecho de haberlo utilizado con anterioridad reduce el proceso de aprendizaje a familiarizarse con los paquetes necesarios para hacer este tipo de análisis.

Por su parte, de R se dice que es más fácil de aprender, incluso para gente ajena al mundo de la programación pues tiene mucha semejanza con la terminología estadística.

Manejo de datos

R es más rápido haciendo análisis estadísticos, al fin y al cabo, para eso fue diseñado, sin embargo, si se utiliza como un lenguaje de programación tradicional dará peores resultados. Además, hay que tener en cuenta que trabaja con los datos almacenados en la memoria del ordenador y esto puede ralentizarlo en caso de conjuntos de datos de gran tamaño.

Gráficos

Para visualizar correctamente los resultados de un análisis de datos es recomendable su representación gráfica, donde de nuevo destaca R ya que incluye unas graficas decentes en su código base y unas excelentes en paquetes (ggplot2, plotly).

Aun así, Python se defiende en este campo gracias a librerías como la mencionada anteriormente matplotlib.

Perspectivas laborables

Otro aspecto a tener en cuenta es la salida laboral, pues la decisión de aprender un nuevo lenguaje abrirá unas puertas u otras. Python, como ya he repetido en varias ocasiones, es un lenguaje multipropósito por lo que abre un espectro más amplio de posibilidades.

R por su parte es más especializado por lo tanto es más solicitado, sobre todo en los últimos tiempos, en campos como el de la estadística, como se puede observar en la ilustración 1 que muestra los porcentajes de trabajos por lenguajes ofertados de la página *indeed*.

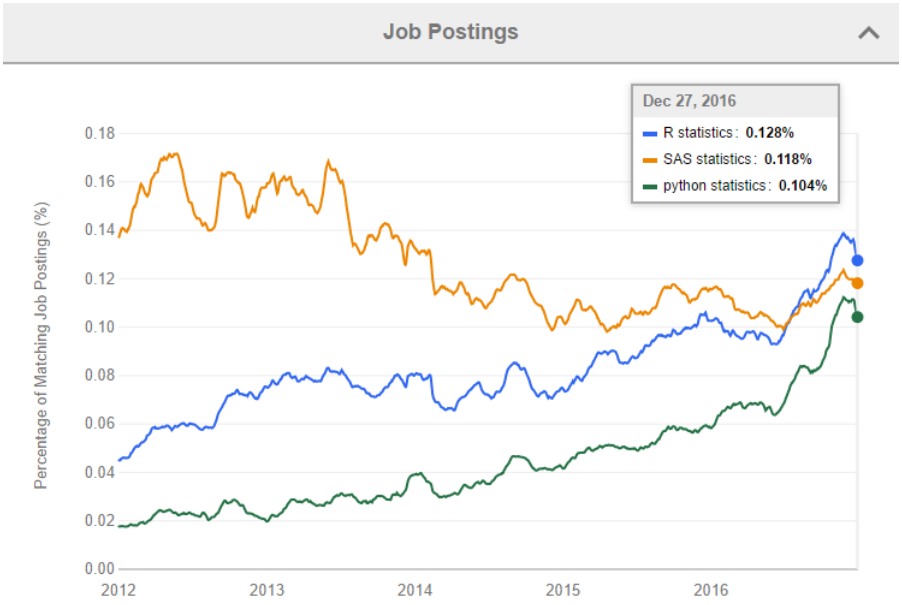


Ilustración 1 Porcentaje trabajos por lenguajes ofertados de estadistica

Resultado

Sopesando detenidamente las opciones al final el lenguaje seleccionado es R. Ambos son muy similares en características, sin embargo, R destaca más en el campo de este proyecto, el estudio de datos, además aprender un lenguaje completamente nuevo ofrece la oportunidad de ampliar mis conocimientos y aumentar mis habilidades, en este caso en un campo especialmente en auge como es la ciencia de datos.

Método de aprendizaje

El lenguaje seleccionado ha acabado siendo R, del que se era completamente ajeno, por tanto, antes de planificar una aplicación que acabe resultando imposible de programar en R debido a sus características internas, es preciso dominarlo con cierta soltura.

Cuando se comienza a aprender un nuevo lenguaje de programación a veces puede resultar un poco abrumador, hay multitud de opciones para hacerlo y, como en todo, hay opciones mejores y peores. Se procede a hacer un breve repaso de ellas señalando porque se desestimaron o no cada una.

Empezar a programar directamente

Empezar cualquier clase de tarea sin preparación suele ser una mala idea y en este caso no es de otro modo. Es cierto que a una persona con conocimientos previos de programación puede resultarle más sencillo que a alguien ajeno a ese mundo, pero sigue sin ser una buena opción ya que al desconocer los entresijos de ese lenguaje en concreto se puede caer con facilidad en mala praxis.

Aprender con un conocido

Si se conoce a alguien que domine el idioma se le puede pedir ayuda, siempre y cuando tenga el tiempo y la capacidad didáctica requerida, así como este de acuerdo con el trato. Lamentablemente, son unas características muy concretas, siendo difícil que se den todas, como es este caso.

Realizar un curso sobre el lenguaje

Recurrir a un curso es una opción perfectamente válida habiendo multitud de ellos, desde presenciales a online. El curso a seleccionar dependería de las circunstancias personales de cada uno valorando que ofrecen y si es lo que se busca. En este caso, estando estudiando y en prácticas de empresa, se valoraba algo más de flexibilidad por lo que no se tomó como primera alternativa, aunque seguía siendo una posibilidad.

Utilizando una web didáctica

Aparte de los cursos académicos antes mencionados existen gran variedad de webs de confianza para introducirse al mundo de la programación, tales como w3schools[6], que no solo incluye una gran variedad de guías para HTML, Javascript o Python si no que se puede hacer exámenes en ella para obtener certificados. Sin embargo, para R no se encontró ninguna que resultara especialmente convincente.

Utilizando un libro o manual

Si existen cursos y webs para aprender a programar también hay gran cantidad de libros para el mismo fin. En el caso de R hay multitud de libros que enseñan su uso, y lo mejor de todo es que se puede acceder a su versión online de modo gratuito. Habiendo aprendido otros lenguajes de este modo con resultados satisfactorios y vistas las facilidades ofrecidas, esta fue la opción elegida.

Aprendizaje

De entre los libros disponibles sobre R el elegido fue ***R for Data Science*** que introduce el lenguaje de un modo ameno. Además, explica el funcionamiento de su principal entorno de desarrollo ***RStudio.***

Conforme se fue profundizando en el conocimiento de R se recurrieron a otros libros de áreas más concretas del lenguaje. Algunos de estos libros fueron: ***Interactive web-based data visualization with R, plotly, and shiny*** y ***advanced R*** utilizados más como material de consulta o el "curso" online de la librería ***shiny***.

Shiny, como se comentó en la comparativa con Python, es una librería para R que permite crear aplicaciones web interactivas de un modo muy sencillo. Durante la etapa de aprendizaje se eligió como base para implementar la aplicación por su potencia y facilidad de uso.

[6] https://www.w3schools.com/

Este paquete permite una gestión intuitiva de la aplicación separando sus componentes según pertenecen a la interfaz gráfica o al servidor, los objetos *input* y *output* respectivamente, relacionando ambos gracias a una serie de elementos que permiten controlar los distintos eventos, como se puede ver en la ilustración 2.

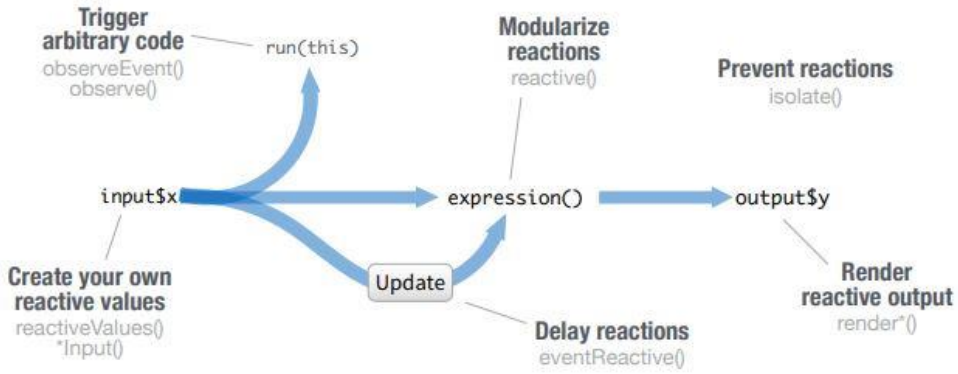


Ilustración 2 Esquema de eventos en shiny

Por último, shiny cuenta con una plataforma, **shinyapps**, que facilita desplegar las aplicaciones creadas con su librería permitiendo controlar su acceso público, la cantidad de memoria asignada a su ejecución… algunas de estas opciones siendo solo accesibles en planes de pago, aunque existe un plan gratuito que es suficiente para la aplicación desarrollada en este proyecto.

En la página web de shiny[7] se pueden encontrar múltiples ejemplos funcionales de aplicaciones desarrolladas con esta herramienta, tres de ellas se pueden apreciar en las ilustraciones 3, 4 y 5.

[7] https://shiny.rstudio.com/

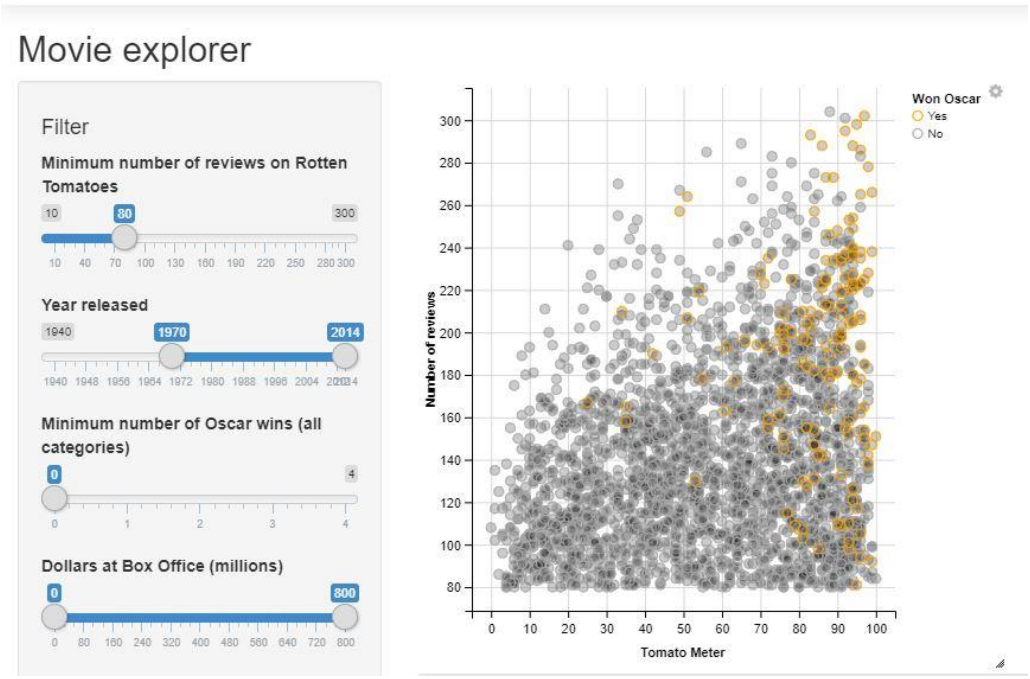


Ilustración 3 Ejemplo 1 de aplicación creada con shiny

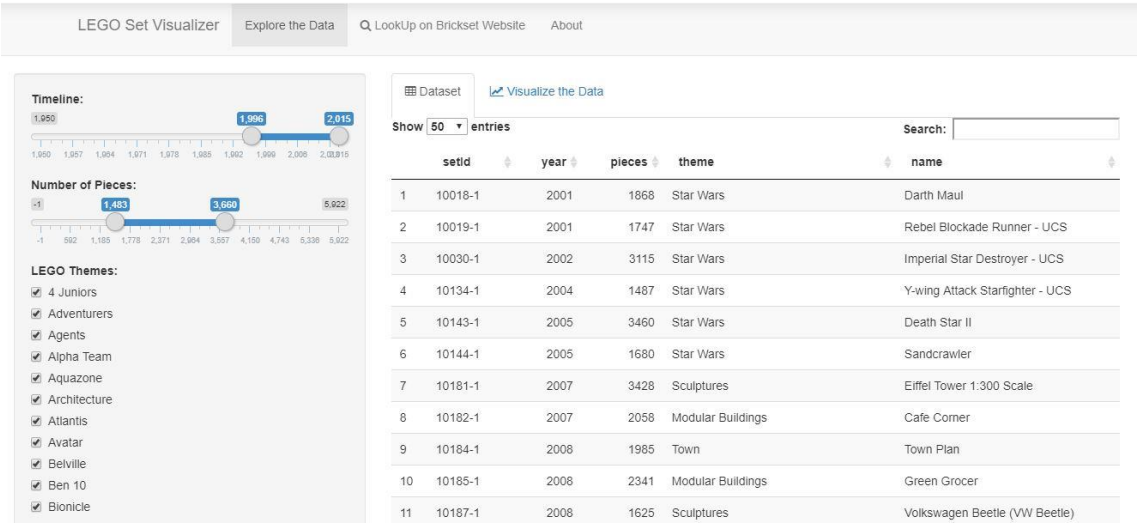


Ilustración 4 Ejemplo 2 de aplicación creada con shiny

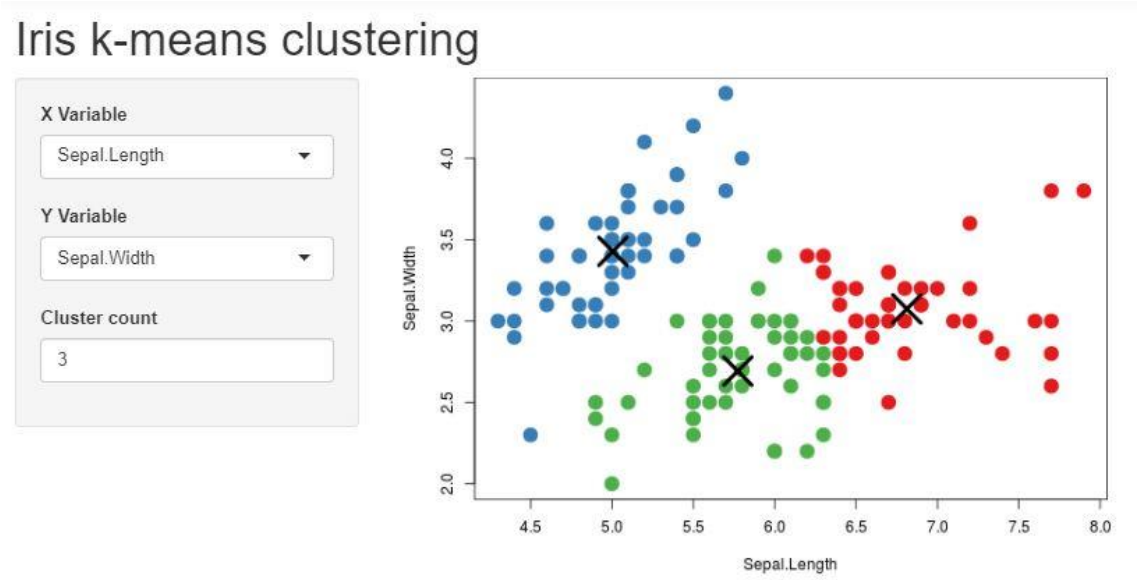


Ilustración 5 Ejemplo 3 de aplicación creada con shiny

Capítulo 3: Análisis de los datos

Una vez familiarizado con R el siguiente paso es analizar las funcionalidades de la aplicación. Pero para saber qué se podía hacer era preciso antes estudiar los datos de los que se disponía.

Análisis

Inicialmente se disponía de dos *data frames* (estructuras de datos en R): **titulados_ofu** y **asignaturas_ofu**.

El primer data frame, **titulados_ofu**, posee información completamente anónima sobre los alumnos matriculados en del Grado en Ingeniería Informática de la ETSINF a partir de 2010 (inclusive) que la han **completado satisfactoriamente** hasta el año 2018 (también inclusive).

Este data frame contiene datos de 999 alumnos repartidos en 56 columnas. Como se puede apreciar, eso es mucha información, sin embargo, no toda se va a utilizar pues hay información redundante o que no se le encuentra un uso adecuado en la aplicación.

Algunos de sus campos destacados son **ANYCOM**, **ANYFIN** que son el año de comienzo y fin respectivamente, **CACA_PROJ** que indica el curso académico del TFG, coincidiendo siempre con el año de fin, siendo uno de los campos redundantes, **MEDIA_EXP** y **MEDIA_OFICIAL** que son la media del alumno en el curso (campos redundantes de nuevo), **NOTA_PROJ** que se trata de la nota del TFG, varios campos de la cantidad de créditos matriculados, siendo poco útiles debido a que todos los alumnos han acabado el grado, **GENERO** que evidentemente indica el género del alumno (V o M), **ING_ESTUDIOS** que son los estudios que tiene el alumno al llegar al grado, **ING_NOTA** que indica nota de ingreso, **RESI_F** que es residencia familiar, **OBTIENE_BECA** que señala si el alumno ha obtenido una beca y **EDAD_31_12_ING** que señala la edad del alumno en el 31 de diciembre del año que ingresó.

Entre todos los datos los señalados en el párrafo anterior serán aquellos que se utilicen, seleccionando solo uno cuando haya redundancia. R, como la potente herramienta de análisis que es, dispone de la orden *summary(…)* que muestra información de los datos que se le pasen. Se ejecuta *summary(…)* sobre los datos y se observa qué nos indica de los campos a utilizar.

Antes de pasar al análisis se explica qué muestra esta instrucción de los diversos campos. De los numéricos se muestran los valores mínimos y máximos, la media, la mediana, los cuartiles primero y tercero y el número de valores no definidos, si los hay. Para los campos con valores no numéricos se muestra las ocurrencias de cada valor.

ANYCOM	ANYFIN	MEDIA_EXP	NOTA_PROJ	GENERO
Min. :2010	Min. :2013	Min. :5.700	Min. : 5.000	M:105
1st Qu.:2011	1st Qu.:2014	1st Qu.:6.500	1st Qu.: 8.000	V:894
Median :2012	Median :2015	Median :6.900	Median : 9.000	
Mean :2012	Mean :2015	Mean :7.114	Mean : 8.733	
3rd Qu.:2013	3rd Qu.:2016	3rd Qu.:7.600	3rd Qu.:10.000	
Max. :2017	Max. :2018	Max. :9.900	Max. :10.000	

Ilustración 6 Estadísticas titulados_ofu 1

Observando los años de comienzo y fin resulta sorprendente que haya alumnos matriculados en 2017 y otros que hayan finalizado en 2013, pues deben de haber finalizado el grado en entre uno y tres años, sin embargo, esto puede ser debido a estudiantes que realizaron el curso para adaptar la titulación del alumno al grado. Los campos MEDIA_EXP y NOTA_PROJ son aproximadamente contrarios entre ellos, el primero con notas bajas y el segundo con altas. En GENERO se aprecia que las alumnas solamente representan un 11,74% del total del alumnado.

ING_ESTUDIOS	RESI_F	OBTIENE_BECA_1	EDAD_31_12_ING
PAU :600	ComVal : 59	-:392	Min. :18.0
Cic. Form.:172	Extr : 2	N:240	1st Qu.:18.0
Ext. UE : 14	ProvVal:522	S:367	Median :19.0
>25 : 7	RestEsp: 89		Mean :21.6
Titulados : 2	Val :327		3rd Qu.:22.0
(Other) : 2			Max. :54.0
NA's :202			

Ilustración 7 Estadísticas titulados_ofu 2

En estos otros campos se aprecia que los estudiantes más comunes son aquellos provenientes de la PAU, seguidos de los del ciclo. En cuanto a la residencia se aprecia que hay más alumnos provenientes de la provincia de valencia seguidos de aquellos de la propia ciudad de valencia. En el campo de beca la mayoría la tienen aquellos que no han pedido beca seguido por los que la han solicitado y recibido. En cuanto a la edad es evidente que la mayoría de los alumnos son jóvenes, siendo los mayores de 22 años una minoría.

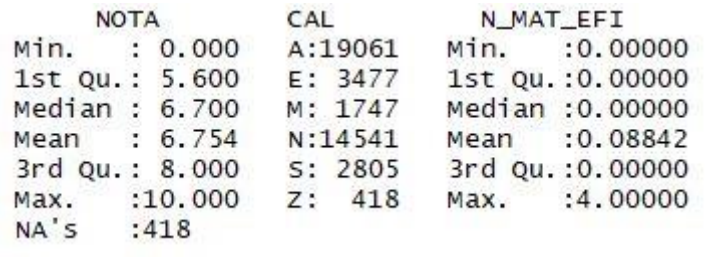
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
5.140	7.111	8.125	8.366	9.552	13.900	202

Ilustración 8 Estadísticas titulados_ofu nota de ingreso

Por último, se estudia la nota de ingreso, que llega como una cadena de texto y debe ser convertido a numérico para su correcto uso. Como este campo es sobre 14 los resultados siguen una distribución más o menos esperada.

El segundo data frame, **asignaturas_ofu**, recoge los datos de la calificación de un alumno cada vez que se presenta a una asignatura. En este conjunto de datos hay 42049 registros distribuidos en 14 columnas.

Los campos más importantes son: **ASI** que es el código de asignatura, **ID** que es el código del alumno, **CACA** que indica el curso académico, **NOTA** que, evidentemente, indica la nota sacada por el alumno en la asignatura, **CAL** que es la calificación obtenida en la asignatura (suspendido, aprobado, notable, excelente, matrícula y no presentado) y **N_MAT_EFI** que señala cuantas veces se había matriculado el alumno en la asignatura (0 para la primera vez, 1 para la segunda, etc.).



NOTA		CAL		N_MAT_EFI	
Min.	: 0.000	A:	19061	Min.	:0.00000
1st Qu.	: 5.600	E:	3477	1st Qu.	:0.00000
Median	: 6.700	M:	1747	Median	:0.00000
Mean	: 6.754	N:	14541	Mean	:0.08842
3rd Qu.	: 8.000	S:	2805	3rd Qu.	:0.00000
Max.	:10.000	Z:	418	Max.	:4.00000
NA's	:418				

Ilustración 9 Estadísticas de asignaturas_ofu

Sobre las notas de las asignaturas se puede apreciar que entre los alumnos de los que se tiene información, aquellos que han finalizado el grado, su rendimiento es bastante bueno, pues el número de no presentados (Z en CAL o NA'S en NOTA) es prácticamente despreciable (sobre el 1%) y el número de notas excelentes es superior al número de suspendidas.

También había ciertos campos de la naturaleza de la asignatura, como es el curso al que pertenece o la cantidad de créditos de la asignatura, sin embargo, carece de un campo muy importante como es el nombre de la asignatura.

Más tarde se le añadió un tercer data frame con la información de las asignaturas que faltaba, **asingaturas_info**, que incluye el **código**, el **nombre**, el número de **créditos**, el **semestre** y el **bloque** (1, 2, 3, cada una de las distintas ramas y optativa), para cada una de las 134 asignaturas.

Capítulo 4: Alcance de la aplicación

Habiendo seleccionado el lenguaje a utilizar y conociendo los datos a manejar llega el momento de definir el alcance de la aplicación. Para ello primero se definen los requisitos y luego, sobre ellos se diseñan los distintos módulos o pestañas.

Requisitos

Los requisitos de este proyecto se definen condicionados por los objetivos específicos enumerados en el primer capítulo.

Del primer subobjetivo propuesto, "Conocer la evolución de los egresados de una titulación en relación a su rendimiento a lo largo de los estudios según distintos parámetros configurables (sexo, edad, cohorte, etc.)", se extraen las siguientes funcionalidades:

- Un modo de mostrar las notas de los estudiantes.
- Filtrar por diversos campos.
- Cálculo de estadísticas sobre el alumnado.

"Realizar agrupamientos y correlaciones bajo diversos parámetros", el segundo subobjetivo, exige los siguientes requisitos:

- Permitir separar los datos según ciertos criterios.
- Mostrar información de cada grupo.
- Seleccionar los datos a utilizar para estudiar las correlaciones entre ellos.
- Generar correlaciones.

Para el último de los subobjetivos, "Realizar predicciones sobre rendimientos futuros o posibles itinerarios de los estudiantes", es preciso implementar ciertas funcionalidades:

- Seleccionar que datos utilizar para la predicción.
- Poder introducir los datos sobre los que realizar la predicción.
- Permitir realizar la predicción.

Para finalizar los requisitos se añaden una serie de estipulaciones no funcionales para toda la aplicación, como son:

- Manejo sencillo.
- Diseño modular para facilitar la adición de nuevas funcionalidades.
- Independencia de los datos.
- Facilidad de adaptación a otros contextos (titulaciones).

Diseño

Una vez definidos los distintos requisitos se procede a definir los diferentes módulos de la aplicación. Los módulos se crearían mediante agrupaciones de requisitos.

El primer módulo se diseñó para mostrar datos de alumnos separados por asignaturas. Este módulo cumple los requisitos: *mostrar las notas de los estudiantes*, *filtrar por diversos campos* y *cálculo de estadísticas sobre el alumnado*, *además de separar los datos en grupos*, en este caso asignaturas. Permitirá cambiar de asignatura, mostrará las notas de la asignatura (un diagrama de barras o un histograma serían lo más idóneo) filtrando los alumnos según diversos criterios y se complementara con las estadísticas adecuadas. Esta será la pestaña Asignaturas.

Se diseña un segundo modulo para realizar comparaciones de estudiantes según campos distintos campos (genero, edad, estudios previos…). Este módulo cumple los requisitos: *permitir separar los datos según ciertos criterios* y *mostrar información de cada grupo*. Una gráfica de tarta permitiría comparar el tamaño de los grupos mientras una tabla mostraría más información de cada uno de ellos. Esta es la pestaña Comparaciones.

Otro módulo estaría totalmente focalizado en los requisitos sobre correlaciones, es decir, *seleccionar los datos a utilizar para estudiar las correlaciones entre ellos* y *generar correlaciones*. En esta pestaña llamada Correlaciones, una gráfica de correlaciones sería el elemento principal, seleccionando las asignaturas que aparecen con algún método aún por decidir, tal vez un *select múltiple*.

Un cuarto módulo realizaría comparaciones sobre alumnos, pero esta vez centrándose en su rendimiento. Los requisitos que cumpliría son los mismos que la segunda, lo que cambiaría es la separación de los grupos. Este módulo crearía grupos en función de sus notas, seleccionando el número de grupos y mostrando datos en una tabla. Debido al nombre común de estos grupos en el mundo estadístico, *clusters*, está pestaña se llamará Clustering.

El quinto y último modulo se diseñó para satisfacer los requisitos de las predicciones: s*eleccionar que datos utilizar para la predicción*, *poder introducir los datos sobre los que realizar la predicción* y *permitir realizar la predicción*. En primer lugar, se predeciría la rama por la que se decantaría el alumno dependiendo de su rendimiento en las asignaturas de los primeros cursos, con posibilidad de ampliar a otro tipo de predicciones. Esta será, como no, la pestaña Predicciones.

Pero, como en cualquier proyecto real, todo lo previamente estipulado puede cambiar según las necesidades del proyecto. Además, es previsible que, una vez familiarizado con los datos, se ideen módulos adicionales.

Diseño gráfico de la aplicación

Una vez definidos los módulos se procede a hacer un diseño a mano de la interfaz gráfica de usuario.

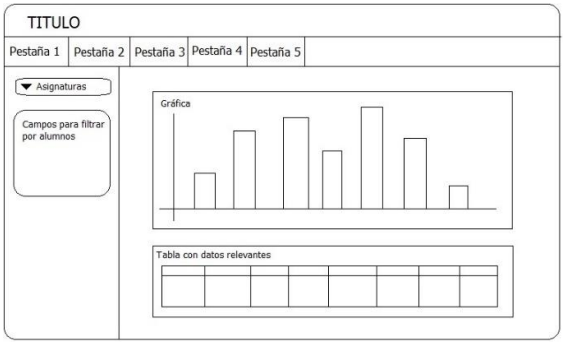


Ilustración 10 Diseño pestana Asignaturas

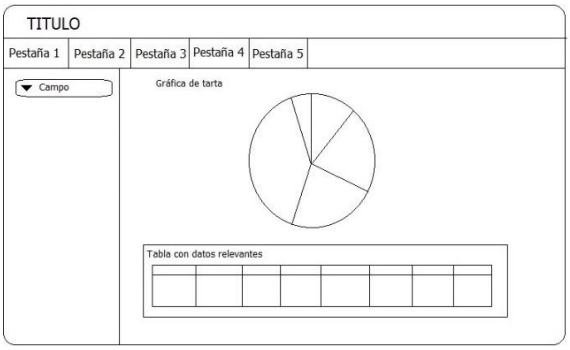


Ilustración 11 Diseño pestaña Comparaciones

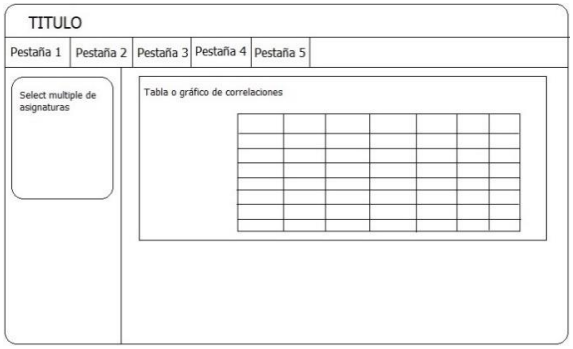


Ilustración 12 Diseño pestaña Correlaciones

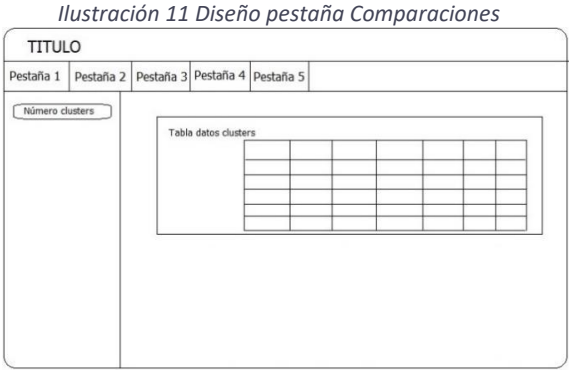


Ilustración 13 Diseño pestaña Clustering

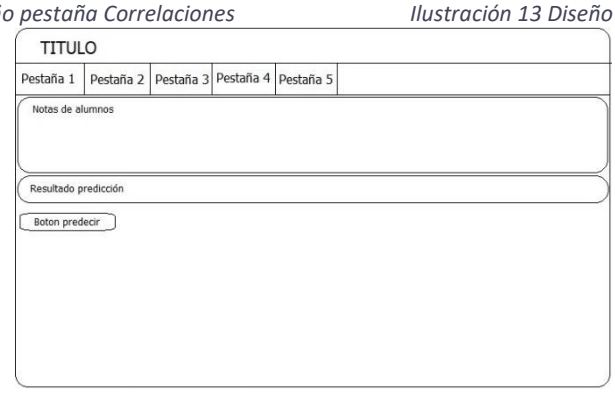


Ilustración 14 Diseño pestaña Predicciones

Después de esbozar la interfaz, se pasó a idear la relación entre bloques. La idea inicial era crear bloques independientes, que solo compartan los datos globales, es decir, los data frames descritos en la sección análisis. Por otro lado, a estos datos se les podrían añadir ciertas variables que puedan ser utilizadas en varios módulos, como pueden ser edad mínima y máxima de los alumnos, primer y último curso con datos… todas calculadas dinámicamente para no tener una dependencia de los datos originales. Por último, sería interesante crear un data frame propio con las notas medias de los alumnos tanto en total como por bloques (primero, segundo, tercero general, rama y optativas) para diversos cálculos que puedan surgir.

Por último, llega el momento de dividir el trabajo en fases. Se han diseñado cinco módulos, los dos primeros más sencillos, pues solamente muestran datos en base a diversos filtros. La primera fase se corresponde con estos dos módulos, sirviendo para sentar las bases de la aplicación y coger más soltura con las herramientas de desarrollo. Los dos siguientes módulos abarcan cálculos más complejos como son las correlaciones o el clustering, que se corresponderán con la segunda fase. Para finalizar la tercera fase se dedicará al quinto modulo, el de predicción. Evidentemente estas fases no están grabadas en piedra, ni pasar a la siguiente fase implica que lo creado con anterioridad este acabado completamente.

Capítulo 5: Desarrollo

Diseñada la aplicación llega el momento de comenzar a programar. Gracias a la librería Shiny para R es muy sencillo crear una aplicación web, permitiendo centrar los esfuerzos en las características y cálculos internos.

Asignaturas y comparaciones

Toda aplicación desarrollada en base al paquete shiny ha de seguir una cierta estructura. Ha de tener un elemento *ui* para contener la interfaz gráfica y un elemento *server* para procesar las reacciones a las interacciones con la aplicación, ambos pudiendo estar contenidos en el mismo archivo, llamándose app.R, o en dos diferentes ui.R y server.R, además de un tercero opcional llamado global.R para los datos compartidos. Es importante seguir la terminología, si no se sigue la aplicación no funcionara.

```
library(shiny)


ui <- ...

server <- ...

shinyApp(ui = ui, server = server)
```

Ilustración 15 Estructura básica de una aplicación de shiny

Para empezar se especificara la interfaz gráfica. Shiny pone al alcance del programador una gran variedad de páginas y paneles que nos permiten distribuir los elementos*: fluidPage, navbarPage, fixedPage, absolutePanel, conditionalPanel, fixedPanel, headerPanel, inputPanel, mainPanel, navlistPanel, sidebarPanel, tabPanel, tabsetPanel, titlePanel y wellPanel*. En el diseño se estipuló que cada uno de los módulos fuese una pestaña de la aplicación, así que, tras consultar la documentación, lo más adecuado para este caso es utilizar un *navbarPage*, para dividir el contenido en pestañas y en cada una de ellas incluir un *tabpanel*. Además, cada pestaña tendrá un menú lateral y un panel para el resto de componentes, generándose con un *sidebarPanel* y un *mainPanel*. Esta será la estructura que seguirán casi todas las pestañas.

```
ui <- navbarPage(
  title = "Aplicación principal",
  tabPanel(
    title = "Asignaturas",
    sidebarPanel(
     ),
    mainPanel(
   )
  )
)
```

Ilustración 16 Código de la base de la interfaz gráfica

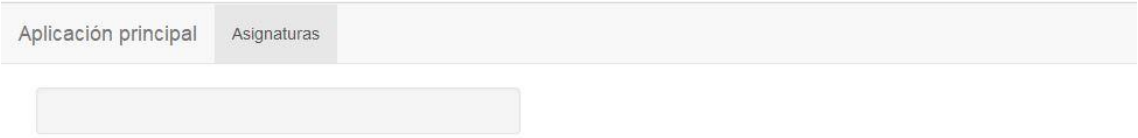


Ilustración 17 Base de la interfaz gráfica

Una vez generada una base para la pestaña se procede a añadir un *select* con las asignaturas, pues el primer módulo servirá para visualizar las notas de cada una de las asignaturas.

```
selectInput("asig","Asignaturas: ", choices = asig$ASI)
```

Ilustración 18 Instrucción en shiny para crear un select

Esta sencilla instrucción crea toda una serie de estamentos en HTML para crear un elemento.

```
▼<div class="form-group shiny-input-container">
   <label class="control-label" for="asig-selectized">Asignaturas: </label>
  ▼<div>
   ▶<select id="asig" tabindex="-1" class="selectized shiny-bound-input" style="display: none;">…</select>
   ▼<div class="selectize-control single">
    ▼<div class="selectize-input items not-full has-options">
       <input type="text" autocomplete="off" tabindex id="asig-selectized" style="width: 4px; opacity: 1;
       position: relative; left: 0px;" class="shiny-bound-input">
     </div>
    ▼<div class="selectize-dropdown single" style="display: none; visibility: visible; width: 375px; top: 34px;
    left: 0px;">
     ▼<div class="selectize-dropdown-content">
        <div class="option" data-selectable data-value="13912">13912</div>
        <div class="option" data-selectable data-value="13911">13911</div>
        <div class="option" data-selectable data-value="13910">13910</div>
        <div class="option" data-selectable data-value="13909">13909</div>
        <div class="option" data-selectable data-value="13908">13908</div>
        <div class="option" data-selectable data-value="13907">13907</div>
        <div class="option" data-selectable data-value="13906">13906</div>
        <div class="option" data-selectable data-value="13905">13905</div>
        <div class="option" data-selectable data-value="13904">13904</div>
        <div class="option" data-selectable data-value="13795">13795</div>
        <div class="option" data-selectable data-value="13788">13788</div>
```

Ilustración 19 código HTML generado por la instrucción previa

Y este código HTML genera el *select* deseado.

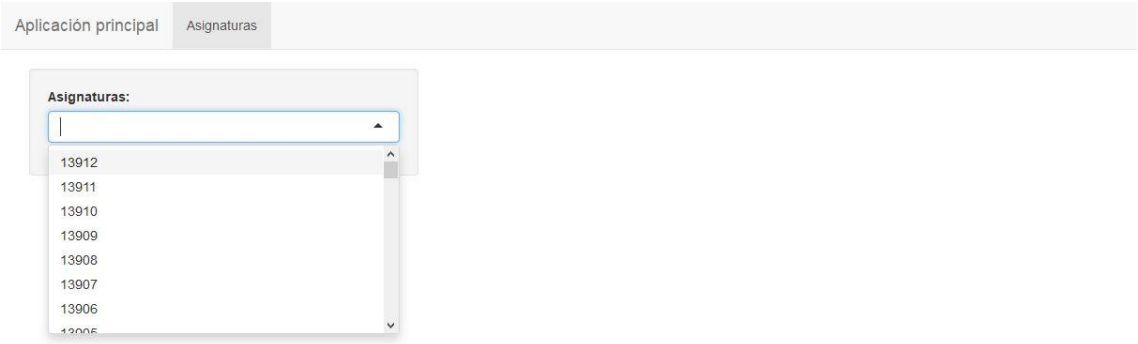


Ilustración 20 Select generado

Después se procede a añadir un histograma para ver las notas de la asignatura seleccionada. Para ello se añade un elemento *plotOutput* en *ui* y un *renderPlot* en *server*, además de un método para filtrar las notas según que asignatura este seleccionada.

```
ui <- navbarPage(
  title = "Aplicación principal",
  tabPanel(
    title = "Asignaturas",
    sidebarPanel(
       selectInput("asig","Asignaturas: ", choices = asig$ASI)
    ),
    mainPanel(
     plotOutput("histograma")
   )
  )
)
```

Ilustración 21 Código de la interfaz añadiendo un histograma

```
server <- function(input, output) {
  datosHist <- reactive({
    asignaturas_ofu %>%
      filter(ASI == input$asig)
  })
  output$histograma <- renderPlot({
    hist(datosHist()$NOTA,
         main = str_c("Histograma ", input$asig), xlab = "Notas", ylab = "Num",
         breaks = seq(0, 10), col = "#3399CC")
  })
}
```

Ilustración 22 Código para llenar el histograma

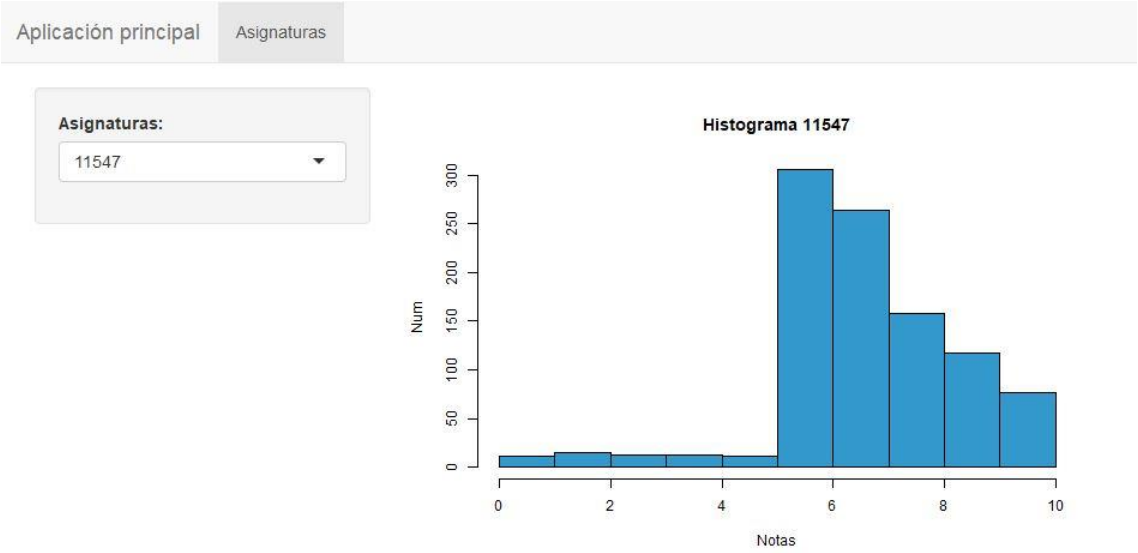


Ilustración 23 Histograma generado

Una vez se genera correctamente el histograma se añaden los campos para filtrar alumnos.

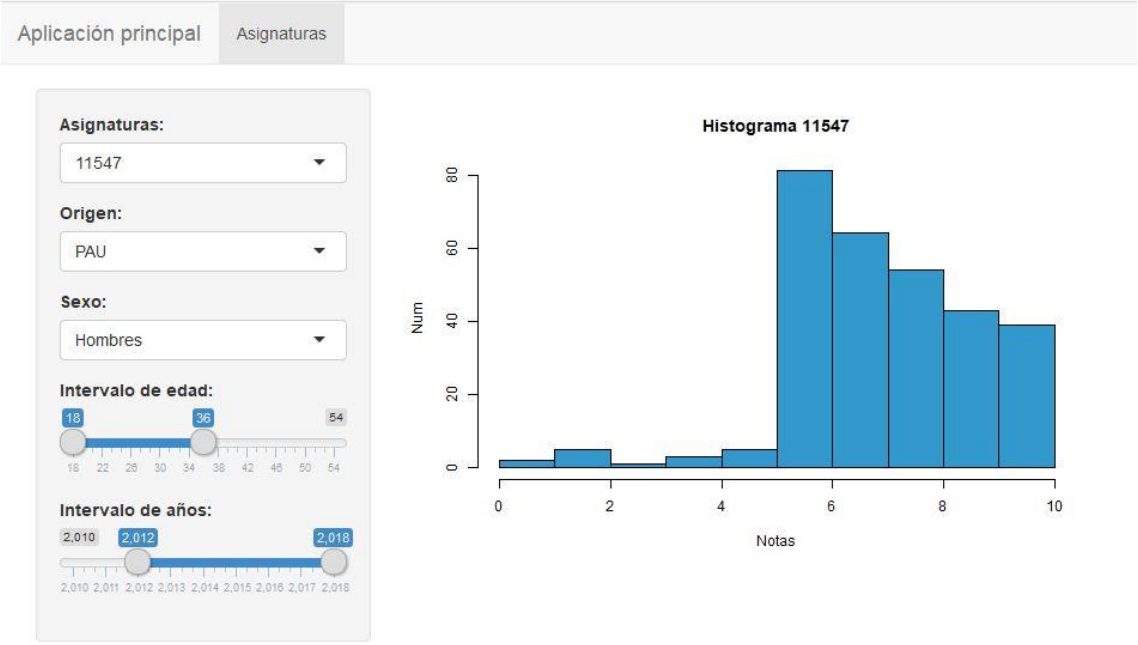


Ilustración 24 Interfaz al añadir filtros

Para acabar con este módulo se añaden una serie de valores estadísticos sobre los datos ya filtrados. Estos datos son el número de alumnos matriculados, el porcentaje de matrículas aprobadas respecto al total de matrículas totales y el número de primeras matriculas, segundas matriculas…

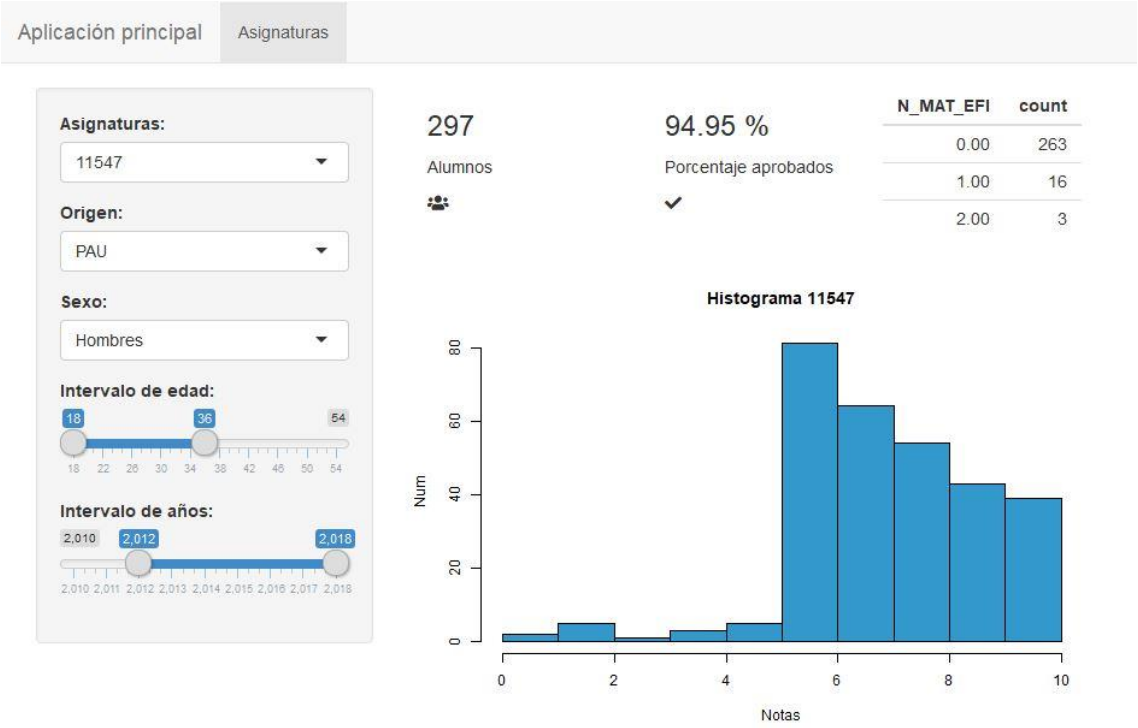


Ilustración 25 Aspecto de la aplicación al finalizar la pestaña

La siguiente pestaña, llamada comparaciones, servirá para comparar datos de los alumnos respecto a diversos campos. Debido a esto partimos de una base semejante a la anterior pestaña, un *select* en este caso para seleccionar el campo a comparar. Los atributos elegidos para comparar son los siguientes: sexo, edad, origen y notas entrada.

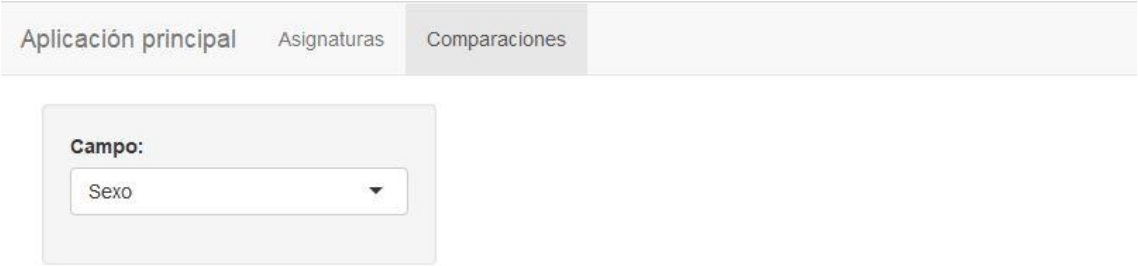


Ilustración 26 Selector de campos en de la pestaña Comparaciones

En este caso hay que separar los datos en grupos. Aquí hay que distinguir entre dos tipos de variables, las numéricas como son edad y notas de entrada y las variables categóricas que son las restantes, sexo y origen. Para las variables categóricas (que son aquellas que tienen un valor concreto entre una serie de posibilidades) la separación es trivial, pero para las numéricas es preciso separar los posibles valores en intervalos.

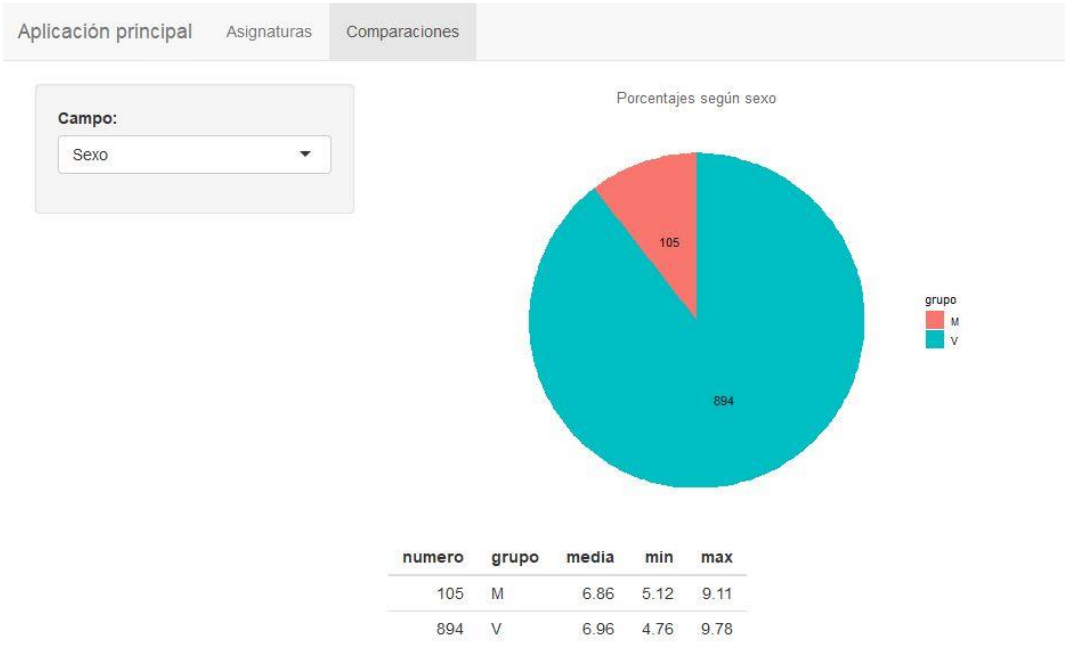
Para separar los datos y recoger información relevante se crearon dos métodos gemelos (uno para variables numéricas y otro para categóricas) que se ocupen de ello.

```
fillList <- function(niveles, fuente, valores, asig) {
  i <- 1
  numero <- c()
  media <- c()
  min <- c()
  max <- c()
  grupo <- niveles
  while(i <= length(niveles)){
    filtrado <- fuente %>% filter(valores == niveles[i])
    datos <- asig %>% semi_join(filtrado, by = c("id","id"))
    medias <- datos %>% group_by(id) %>% summarise(media = mean(NOTA, na.rm = TRUE))
    numero[[i]] <- filtrado %>% nrow()
    media[[i]] <- mean(medias$media, na.rm = TRUE)
    min[[i]] <- min(medias$media, na.rm = TRUE)
    max[[i]] <- max(medias$media, na.rm = TRUE)
    i <- i + 1
  }
  filtrado <- fuente %>% filter(is.na(valores))
  if (nrow(filtrado) != 0) {
    datos <- asig %>% semi_join(filtrado, by = c("id","id"))
    medias <- datos %>% group_by(id) %>% summarise(media = mean(NOTA, na.rm = TRUE))
    numero[[length(niveles)+1]] <- filtrado %>% nrow()
    media[[length(niveles)+1]] <- mean(medias$media, na.rm = TRUE)
    min[[length(niveles)+1]] <- min(medias$media, na.rm = TRUE)
    max[[length(niveles)+1]] <- max(medias$media, na.rm = TRUE)
    grupo[[length(niveles)+1]] <- "Desconocido"
  }
  tibble(numero, grupo, media, min, max)
}
```

Ilustración 27 Función para separar las asignaturas en grupos

En este método (y su equivalente para datos numéricos, fillList2) se separan los alumnos en grupos, cuenta cuantos hay en cada uno, calcula la media de cada grupo y separa la nota mínima y máxima.

Utilizando estos datos se procede a crear una gráfica de tarta y una tabla con el resto de información.



numero	grupo	media	min	max
105	M	6.86	5.12	9.11
894	V	6.96	4.76	9.78

Ilustración 28 Pestaña comparaciones la gráfica y la tabla

Correlaciones y clustering

Finalizadas las pestañas Asignaturas y comparaciones se pasa a implementar Correlaciones y Clustering. Para estas dos operaciones hay que tener ciertas cosas en cuenta, lo primero de todo saber en qué consisten. En el campo de la estadística una correlación entre variables se refiere a que los valores de esas variables aumentan o disminuyen sistemáticamente respecto a los valores de las demás (aunque esta correlación no tiene por qué implicar causalidad). El clustering (o algoritmo de agrupamiento en español) consiste en agrupar una serie objetos según unos criterios, por lo general distancia o similitud.

Empezando por las correlaciones, las variables a estudiar serán asignaturas. Los valores a comparar serán las notas de los matriculados en ella. Cuando se calcula la correlación entre variables se obtiene un valor conocido como fuerza. Esta fuerza va desde -1 hasta 1, viendo el significado de este número en la tabla

Fuerza	**Nivel de la relación**
-1,0 a -0,5 o 1,0 a 0,5	Alto
-0,5 a -0,3 o 0,3 a 0,5	Medio
-0,3 a -0,1 o 0,1 a 0,3	Bajo
-0,1 a 0,1	Ninguna o muy baja

Tabla 1 Nivel de la relación según la fuerza

Es importante señalar que, para poder calcular la correlación, el número de valores debe ser el mismo. Así pues, como los datos son de alumnos que han acabado el grado se sabe que todos han cursado y aprobado las asignaturas generales. Para empezar, se calcularía la correlación entre las notas aprobadas de los dos primeros cursos.

Primero se separan las notas de las asignaturas y luego los pasamos a la función char.correlation(…) del paquete *PerformanceAnalytics* que calcula la correlación entre los datos.

```
notas <- reactive({
  asigSep <- asignaturas_ofu %>% filter(BLO == input$bloques)
  separarAsi(asignaturas_ofu, unique(asigSep$ASI))
})
output$correlGraf <- renderPlot({
  chart.Correlation(notas(), histogram = F, pch = 19)
})
```

Ilustración 29 Código para crear el gráfico de correlaciones

```
separarAsi <- function(datos, valores){
  lista <- c()
  i <- 1
  for(j in valores){
    lista[[i]] <- datos %>% filter(ASI == j) %>% filter(NOTA>=5)
      %>% arrange(desc(ASI)) %>% select(NOTA)
    lista[[i]] <- as.numeric(as.character(unlist(lista[[i]])))
    i <- i + 1
  }
  result <- data.frame(lista)
  colnames(result) <- valores
  result
}
```

Ilustración 30 Función que separa las notas de las asignaturas del bloque

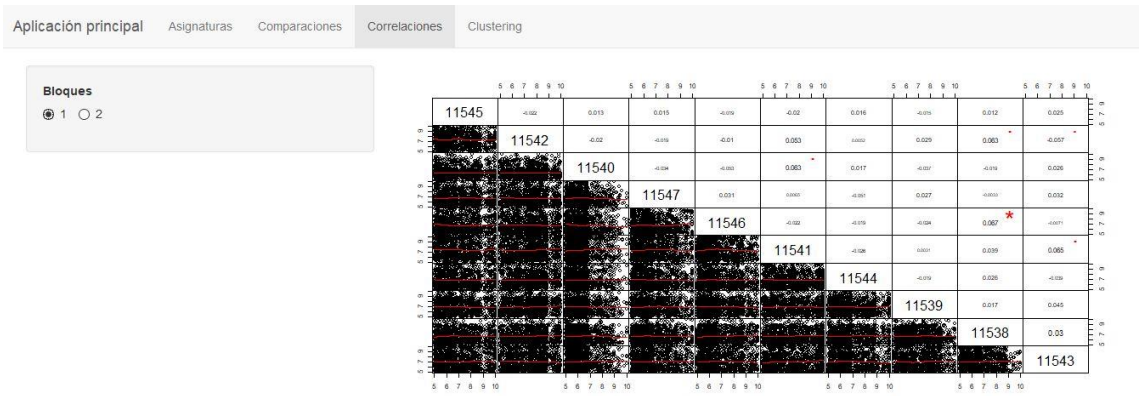


Ilustración 31 Pestaña con el gráfico de correlaciones

Pasando al clustering de alumnos hay que decidir según qué campo agruparlos. De momento se toma la decisión de agruparlos por la nota media. De nuevo en R existe una función muy sencilla para hacer cálculos complejos, en esta ocasión kmeans(…) al que solamente hay que pasarle el campo a agrupar y el número de grupos. Una vez ejecutada la función se puede extraer toda clase de información interesante. Se empezará mostrando el grupo, el centro del grupo y el número de miembros.

```
output$clusterTab <- renderTable({
  km <- kmeans(notasPorAlumnos$NOTA, input$clusterK)
  clust <- notasPorAlumnos %>% mutate(cluster = km$cluster)
  medias <- clust %>% group_by(cluster)
    %>% summarise(media = mean(NOTA), num = n())
})
```

Ilustración 32 Código para separar en grupos y mostrar los datos

cluster	media	num
1	6.96	290
2	5.65	137
3	7.69	194
4	6.33	263
5	8.63	115

Ilustración 33 Pestaña de clustering con la tabla de datos

Aunque se han desarrollado los módulos deseados, aún se encuentran en un estado muy inicial y los gráficos de las pestañas previas no resultaban del todo satisfactorios, por estos motivos se decidió hacer una nueva iteración antes de pasar a la pestaña Predicciones. Además, en este momento se recibió el nuevo data frame con la información de las asignaturas, **asignaturas_info**, así que se abrían nuevas posibilidades.

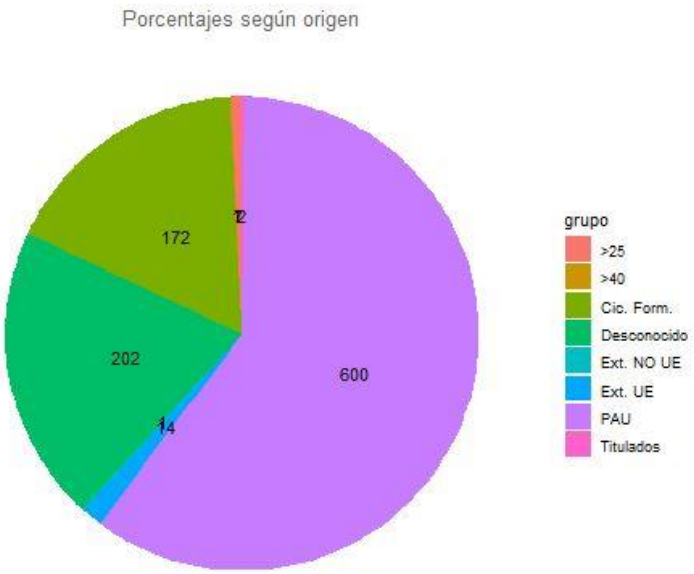


Ilustración 34 Gráfico de tarta con los números solapados

Se volvió sobre la pestaña “Asignaturas” y se cambió el *select* original de asignaturas por uno nuevo que muestra el nombre en vez del código y unos *radio buttons* que permiten navegar por los distintos bloques.

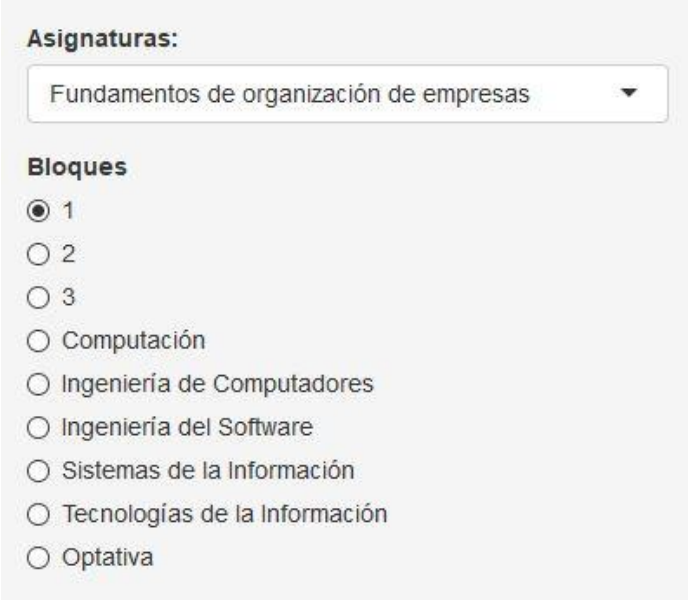


Ilustración 35 Nuevo select de asignaturas con filtrador por bloque

Después se cambió la gráfica de R por la del paquete plotly, que además de ser más atractiva visualmente ofrece una serie de funcionalidades añadidas como son el exportado a png, hacer zoom, mover los ejes…

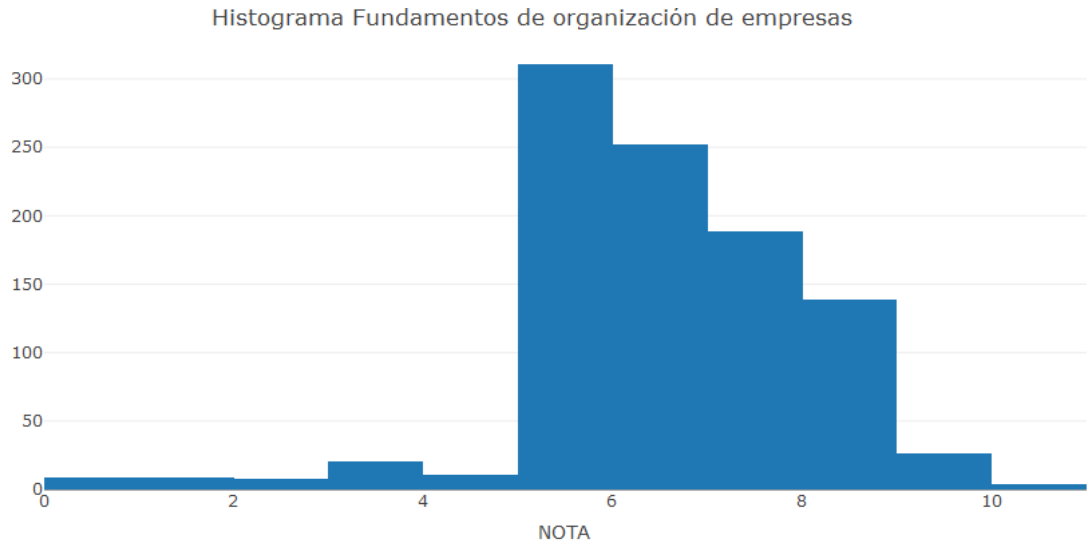


Ilustración 36 Nuevo histograma creado con plotly

Visto el buen resultado de estos cambios se introdujeron también en la pestaña de comparaciones, permitiendo ahora hacer comparaciones por asignaturas. Además, se añadió un histograma para tener una idea visual de las notas de los alumnos.

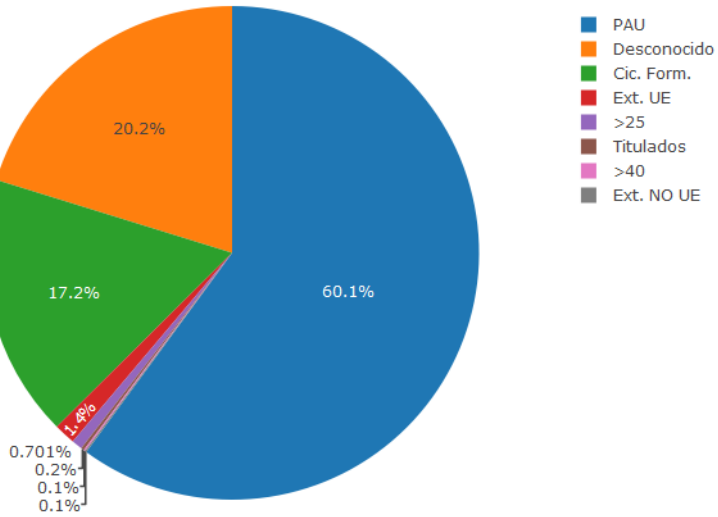


Ilustración 37 Gráfica de tarta creada con plotly

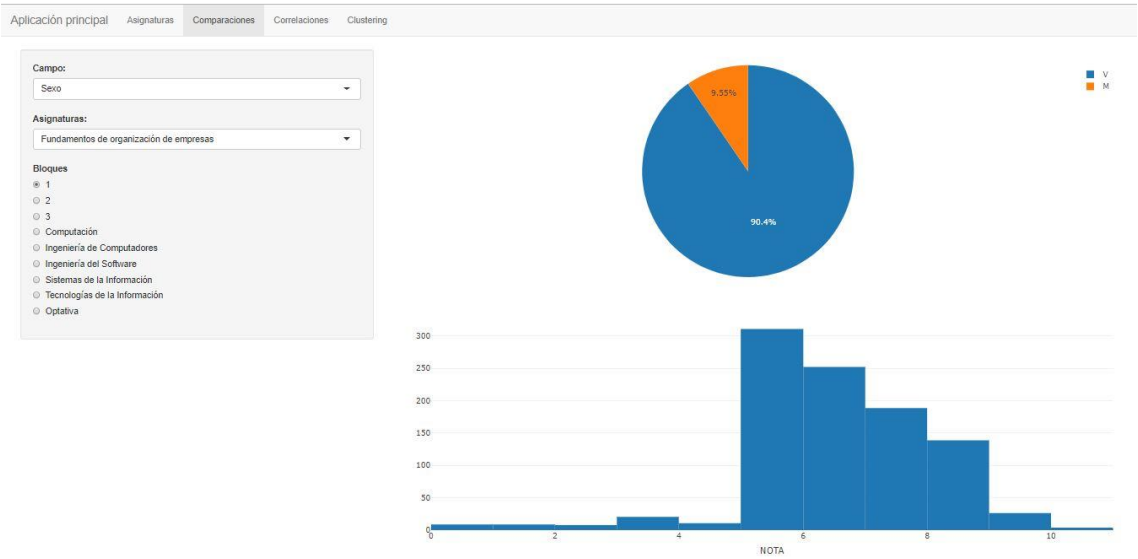


Ilustración 38 Nueva versión de la pestaña Comparaciones

Gracias al nuevo selector de asignaturas se puede implementar fácilmente una gran mejora en correlaciones, como es utilizar una variante múltiple suya para seleccionar aquellas asignaturas que se desean comparar, permitiendo un análisis más concreto y una mejor representación visual, adaptando la función que separa las notas de los alumnos que están en las asignaturas seleccionadas.

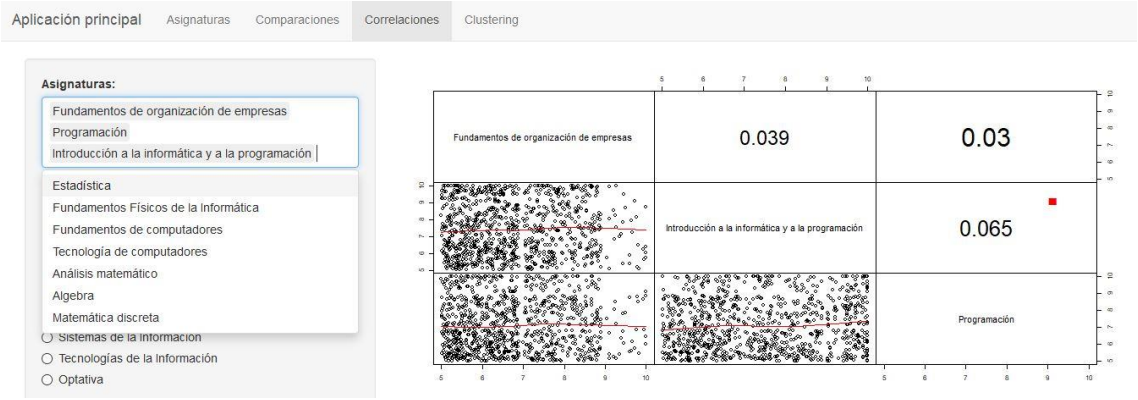


Ilustración 39 Ventana de correlaciones con la posibilidad de seleccionar asignaturas

```
separarAsi <- function(datos, valores){
  lista <- c()
  alumnos <- unique(datos %>% filter(!is.na(NOTA)) %>% filter(NOTA>=5) %>% select(id))
  for (j in valores$ASI){
    aux <- datos %>% filter(!is.na(NOTA))
      %>% filter(NOTA>=5) %>% filter(ASI == j)
    alumnos <- alumnos %>% filter(id %in% aux$id)
  }
  i <- 1
  for(j in valores$ASI){
    lista[[i]] <- datos %>% filter(ASI == j) %>% filter(id %in% alumnos$id)
      %>% filter(NOTA>=5) %>% arrange(desc(ASI)) %>% select(NOTA)
    lista[[i]] <- as.numeric(as.character(unlist(lista[[i]])))
    i <- i + 1
  }
  result <- data.frame(lista)
  colnames(result) <- valores$nombre
  result
}
```

Ilustración 40 Código para seleccionar las notas de los alumnos matriculados en las asignaturas seleccionados

Para mejorar la pestaña de clustering se decidió añadir a la tabla de los clusters algunos campos del alumno que ocupa el centro de cada uno de ellos.

cluster	num	media	ANYCOM	ANYFIN	GENERO	ING_ESTUDIOS	RESI_F	BECA	EDAD_ING
1	432	7.03	2012.00	2016.00	V	PAU	ProvVal	S	19.00
2	352	6.03	2011.00	2014.00	V	NA	ProvVal	-	21.00
3	215	8.28	2013.00	2016.00	V	PAU	Val	S	18.00

Ilustración 41 Tabla de clustering con datos de los alumnos centrales de cada grupo

Predicciones

Para la pestaña Predicciones en primer lugar se desarrolla la interfaz, como en los otros casos. Siguiendo el diseño creado previamente se crean una serie de *inputs numéricos* para seleccionar las notas del primer y segundo año, además del botón que dará lugar al cálculo de la predicción.

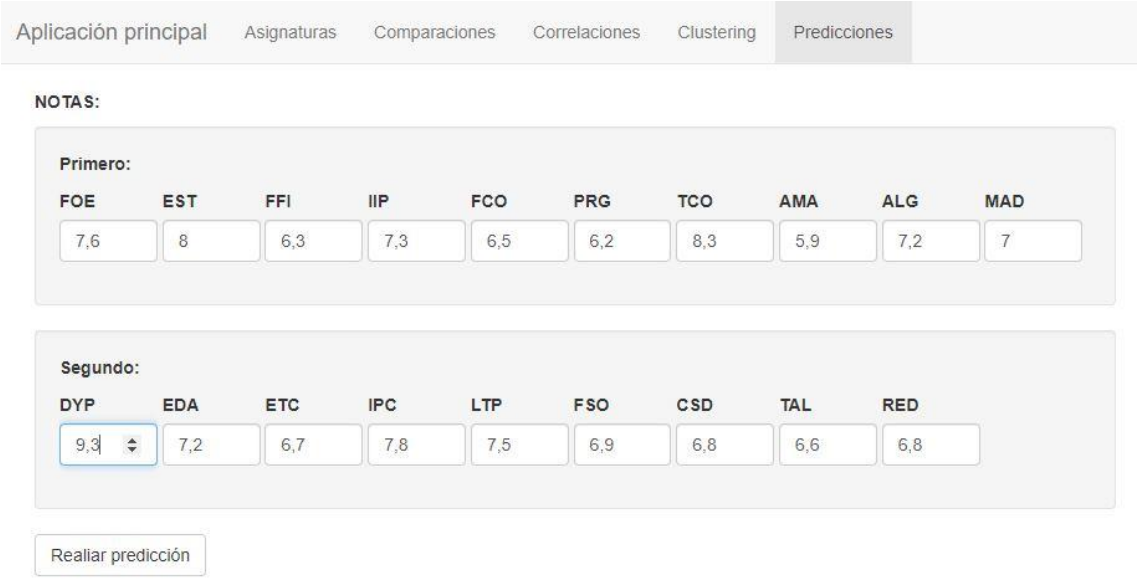


Ilustración 42 Pestaña de predicciones

Una vez la interfaz está preparada se procede al desarrollo de los cálculos internos. Para la realización de predicciones existe el paquete de R caret. Caret permite con unas pocas instrucciones el crear una gran variedad de modelos predictivos.

Pero al tener disponibles una gran variedad de métodos de predicción, ¿cuál se debe escoger? Para determinar cuál es el más idóneo se procedió a hacer comparaciones entre los modelos utilizando un 70% de los alumnos disponibles como base de entrenamiento y el 30% restante como test.

Primero hay que asignar a cada alumno su rama. Para ello se crea una nueva columna en el data frame **titulados_ofu** con la rama de cada uno de ellos, ya que tener a mano la rama de los estudiantes puede resultar muy útil en otras pestañas y no solo Predicciones.

```
encuentraRama <- function(alumno) {
  head(unique((asignaturas_ofu %>%
                 filter(id == alumno) %>% filter(BLO == 3) %>%
                 merge(asignaturas_info, by.y = "codigo", by.x = "ASI") %>%
                 filter(bloque != 3) %>% group_by(bloque) %>%
                 summarise(n = n()) %>% arrange(desc(n)))$bloque),1)
}
```

Ilustración 43 Función para encontrar rama

```
titulados_ofu$Rama <- as.factor(unlist(map(titulados_ofu$id, encuentraRama)))
```

Ilustración 44 Instrucciones para crear la nueva columna de rama

Además, para realizar las predicciones es necesario enlazar cada alumno con las notas de las asignaturas que servirán para predecir, es decir, las de primero y segundo. Como es un cálculo lento se procede a crear otro data frame donde almacenar esos valores, llamado **notasPorAlumnosSep**.

Creados los data frames necesarios se procede a separar los datos en los grupos de entrenamiento y test.

```
set.seed(3456)
crossover <- notasPorAlumnosSep %>% merge(alumnos_ofu, by = c("id", "id"))

trainIndex <- createDataPartition(crossover$rama, p = .7, list = FALSE, times = 1)

dtrain<-crossover[trainIndex,]
dtest<-crossover[-trainIndex,]
```

Ilustración 45 Código para separar un grupo de entrenamiento y uno de pruebas

Para crear un modelo en caret (y en general en cualquier sistema predictivo) primero hay que hacer un "entrenamiento", en este caso mediante la función train(…). A esta función se le pasa varios argumentos, el valor a predecir separado por el símbolo ~ de las variables para a evaluar, el conjunto de datos, el método de predicción y cualquier parámetro extra que requiera algún método en específico.

Las técnicas a comparar son lineal discriminante, que utiliza un algoritmo para tratar de crear una frontera entre los grupos, k vecinos que comprueba mediante unos cálculos de distancias cuales son los vecinos más cercanos y asigna el grupo según ellos y rpart que crea un árbol de decisión por el que pasa los datos a predecir.

Método	Precisión
Lineal discriminante	40,51%
10 vecinos	37,59%
15 vecinos	39,41%
20 vecinos	37,20%
Rpart	40,51%

Tabla 2 Comparación precisión predicciones

Como se puede observar todas las técnicas son bastante semejantes, siendo las mejores en este caso lineal discriminante y rpart. Se utilizará lineal discriminante para los cálculos de la predicción.

```
dtrain <- notasPorAlumnosSep %>% merge(titulados_ofu, by = c("id", "id"))
fitControl <- trainControl(
  method = "cv",
  number = 10,
  savePredictions = TRUE
)
modelo<-train(rama~FOE+EST+FFI+IIP+FCO+PRG+TCO+AMA+ALG+MAD+DYP+EDA+ETC+IPC+
              data=dtrain,
              method="lda",
              trControl=fitControl)
```

Finalmente se añade una tabla para mostrar el porcentaje de probabilidad de que el alumno con las notas introducidas entre en cada una de las ramas.

Ilustración 46 Pestaña de predicciones funcional

Después de crear la pestaña de predicciones, se procede a añadir una serie de mejoras a algunas de las otras pestañas.

El histograma de la sección de comparaciones de momento resulta poco útil, así que se decidió separar por colores según los grupos. Además, se añadió la posibilidad de filtrar por todas las asignaturas tanto en general como por bloque. Por último se añade un filtro por cursos.

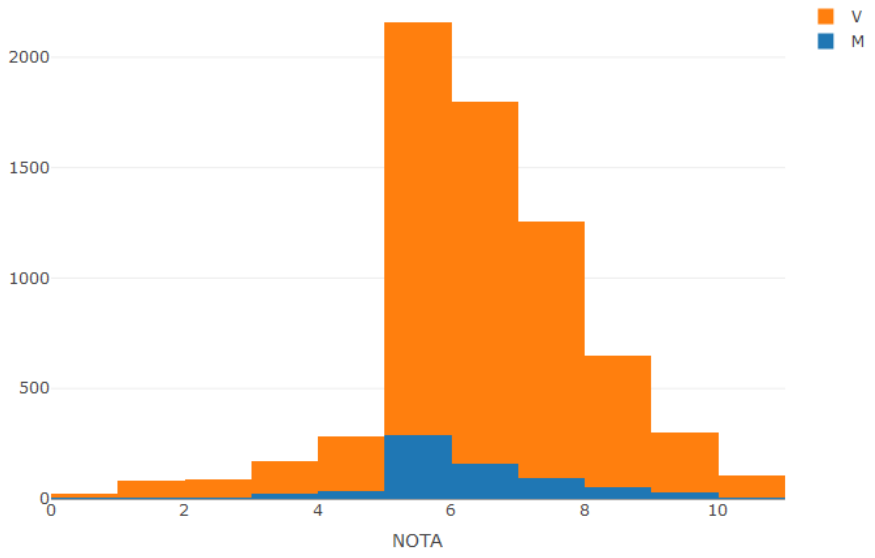


Ilustración 47 Histograma de Comparaciones con las notas separadas por los grupos

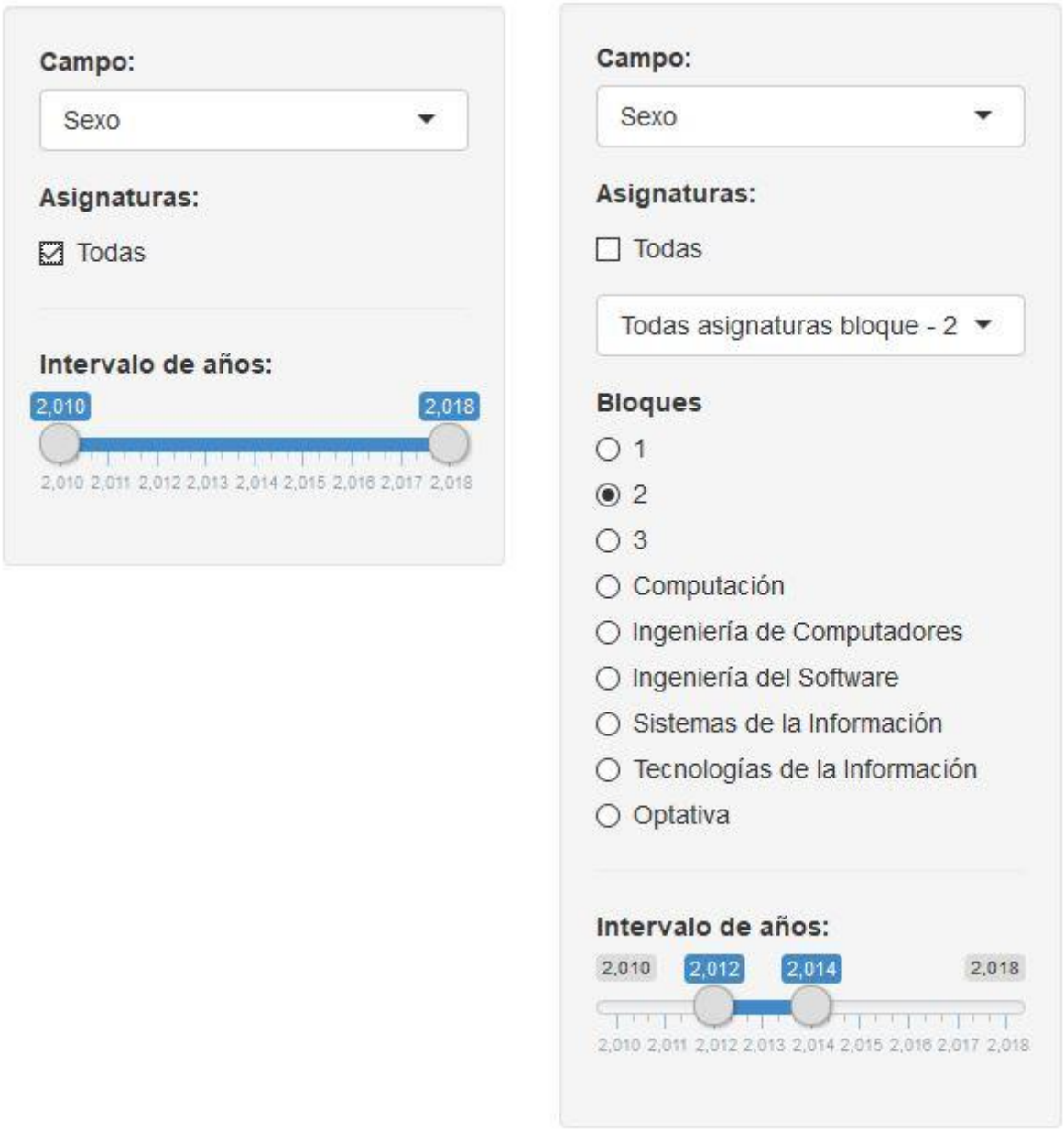


Ilustración 48 Filtros de Comparaciones desplegados y sin desplegar

Para mejorar las correlaciones en vez de utilizar un único *select* múltiple que se vacia cada vez que se cambia de bloque y por tanto hacía imposible hacer comparaciones entre cursos/ramas, se cambió por cuatro *selects* independientes, dos de ellos inhabilitados inicialmente, pudiendo seleccionar el bloque que se desee.

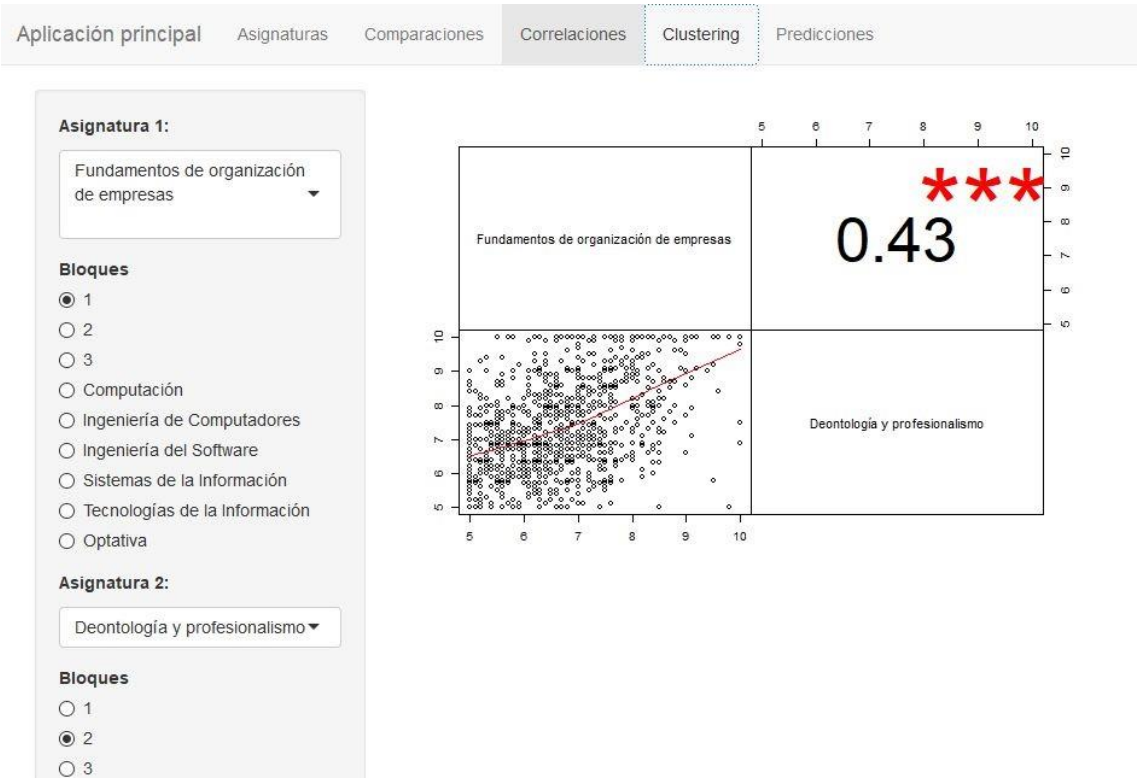


Ilustración 49 Pestaña de Correlaciones con varios selects

A la ventana de clustering se añadió un *select* para poder elegir campos de los grupos y mostrar unas gráficas que muestren información relevante. Para campos categóricos se generan gráficos de tarta y para los numéricos diagramas de caja y bigotes.

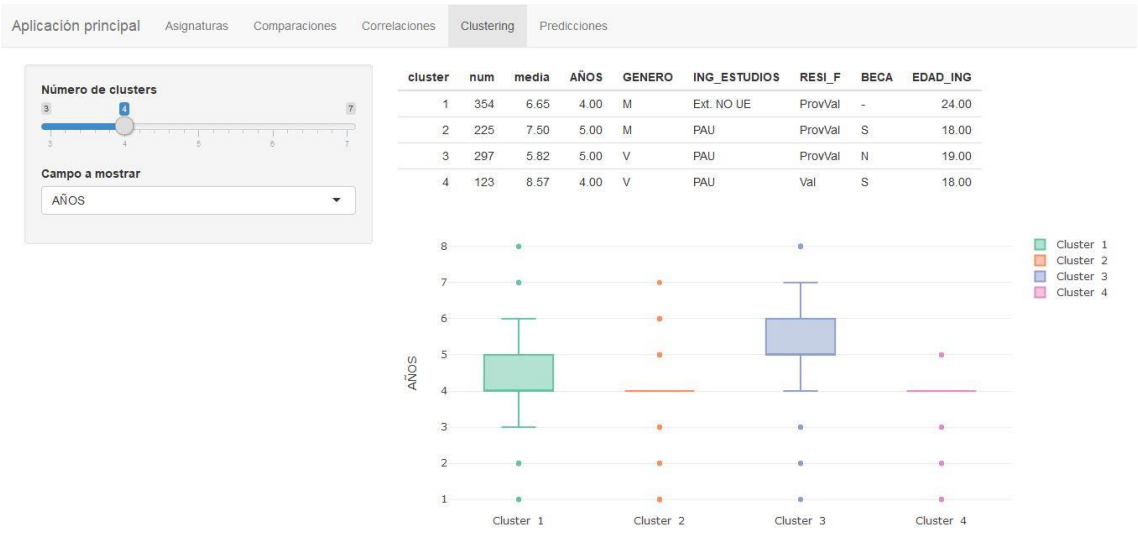


Ilustración 50 Pestaña de clustering con gráficos de cajas y bigotes

Ilustración 51 Pestaña de clustering con gráficos de tartas

```
output$clusterGraf <- renderPlotly({
  clust <- titulados_ofu %>% merge(notasPorAlumnos, by = c("id", "id"))
  %>% mutate(cluster = km()$cluster, AÑOS = ANYFIN - ANYCOM + 1, EDAD_ING = EDAD_31_12_ING, BECA = OBTIENE_BECA_1)
  if (typeof(unlist(clust %>% select(input$campo4))) == "integer"){
    max <- max(clust %>% select(cluster))
    lista <- list(c(0,0), c(0,1), c(0,2), c(1,0), c(1,1), c(1,2), c(2,0))
    plot <- plot_ly()
    for(i in 1:max) {
      grupo <- clust %>% group_by(get(input$campo4)) %>% filter(cluster == i) %>% summarise(num = n())
      names(grupo)[1] <- "id"
      plot <- plot %>% add_pie(data = grupo, labels = ~id, values = ~num, name = paste("Cluster ", i),
                               sort = FALSE, domain = list(row = lista[[i]][1], column =lista[[i]][2]))
    }
    plot %>% layout(title = input$campo4, showlegend = F,
                    grid=list(rows=3, columns=3),
                    xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
                    yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))
  } else {
    plot <- plot_ly(clust %>% mutate(clusterS = paste("Cluster ", cluster)),
                    y=~get(input$campo4), color=~clusterS, type="box") %>%
      layout(yaxis = list(title = input$campo4))
  }
})
```

Ilustración 52 Función para generar las gráficas de la pestaña Clustering

Tras implementar la pestaña de predicciones y mejorar alguna de las demás, se procede a hacer otra iteración para añadir aún más funcionalidades, mejorar más las ya existentes y añadir textos informativos en las diferentes secciones.

Está segunda vuelta comienza, de nuevo, por las predicciones. En primer lugar, se añade un *checkbox* para decidir utilizar las asignaturas del segundo curso o solo las del primero. Además, si se necesita predecir las ramas de varios alumnos es pesado tener que introducir a mano las notas de todos ellos. Por ello se añadió una pestaña alternativa donde poder subir archivos .csv con las notas de los alumnos, pudiendo seleccionar si se desea utilizar asignaturas de segundo o no, para generar una tabla con los porcentajes.

Ilustración 53 Pestaña de predicciones con opción de solo utilizar las asignaturas de primero

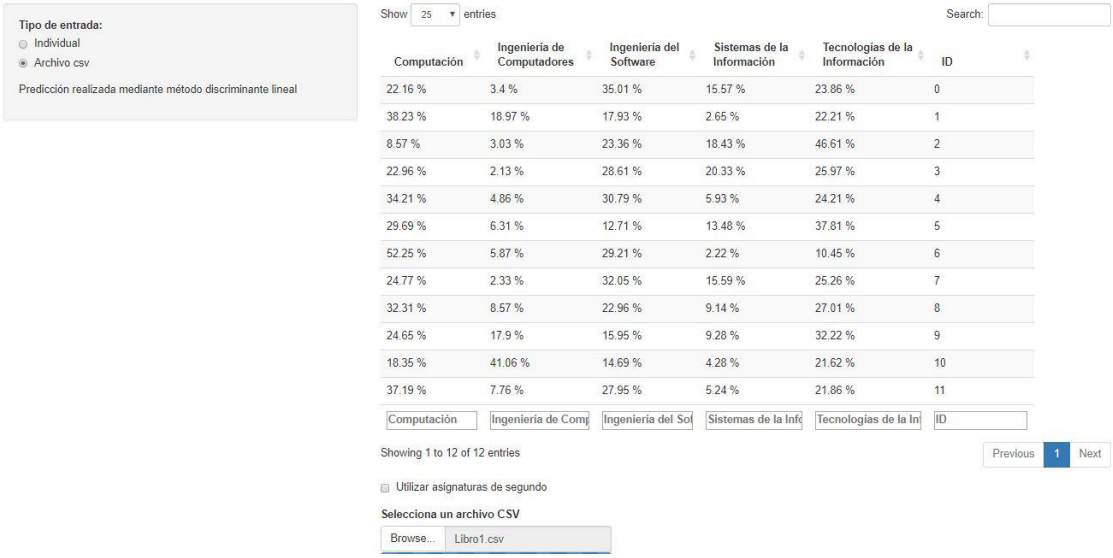


Ilustración 54 Sección para predecir desde un archivo csv

En la pestaña por asignaturas, los datos y tablas de información general que aparecían no resultaban todo lo claras que un primer momento parecieron, así que se sustituyeron por unos más convincentes.

Ilustración 55 Nuevos datos mostrados en la pestaña Asignaturas

Como en el momento de esta segunda iteración ya se disponía de las ramas de los alumnos se añadió un nuevo campo en Comparaciones para poder hacer comparaciones sobre los alumnos de una única rama.

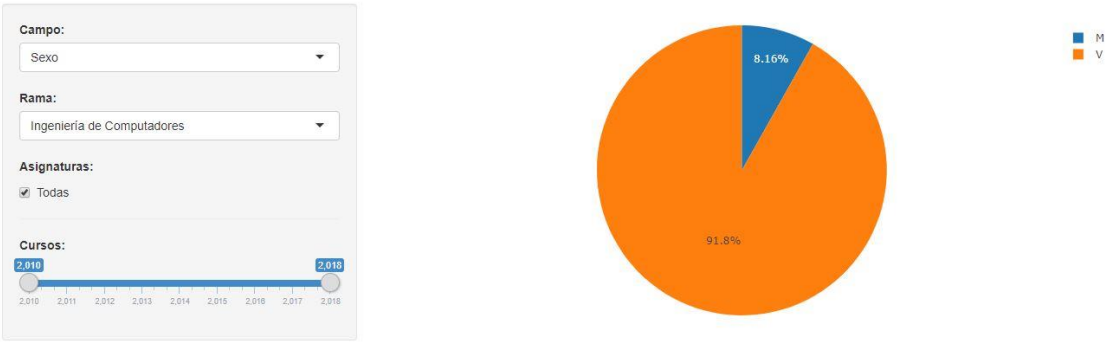


Ilustración 56 Pestaña Comparaciones con la posibilidad de seleccionar rama

La pestaña Clustering, aunque había mejorado en comparación con sus primeras versiones aun parecía bastante desaprovechada. Para mejorarla lo primero que se hizo fue cambiar la variable de agrupado. En vez de utilizar la media total se utilizaría la media de las partes generales de los tres primeros cursos por separado. A continuación, se añadieron algunos campos para filtrar los alumnos a agrupar. Por último, como se utilizaban datos de las medias de los cursos se añadió una gráfica para poder ver visualmente los grupos, pudiendo variar los valores de los ejes.

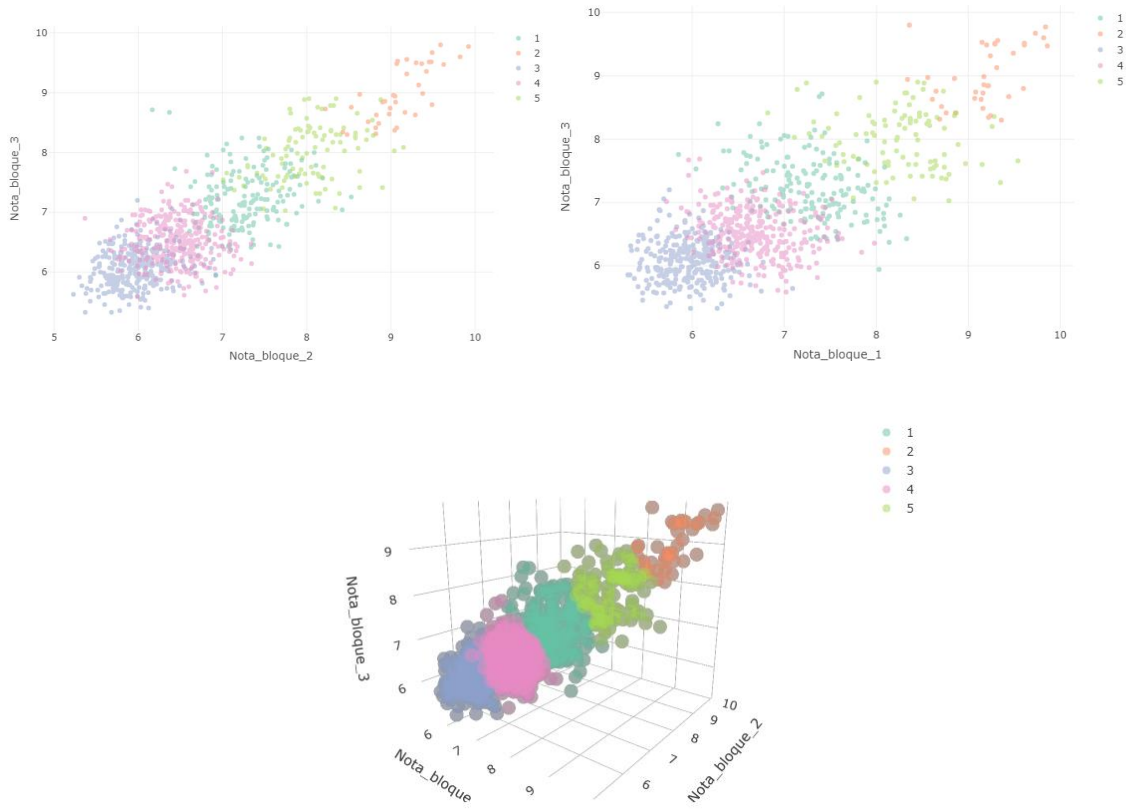


Ilustración 57 Filtros de la pestaña clustering y las diversas gráficas que se pueden generar

Por último en esta iteración, se decidió crear un nuevo módulo que no estaba previsto en el plan inicial. Esta nueva pestaña se creó para poder estudiar el rendimiento de los alumnos según su nota de ingreso.

Para esta nueva pestaña se pensó en una gráfica de puntos, en el eje x las notas de selectividad y en el eje y la nota media del grado. Sin embargo una pestaña así, aunque útil, parecía insuficiente. En vez de utilizar siempre la media total se utilizaría un *select* para decidir qué media mostrar (total, primer curso, segundo…) y unos filtros de edad y año de entrada. Además, se añadió la posibilidad de cambiar el color de los puntos según pertenezcan a varias categorías diferentes.

```
output$puntos <- renderPlotly({
  datos <- porRama %>% filter(!is.na(ING_NOTA)) %>%
    filter(between(EDAD_31_12_ING, input$edad6[1], input$edad6[2])) %>%
    filter(between(ANYCOM, input$anyo6[1], input$anyo6[2])) %>%
    mutate(BECA = OBTIENE_BECA_1, EDAD_ING = EDAD_31_12_ING)
  datos$Nota_ingreso <- as.numeric(levels(datos$ING_NOTA))[datos$ING_NOTA]
  datos <- datos %>% merge(notasPorAlumnos, by=c("id","id"))
  plot <- NULL
  switch (input$selectMedia,
    Total = plot <- plot_ly(datos, x=~Nota_ingreso, y=~MEDIA_OFICIAL, color=~get(input$color6), type = "scatter", mode="markers"),
    Primero = plot <- plot_ly(datos, x=~Nota_ingreso, y=~Nota_bloque_1, color=~get(input$color6), type = "scatter", mode="markers"),
    Segundo = plot <- plot_ly(datos, x=~Nota_ingreso, y=~Nota_bloque_2, color=~get(input$color6), type = "scatter", mode="markers"),
    Tercero = plot <- plot_ly(datos, x=~Nota_ingreso, y=~Nota_bloque_3, color=~get(input$color6), type = "scatter", mode="markers"),
    Rama = plot <- plot_ly(datos, x=~Nota_ingreso, y=~Nota_rama, color=~get(input$color6), type = "scatter", mode="markers"),
    Optativas = plot <- plot_ly(datos, x=~Nota_ingreso, y=~Nota_optativas, color=~get(input$color6), type = "scatter", mode="markers")
  )
  plot %>% layout(xaxis = list(title = "Nota de ingreso (PAU)"), yaxis = list(title = paste("Media ", input$selectMedia)))
})
```

Ilustración 58 Función para crear la tabla de la pestaña PAU

Ilustración 59 Pestaña PAU

Fase de cierre

Una vez finalizadas las pestañas predefinidas más una que no lo estaba se pasó a lo que se llamó “fase de cierre”. Esta “fase” en vez de ser como las otras donde se añadía funcionalidades adicionales fue una etapa de revisión corrigiendo posibles errores que hayan podido pasar desapercibidos y cambiar nombres de los diferentes *inputs* para que todo quedará más claro.

Para finalizar la aplicación se añadió una ventana principal, abierta al iniciar la aplicación, donde se muestra cierta información. Además, se le asigno un nombre a la herramienta: HARA ETSINF proveniente del título del TFG, “Desarrollo de una herramienta para el análisis de rendimiento del alumnado del Grado en Ingeniería Informática de la ETSINF”.

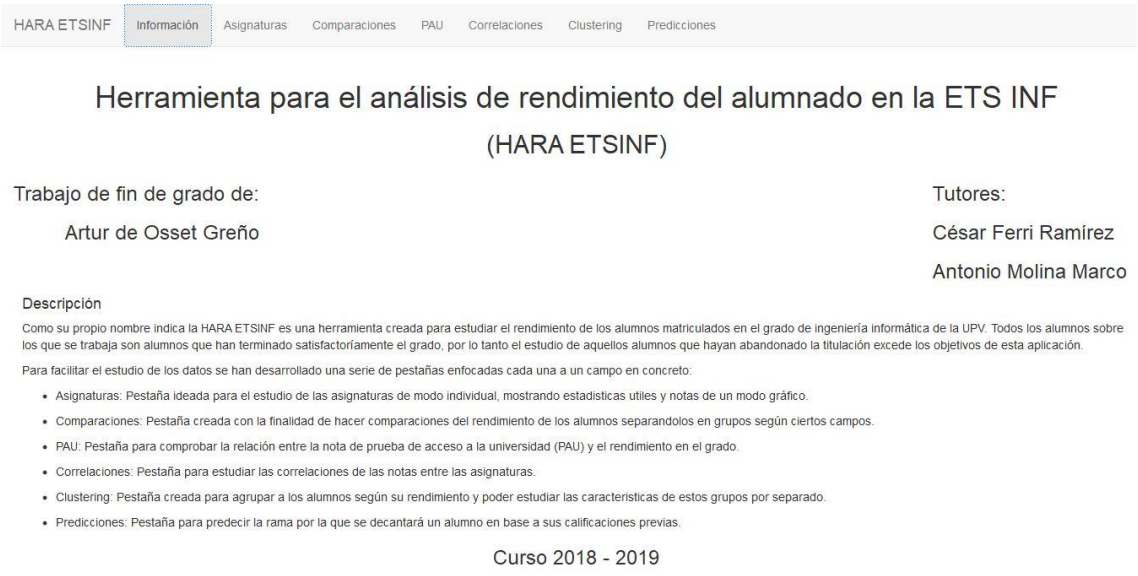


Ilustración 60 Pestaña de información

La aplicación se encuentra actualmente desplegada en el siguiente enlace:

https://ardeos.shinyapps.io/tfg-3/

Capítulo 6: Análisis de resultados con la aplicación

Con la aplicación desarrollada es el momento de poner en práctica su utilidad realizando diverso análisis. Debido a la abrumadora cantidad de datos, este análisis se basó en un espectro de datos limitados. Se utilizaron solamente los alumnos que entraron en el año 2010 y el 2011.

Asignaturas

En primer lugar, se procede realizar un análisis utilizando la pestaña "Asignaturas". Como hay ciento treinta y cuatro asignaturas, se decidió solo utilizar las asignaturas de primero y segundo, en los cursos de 2010-2011 y 2011-2012.

Asignatura	Total mat.	No pres.	Mat. aprob.	Tasa rend.	Tasa éxito
FOE	240	5	193	80,42%	78,33%
EST	241	2	235	97,51%	96,68%
FFI	239	0	174	72,80%	72,80%
IIP	242	16	202	83,47%	76,86%
FCO	241	0	218	90,46%	90,46%
PRG	241	12	189	78.42%	73,44%
TCO	241	1	177	73,44%	73,03%
AMA	240	8	213	88,75%	85,42%
ALG	241	1	207	85,89%	85,48%
MAD	242	1	211	87,19%	86,78%

Tabla 3 Asignaturas primero curso 2010-2011

Inicialmente veamos las asignaturas de primero del curso 2010-2011. El número de matrículas es bastante semejante, lo normal en este caso. Si nos fijamos en la columna "no presentados" hay dos asignaturas que sobresalen: las de programación, seguidas por análisis matemático. Después nos saltamos la columna "matrículas aprobadas" y pasamos directamente a las tasas, que hacen cálculos utilizando esa columna que nos hemos saltado. La asignatura con mejor tasa de rendimiento (mat. aprobadas/mat. totales) es EST con amplia diferencia. Después estarían FCO, AMA y MAD. Se aprecia una cierta ventaja de las asignaturas matemáticas. En el otro extremo nos encontramos FFI, TCO y PRG, que son dos asignaturas centradas en el hardware y una de programación. Observando la tasa de éxito (mat. aprobadas/mat. presentados) se ve que solo hay una diferencia notable en tres asignaturas, IIP, PRG y AMA, teniendo una bajada de un 7, un 5 y un 3,5 por ciento con respecto a la tasa de rendimiento.

Asignatura	1ª mat	Total mat.	No pres.	Mat. aprob.	Tasa rend.	Tasa éxito
FOE	Si	233	1	229	98,28%	97,85%

FOE	No	45	0	43	95,56%	95,56%
EST	Si	232	0	216	93,10%	93,10%
EST	No	5	0	5	100%	100%
FFI	Si	232	0	197	84,91%	84,91%
FFI	No	63	0	60	95,24%	95,24%
IIP	Si	231	1	193	83,55%	83,12%
IIP	No	38	0	33	86,84%	86,84%
FCO	Si	232	0	217	93,53%	93,53%
FCO	No	22	0	22	100%	100%
PRG	Si	232	2	194	83,62%	82,76%
PRG	No	50	0	49	98%	98%
TCO	Si	230	3	185	80,43%	79,13%
TCO	No	62	0	59	95,16%	95,16%
AMA	Si	231	0	201	87,01%	87,01%
AMA	No	26	0	23	88,46%	88,46%
ALG	Si	230	3	203	88,26%	86,96%
ALG	No	32	0	30	93,75%	93,75%
MAD	Si	231	0	219	94,81%	94,81%
MAD	No	30	0	30	100%	100%

Tabla 4 Asignaturas primero curso 2011-2012

Pasemos a las asignaturas de primero del curso 2011-2012. En este caso hemos separado las estadísticas en primeras y segundas matriculas (indicado por la columna "1ª mat" y por el color gris de las filas). A primera vista se pueden afirmar varias cosas: Los alumnos de segunda matrícula se han presentado a todas las asignaturas; su tasa de rendimiento/éxito es muy elevada, por encima del 90% en todos los casos salvo dos que son IIP y AMA con un 87% y un 88% (redondeando); entre los nuevos alumnos de este curso hay menos no presentados, 10 entre todas las asignaturas, un número inferior al de alumnos no presentados de la segunda asignatura del año anterior con más no presentados; la tasa de rendimiento mínima es un 80%, un 8% más que la mínima del año anterior.

Asignatura	Total mat.	No pres.	Mat. aprob.	Tasa rend.	Tasa éxito
DYP	201	0	200	99,50%	99,50%
EDA	196	13	156	79,59%	72,96%
ETC	203	12	154	75,86%	69,95%
IPC	234	0	220	94,02%	94,02%
LTP	203	1	139	68,47%	67,98%
FSO	267	2	252	94,38%	93,63%
CSD	198	0	197	99,49%	99,49%
TAL	199	1	170	85,43%	84,92%
Inglés bajo	113	3	106	93,81%	91,15%
Inglés alto	78	3	63	80,77%	76,92%
RED	209	8	185	88,52%	84,69%

Tabla 5 Asignaturas segundo curso 2011-2012

Ahora volvamos a los alumnos de la primera promoción, o al menos a aquellos que pasaron a segundo al primer intento. Se puede ver que esta promoción sigue con unos niveles de no presentados relativamente elevados, con un máximo de 13 en EDA. En cuanto a tasas vemos que hay dos asignaturas con, en ambas tasas, un 99,50% es decir que de los doscientos alumnos (más o menos) matriculados solo ha suspendido uno. En el otro extremo tenemos a LTP con aproximadamente un 68% en las dos tasas, el mínimo visto en todas las asignaturas y cursos analizados. Justo detrás se encuentran EDA y ETC cuyas tasas de rendimiento son aproximadamente un 80% y un 76% respectivamente, pero cuyas tasas de éxito son unos seis puntos inferiores.

De esta pestaña se ha podido sacar mucha información y eso que no se han mirado aún los histogramas. En cualquier caso, conviene recordar que todos estos alumnos han finalizado satisfactoriamente el grado en ingeniería informática, por lo que no se cuenta aquellos alumnos que la hayan abandonado a mitad.

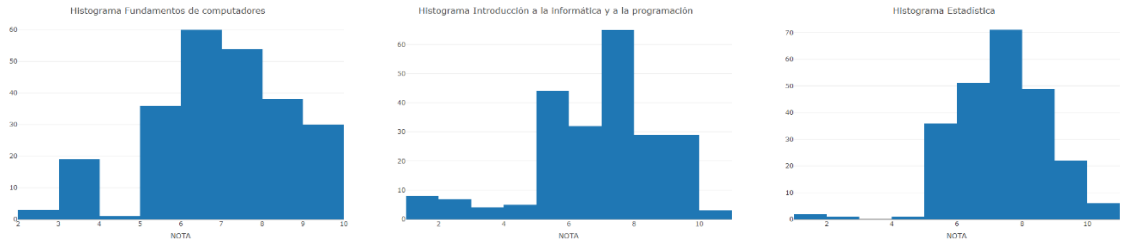


Ilustración 61 Histogramas de primero del curso 2010-2011

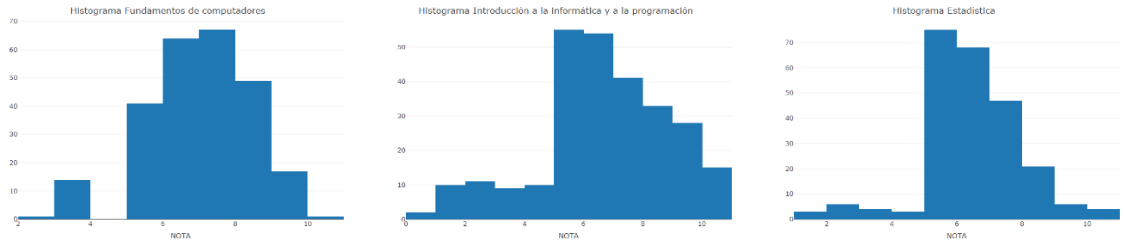


Ilustración 62 Histogramas de primero del curso 2011-2012

Aquí tenemos los histogramas de tres asignaturas de primero, de izquierda a derecha FCO, IIP y EST, curso 2010-2011 y curso 2011-2012 abajo. En base a la forma se puede distinguir a simple vista que en el curso 2011-2012 la proporción de notas bajas, aunque aprobadas, es mayor en general. Por otra parte, en FCO se pasa de casi treinta "dieces" hasta uno solo sin embargo en IIP se pasa de unos dos a casi veinte.

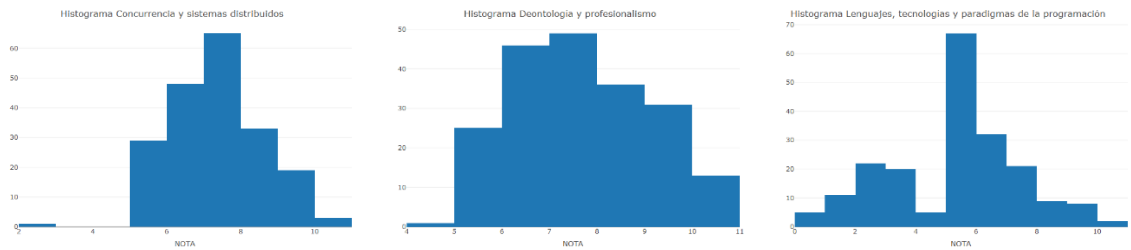


Ilustración 63 Histogramas de segundo del curso 2011-2012

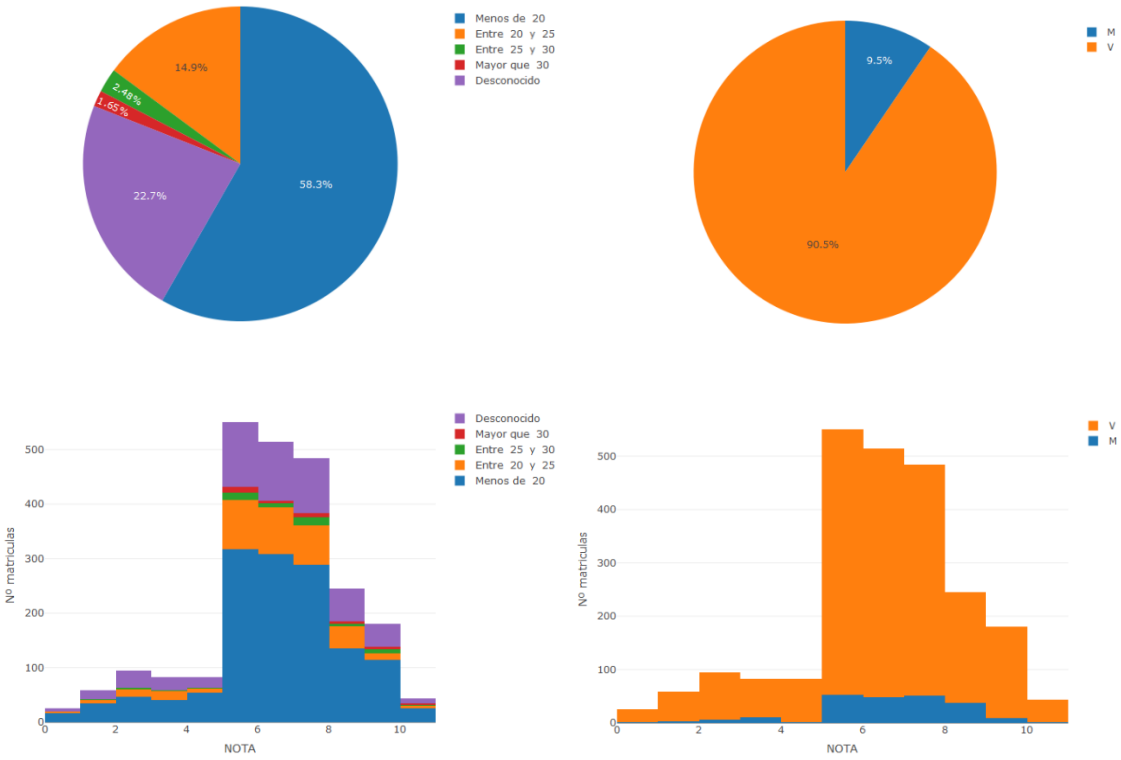
Del segundo curso seleccionamos las asignaturas con mejores tasas y la de peores tasas, es decir CSD, DYP y LTP. Nada más mirar resulta evidente que LTP no solo tiene

una peor calidad en las tasas, sino también en las notas. CSD y DYP son bastante semejantes en cuanto a notas, destacando levemente DYP en el número de “dieces”.

Comparaciones

Continuemos con la pestaña de comparaciones. Se analizó lo mismo, primero y segundo en los cursos 2010-2011 y 2011-2012, estudiando el bloque en conjunto sin centrarnos en asignaturas en específico.

Curso 2010-2011, primero



alumnos	grupo	media	min	max
141	Menos de 20	6.76	5.10	9.61
36	Entre 20 y 25	6.64	5.42	8.23
6	Entre 25 y 30	7.01	5.87	8.43
4	Mayor que 30	7.30	6.25	8.36
55	Desconocido	6.85	5.61	9.26

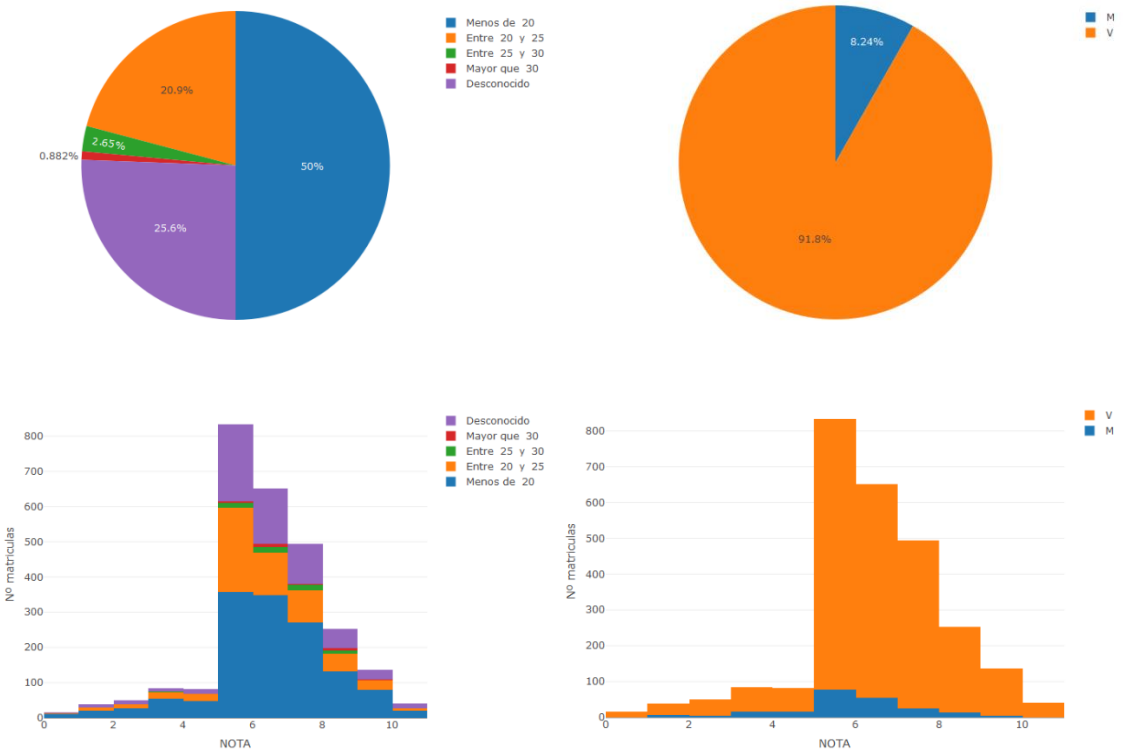
alumnos	grupo	media	min	max
23	M	6.82	5.65	8.23
219	V	6.77	5.10	9.61

Ilustración 64 Datos de comparaciones de primero del curso 2010-2011

Vamos a analizar solo dos campos, edad y género. Mirando los gráficos de tarta se puede comprobar algunas cosas, la primera de ellas que más de la mitad de los alumnos entraron con menos de veinte años. La segunda que el segundo “intervalo” de edad más numeroso es desconocido, lo que significa que hay un 22% de alumnos sin edad registrada. Lo tercero es que se confirma que, de momento, la informática es una carrera prominentemente masculina, con un 9,5% de mujeres.

Mirando los histogramas y las tablas no hay mucho que destacar, todos los grupos son bastante semejantes, medias ligeramente superiores en los grupos más pequeños, salvo en el intervalo de edad de 20 a 25.

Curso 2011-2012, primero



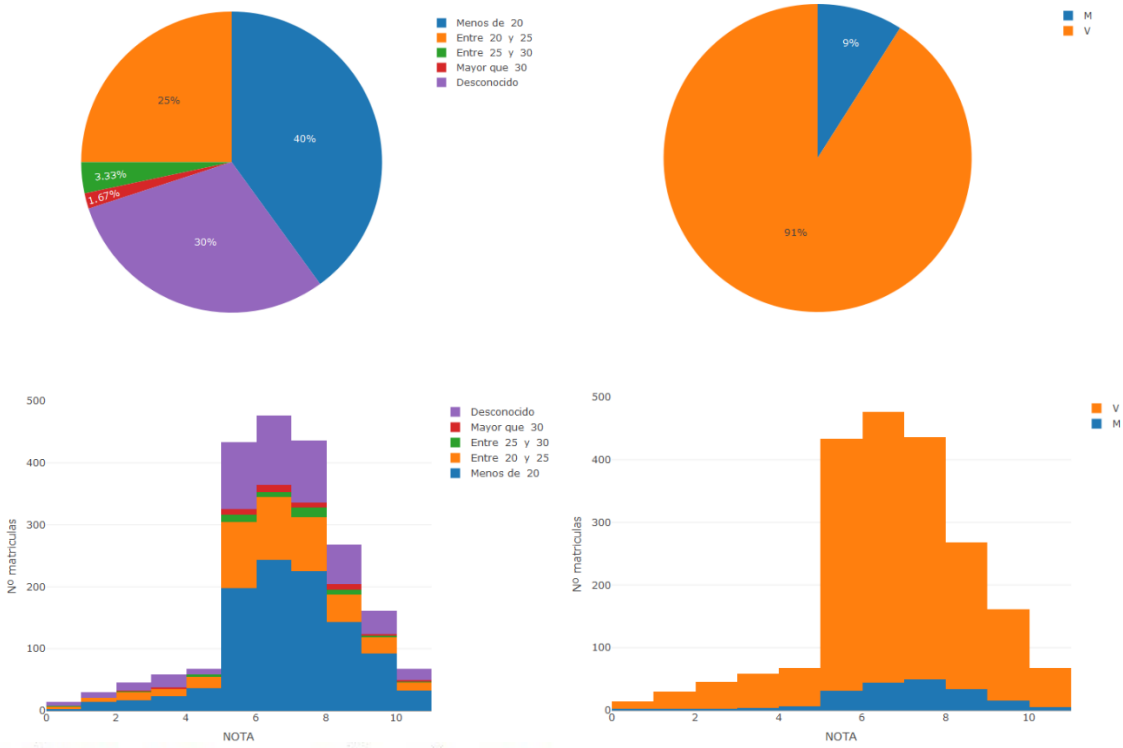
alumnos	grupo	media	min	max
170	Menos de 20	6.60	5.00	9.84
71	Entre 20 y 25	6.36	5.17	8.63
9	Entre 25 y 30	6.78	5.58	7.95
3	Mayor que 30	7.32	6.61	8.06
87	Desconocido	6.42	5.27	9.61

alumnos	grupo	media	min	max
28	M	6.24	5.00	8.10
312	V	6.54	5.10	9.84

Ilustración 65Datos de comparaciones de primero del curso 2011-2012

Analizando los mismos campos se puede observar que prácticamente se cumplen los mismos patrones que el año anterior. Lo más destacable es el descenso hasta el 50% de los alumnos con una edad menor de 20 años, pero puesto que el número de alumnos con edad desconocida no solamente no ha menguado si no que se ha incrementado puede no ser más que un error por falta de datos. También se puede apreciar un cierto empeoramiento en las notas, especialmente en el caso de las alumnas, llegando a no haber un diez de una alumna en todo primero.

Curso 2011-2012, segundo



alumnos	grupo	media	min	max
120	Menos de 20	6.90	5.10	9.37
75	Entre 20 y 25	6.95	5.00	9.50
10	Entre 25 y 30	6.52	5.15	8.21
5	Mayor que 30	6.92	5.84	8.78
90	Desconocido	6.76	5.00	9.59

alumnos	grupo	media	min	max
27	M	6.95	5.25	9.04
273	V	6.85	5.00	9.59

Ilustración 66Datos de comparaciones de segundo del curso 2011-2012

En cuanto a los alumnos de segundo, una vez más, siguen la misma línea. Ha habido una muy leve (un 0,1 aprox.) mejoría en las medias de los grupos por género y también en los intervalos más jóvenes de edad, cayendo los mayores, bajando la media entre 0,4 y 0,5 puntos. Mención aparte para el valor de edad desconocido, alcanzando un porcentaje del 30%.

PAU

Se pasa a estudiar el grupo de alumnos en la pestaña PAU. La aplicación permite cambiar el grupo que da color a los puntos, pero en este análisis superficial solamente se utilizara el campo género.

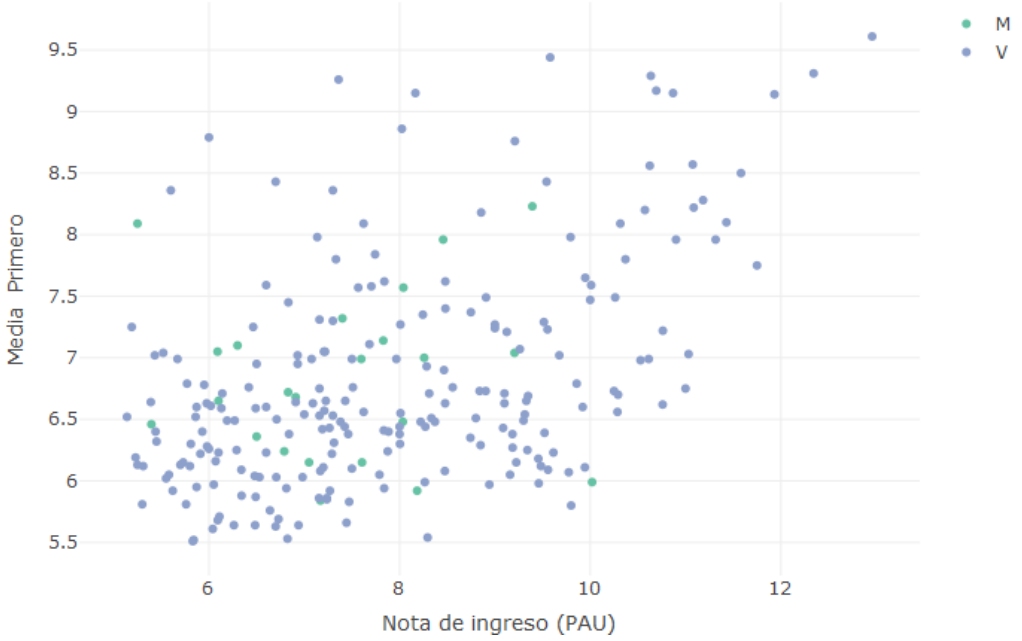


Ilustración 67 Tabla de notas medias de primero y PAU de los alumnos de 2010

Comparando con la media de primero es fácil ver que hay una correlación entre las notas comparadas. También es digno de mención que hay más alumnos con una media de primero mayor en proporción a su nota PAU. En cuanto a separación por género, lo más destacable es el hecho de que no hay ni una sola alumna con una nota de PAU mayor a 10 y la que tiene el diez el primer curso obtuvo una media proporcional algo inferior como es un 6.

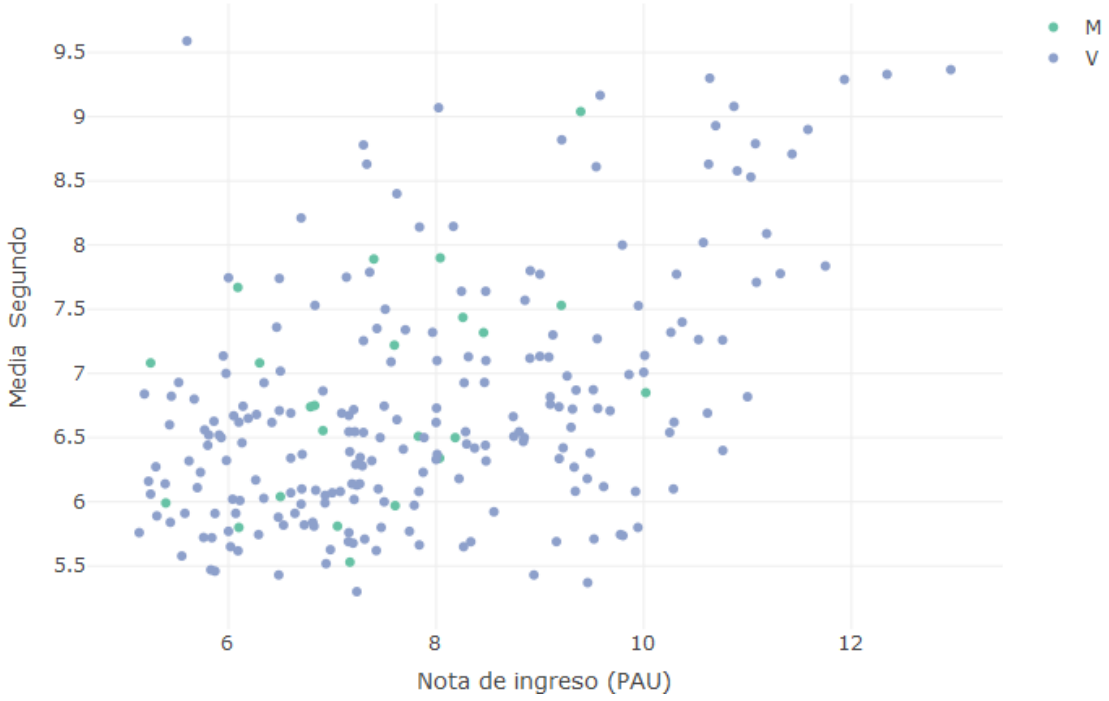


Ilustración 68 Tabla de notas medias de segundo y PAU de los alumnos de 2010

En el segundo curso parece que los puntos se compactan en torno a una línea central imaginaria. El caso concreto visto antes, la alumna con nota de PAU 10 ha aumentado su nota hasta casi un siete. Además, hay otro caso digno de mención, el alumno con nota PAU de 5,6 que ha sacado un 9,6 en la media de segundo. Buscando su nota media de primero en la gráfica anterior vemos que era un 8,4 mostrando una gran mejoría.

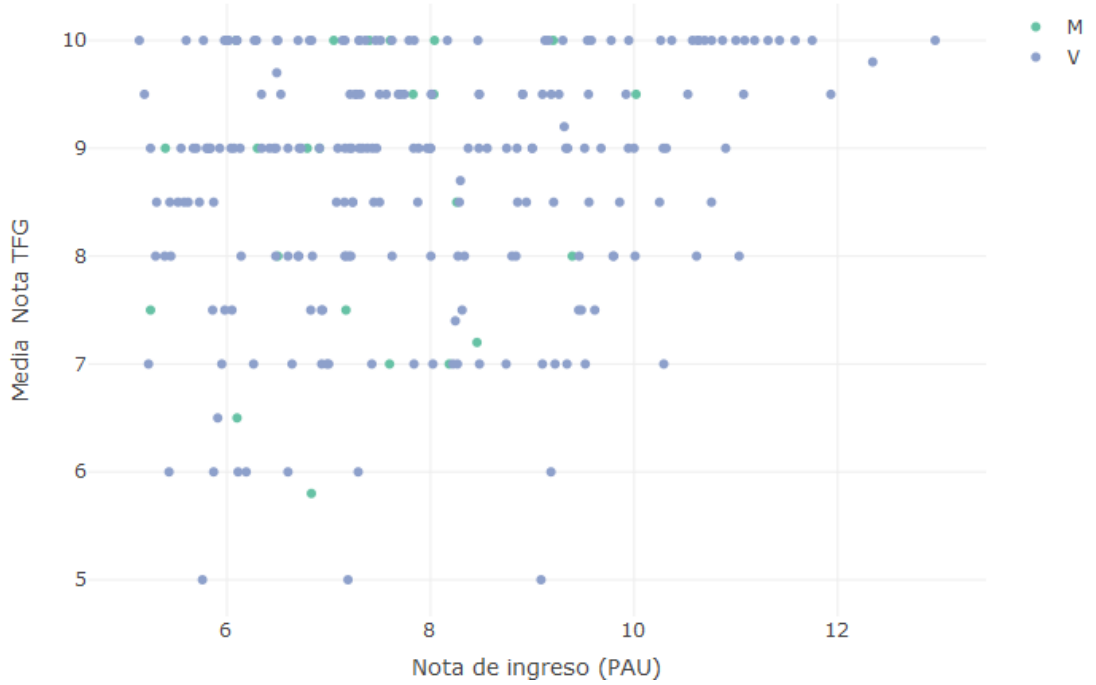


Ilustración 69 Tabla de notas de TFG y PAU de los alumnos de 2010

En vez de estudiar todas las notas medias de cada curso pasemos directamente a la nota del TFG. A simple vista se puede comprobar la tendencia de unas notas altas en los TFG, con muy pocos casos en la parte inferior de la línea imaginaria que va desde la esquina inferior izquierda a la superior derecha. Observando nuestros dos casos concretos vemos que el alumno de nota PAU 5,6 ha sacado un 10 y la alumna de nota PAU 10 ha sacado un 9,5.

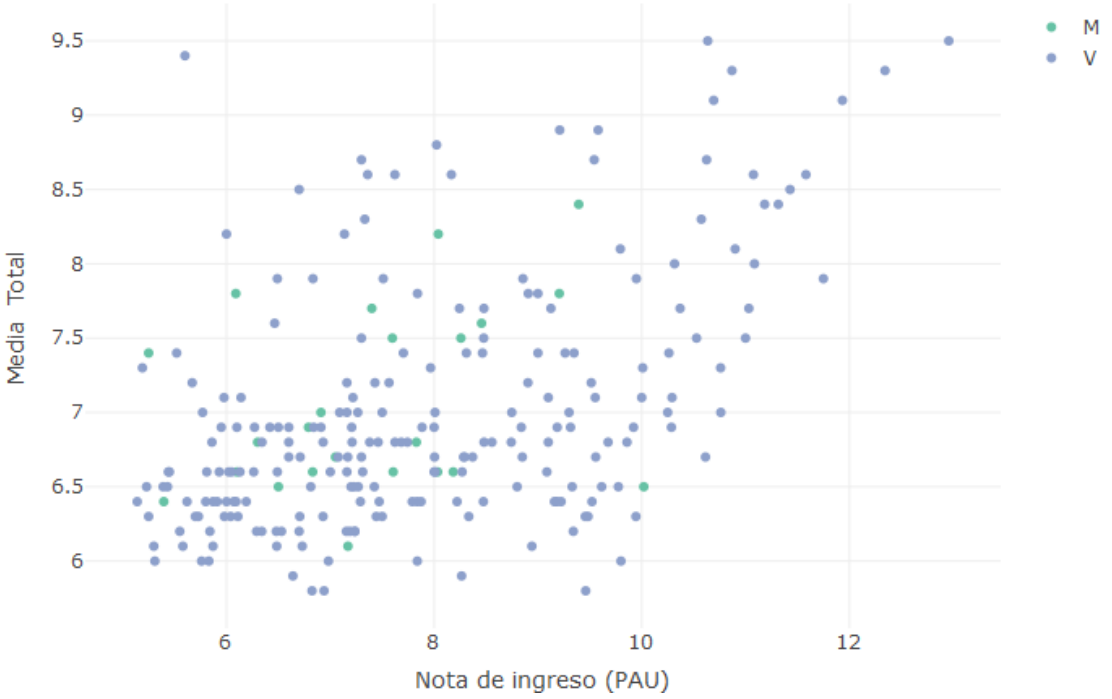


Ilustración 70Tabla de notas medias totales y PAU de los alumnos de 2010

Observemos las medias totales ahora. La distribución de los puntos es semejante a la de los primeros cursos. Estudiando casos concretos destacaré cuatro. El primero, la alumna de nota PAU 10, sacando nada más que una media de 6,5. Segundo el alumno de nota PAU 5,6 siendo lo opuesto del anterior caso, teniendo una nota media total de 9,4. Tercero la alumna con mejor nota media, un 8,4 teniendo en la PAU un 9,4 la segunda mejor de las alumnas. Por último, un caso bastante claro, el estudiante con mejor nota media y PAU, 9,5 y 13 respectivamente.

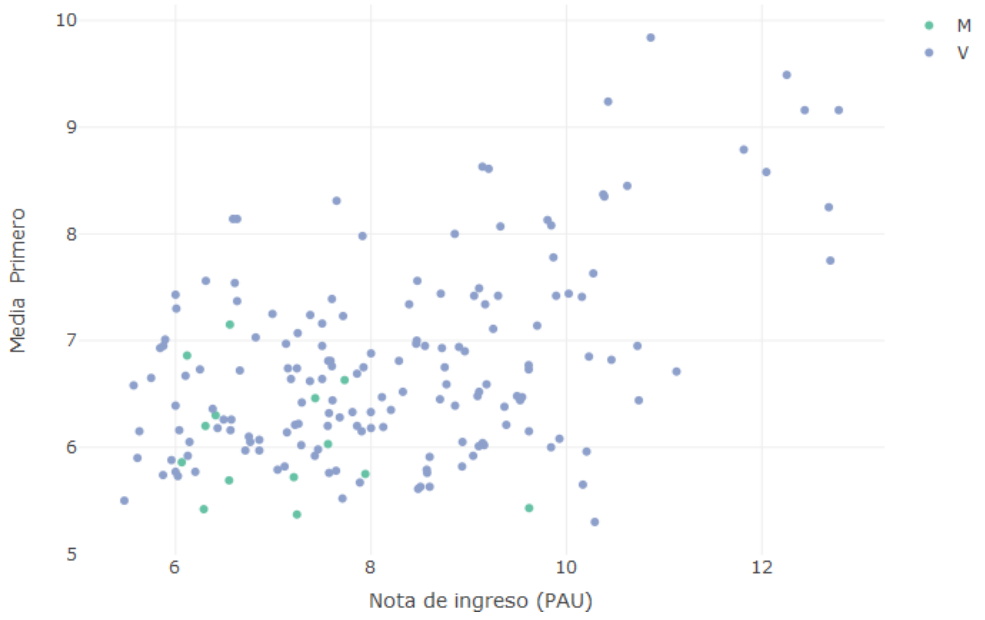


Ilustración 71 Tabla de notas medias de primero y PAU de los alumnos de 2011

Con respecto a los alumnos que entraron en 2011 parece que tienen unas notas medias ligeramente inferiores. Este grupo resulta peor para las alumnas, sacando la de nota PAU más alta (un 9,6) una media de 5,4 en primero y la de nota más alta en general un 7,2. Es destacable un alumno con una nota media de primero de 9,8 la mayor media vista hasta el momento.

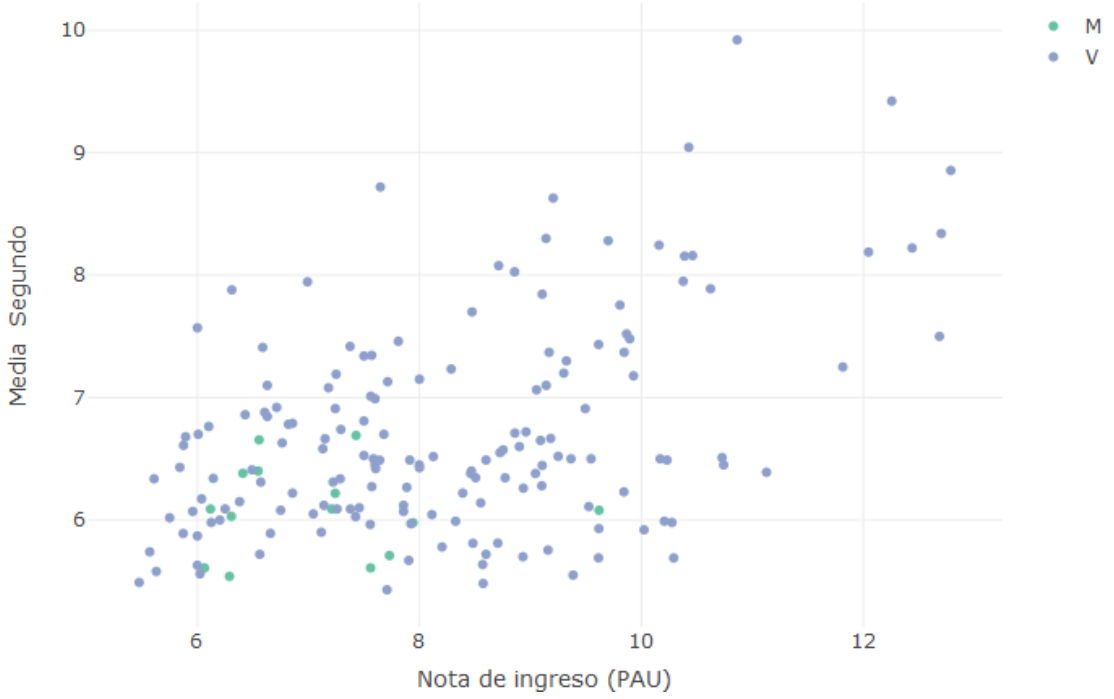


Ilustración 72 Tabla de notas medias de segundo y PAU de los alumnos de 2011

En segundo las notas parecen seguir el mismo patrón que en primero, al menos en los alumnos masculinos, en los alumnos femeninos las cosas parecen empeorar, siendo la media más alta de segundo un 6,7.

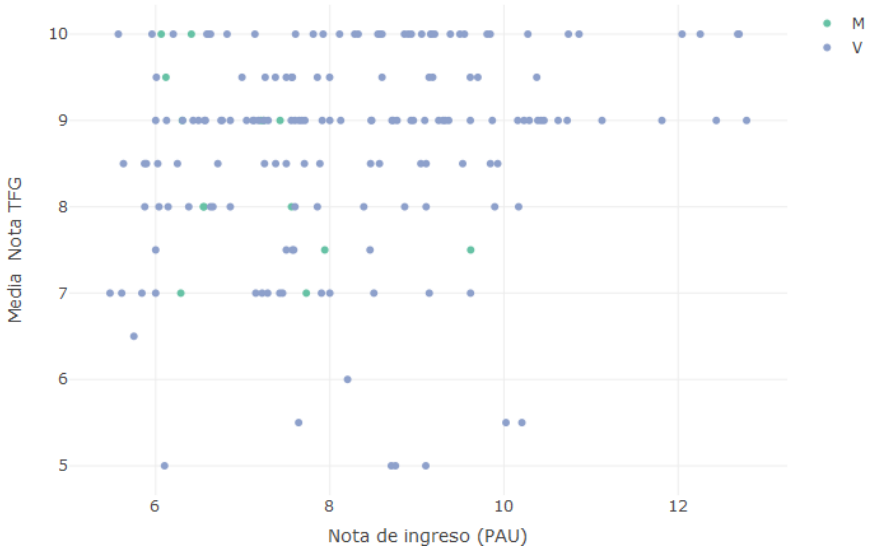


Ilustración 73 Tabla de notas de TFG y PAU de los alumnos de 2011

En cuanto a notas de TFG la tendencia sigue siendo notas altas. En el caso de las alumnas casi parece haber una correlación inversa con respecto a la nota PAU.

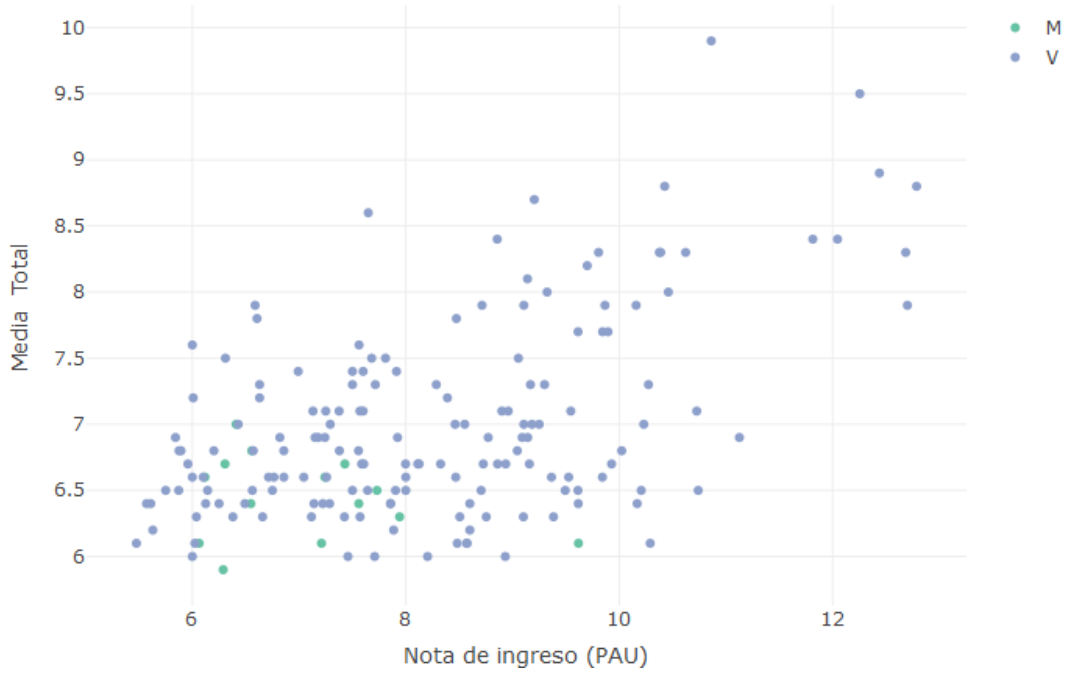


Ilustración 74 Tabla de notas medias totales y PAU de los alumnos de 2011

Finalmente llegamos a la nota media total de los alumnos de 2011. De nuevo en el caso de los hombres vemos casos de todo tipo, sin embargo, en mujeres los resultados no son demasiado alentadores, siendo su mayor nota media un 7.

Correlaciones

Como en esta pestaña no se pueden seleccionar cursos o alumnos se ha elegido analizar aquellas asignaturas que en el análisis de la pestaña “Asignaturas” parecieran tener una correlación, para comprobar hasta qué punto es así y algunas asignaturas de temática semejante. En las tablas/gráficas de correlaciones aparecen en ocasiones asteriscos, estos indican la relevancia de la correlación, cuantos más hay más relevante es.

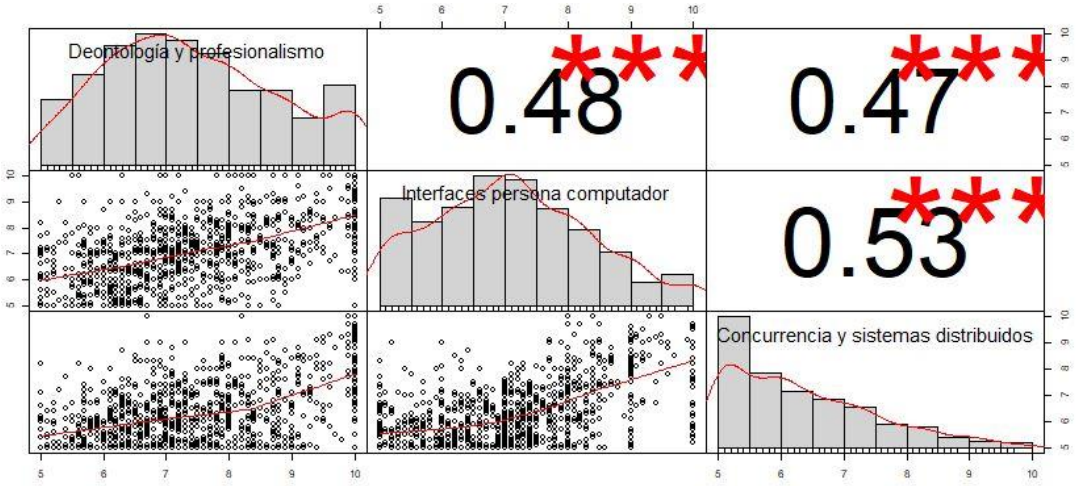


Ilustración 75 Correlaciones entre DYP, IPC y CSD

En el análisis de las asignaturas de segundo DYP, IPC y CSD son aquellas con mejores tasas. Se puede observar que existe correlación, aunque no es de las mayores posibles.

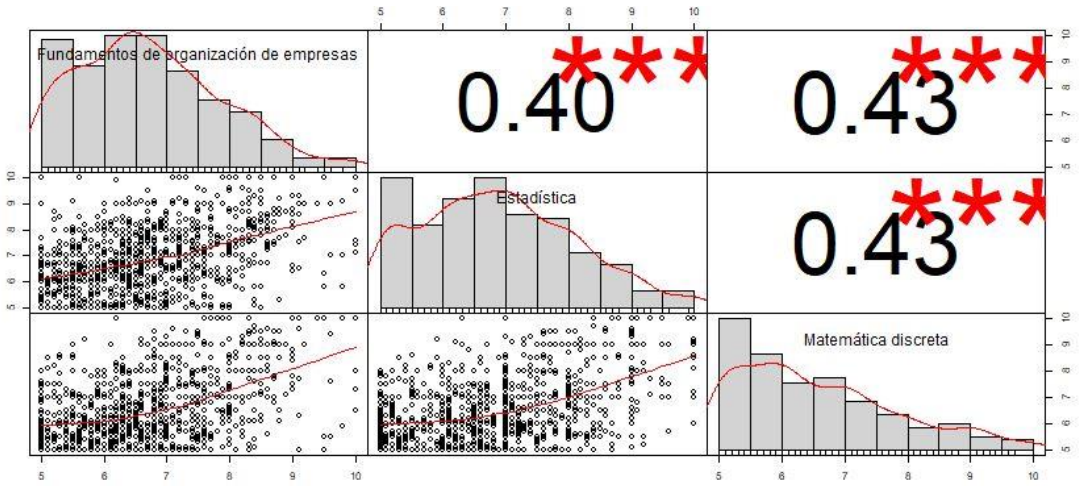


Ilustración 76 Correlaciones entre FOE, EST y MAD

En el análisis de asignaturas de primero del curso de 2010-2011 las asignaturas con mejores tasas fueron FOE, EST y MAD. Como podemos comprobar la correlación es algo menor en este caso.

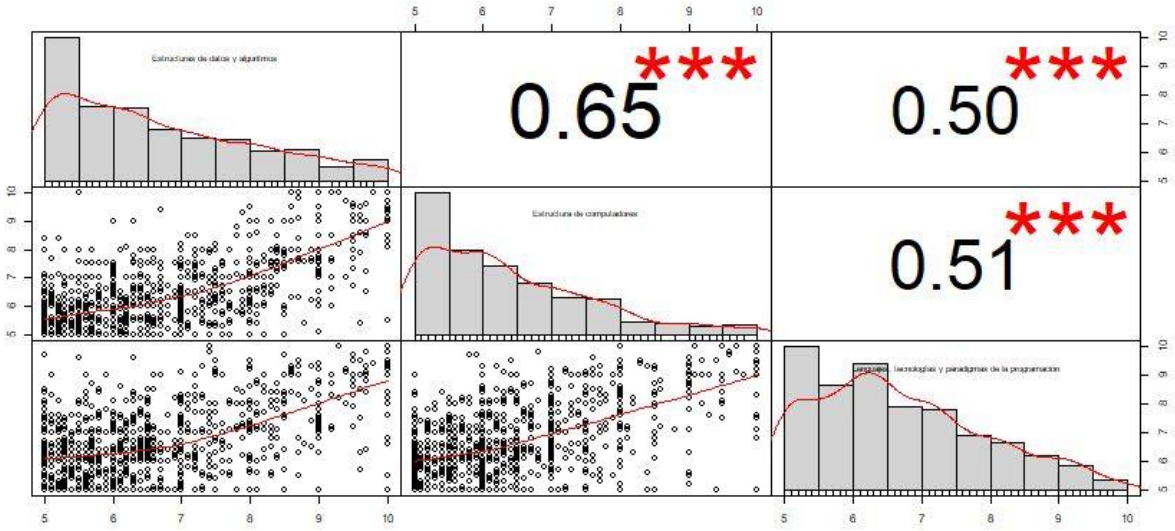


Ilustración 77 Correlaciones entre EDA, ETC y LTP

En el análisis de asignaturas de segundo las asignaturas con peores resultados fueron EDA, ETC y LTP. Entre estas asignaturas se puede ver una correlación más o menos media entre ETC y LTP, así como entre EDA y LTP, ambas alrededor de 0,50. Por último hay una correlación bastante alta, EDA con ETC alcanzando el 0,65.

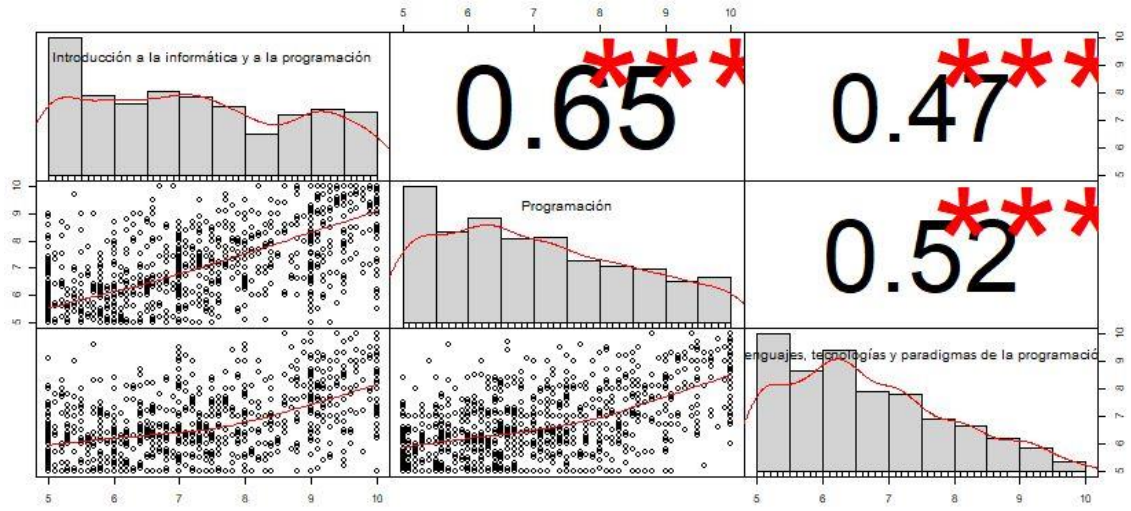


Ilustración 78 Correlaciones entre IPP, PRG y LTP

Pasando a bloques de temática podemos comprobar que entre IIP, PRG y LTP, tres asignaturas centradas en la programación, las correlaciones son muy semejantes a las de la comparación anterior, destacando la relación entre IIP y PRG.

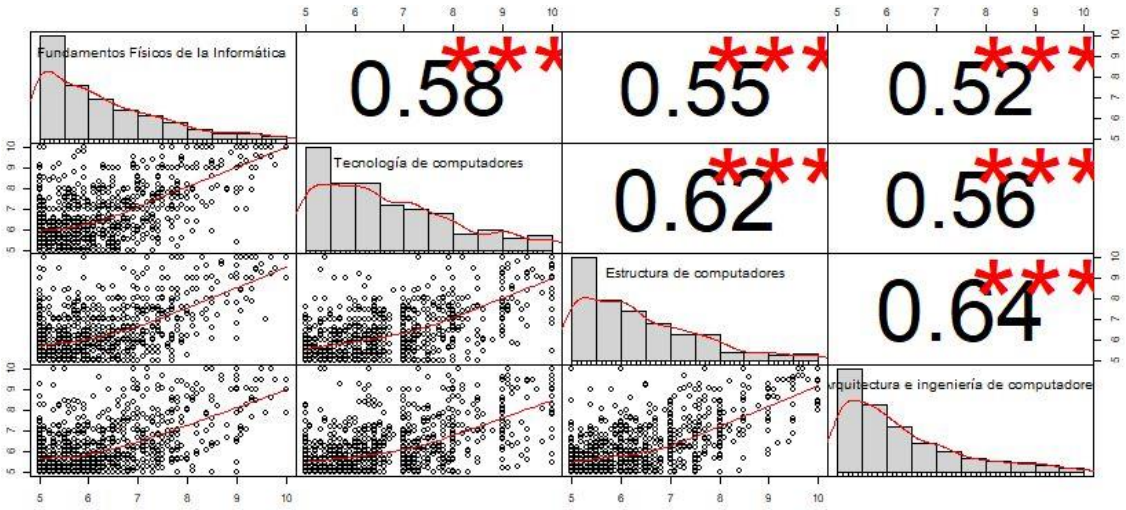


Ilustración 79 Correlaciones entre FFI, TCO, ETC y AIC

Comparando asignaturas de hardware, FFI, TCO, ETC y AIC podemos comprobar que hay una correlación más igualada entre todos los elementos comprados, todos superando el 0,50.

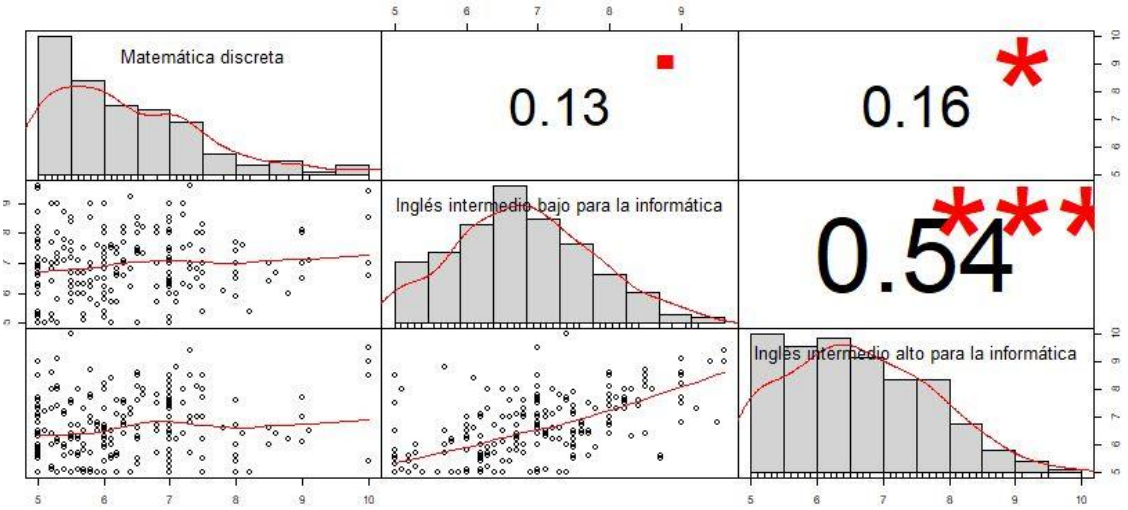


Ilustración 80 Correlaciones entre MAD, inglés bajo e inglés alto

Por último, se comprarán dos asignaturas de misma temática (inglés niveles bajo y alto) y una de diferente campo (MAD). Solamente observando las correlaciones es fácil saber cuándo se comparan las dos asignaturas de inglés, teniendo una correlación de 0,54 mientras que las demás no llegan al 0,20.

Clustering

Como en la pestaña de clustering se puede volver a filtrar por años de entrada, se vuelve a los dos grupos de estudios que se han estado analizando, los alumnos entrados en 2010 y los entrados en 2011. En ambos casos se ha optado por hacer cuatro “clusters”.

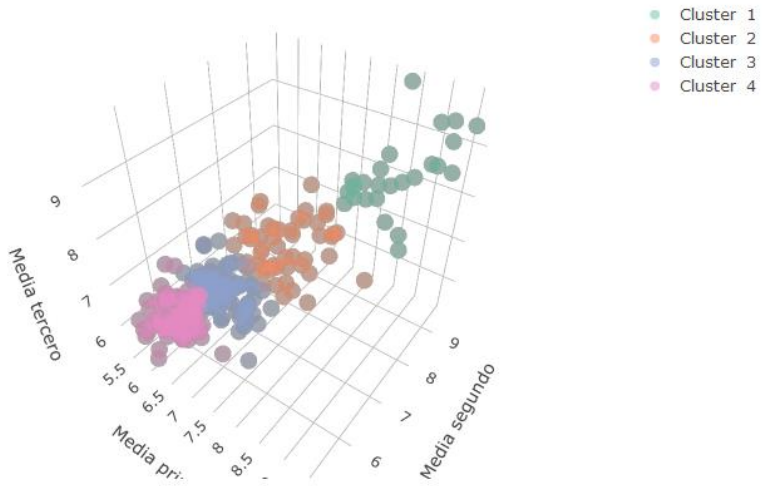


Ilustración 81 Gráfica 3d clusters de alumnos de 2010

Centroides

	Media primero	Media segundo	Media tercero	Nº alumnos
1	8.68	8.77	8.52	25
2	7.35	7.41	7.28	51
3	6.61	6.54	6.46	91
4	6.06	5.89	6.05	75

Ilustración 82 Tabla de centroides de alumnos de 2010

Con la gráfica 3d y la tabla de centroides se puede ver que se los grupos se han generado con una numeración inversa a la nota, además los dos grupos de menor cantidad de alumnos son los de mayor nota, mientras que el más numeroso no llega a ser el de menor nota.

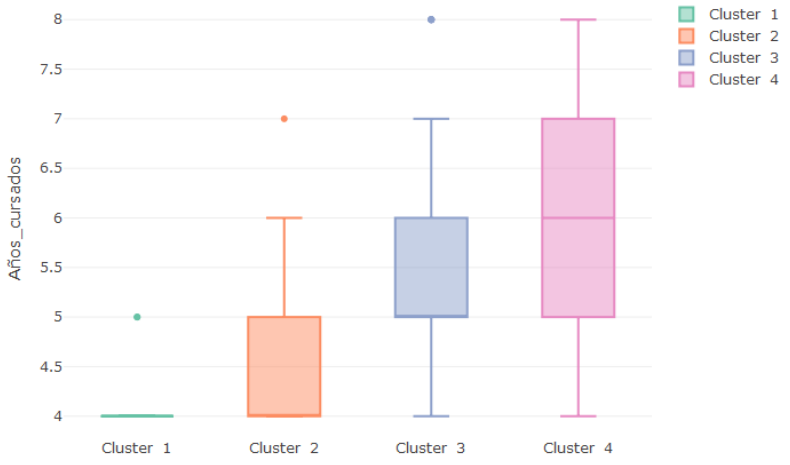


Ilustración 83 Gráfica de años cursados de clusters de alumnos de 2010

Observando los años cursados salta a la vista que a mejores notas más rápido se acaba la carrera.

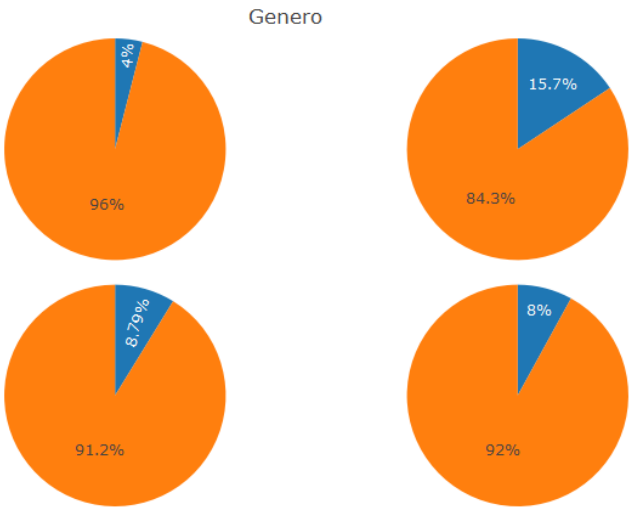


Ilustración 84 Gráficas de género de clusters de alumnos de 2010

Antes de entrar a analizar los porcentajes por género se procede a aclarar que en la aplicación la información de los gráficos de tarta agrupados aparece al pasar el cursor por encima, por lo tanto, la imagen exportada puede resultar un poco confusa. En cualquier caso, el orden de las gráficas siempre es el mismo, representan a los clusters de izquierda a derecha y de arriba a abajo. En este caso el color naranja representa a los hombres y el azul a las mujeres. Se aprecia que el grupo con menos proporción de mujeres, un 4%, es el de mayor nota mientras que el de mayor proporción es el de segunda mejor nota.

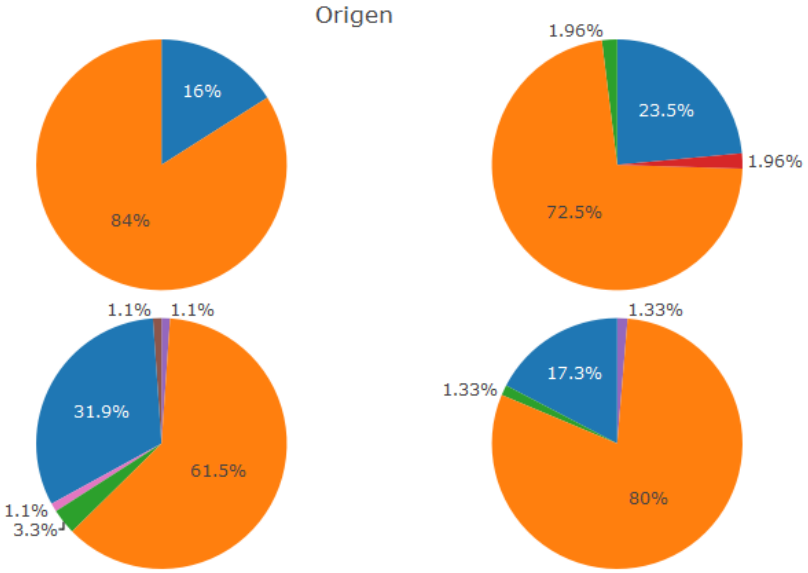


Ilustración 85 Gráficas de orígenes de clusters de alumnos de 2010

Colores: naranja: PAU, Azul: ciclo formativo, verde: extranjero de la UE, rosa: extranjero de fuera de la UE, morado: mayor de 25 años, marrón: mayor de 40 años, rojos: titulados.

En este caso se ve que los alumnos se pueden dividir en dos grandes grupos PAU y ciclo, pues el resto de orígenes no tienen una muestra lo suficientemente grande como para ser representativa. De estos dos grupos el primero resulta ampliamente mayor, destacando sobre todo en los grupos de notas más bajas y altas, mientras que los alumnos de ciclo formativo se concentran en mayor densidad en el grupo más numeroso.

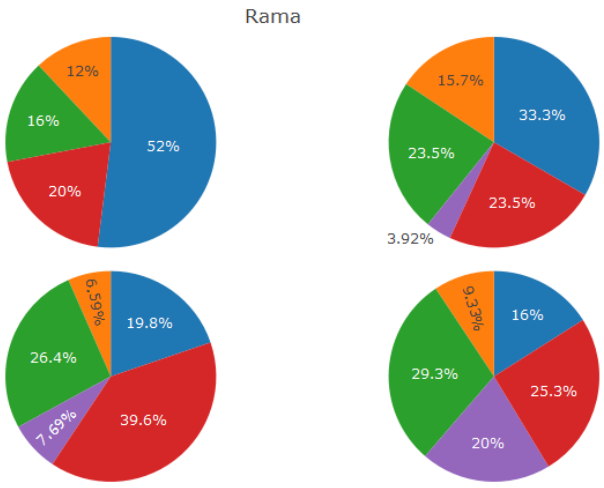


Ilustración 86 Gráficas de ramas de clusters de alumnos de 2010

Colores: Azul: computación, morado: tecnologías de la información, naranja: ingeniería de computadores, verde: ingeniería del software, rojo: sistemas de la información.

Centrándonos en las ramas de los alumnos se ve que hay una correlación entre la nota y el porcentaje de computación, llegando al 50% en los mejores alumnos. Después se puede observar una correlación inversa de la nota con ingeniería del software. De las restantes, tecnologías de la información es la menos concurrida seguida de ingeniería de computadoras. Sistemas de la información se mantiene sobre un 20% salvo en el grupo mayoritario donde se alza como la opción más común.

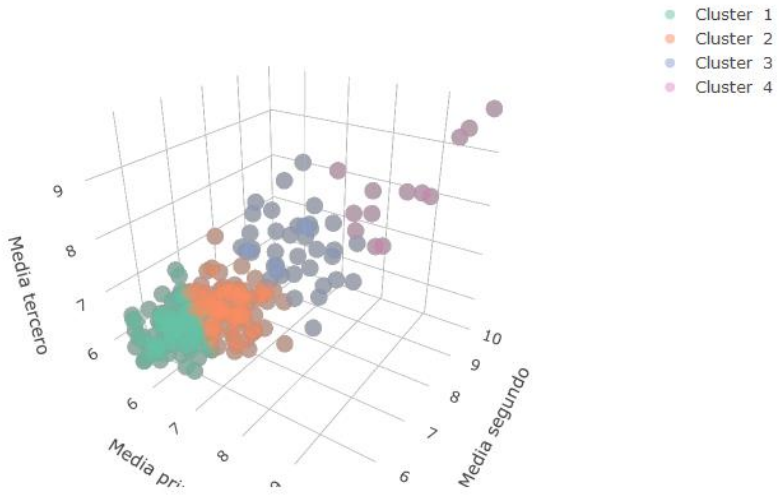


Ilustración 87 Gráfica 3d clusters de alumnos de 2011

Centroides

	Media primero	Media segundo	Media tercero	Nº alumnos
1	5.97	6.02	6.15	104
2	6.80	6.65	6.62	83
3	7.67	7.72	7.69	37
4	9.26	9.10	8.94	7

Ilustración 88 Tabla de centroides de alumnos de 2011

Pasando al siguiente grupo de estudio, lo primero que se comprueba es que los grupos de los alumnos matriculados en 2011 están ordenados de modo inverso, numeración y notas creciente. A continuación, nos fijamos en que la cantidad de notas elevadas es significativamente menor en comparación con los alumnos ingresados en el año anterior.

Ilustración 89 Gráfica de años cursados de clusters de alumnos de 2011

Si nos fijamos en los años cursados salta a la vista que hay alumnos que han tardado tres años en completar la carrera, lo más probable es que esto se deba al curso de adaptación a grado. Por otro lado, se observa que por lo general se repite el mismo patrón que en el anterior grupo analizado, aunque hay un empate en los cursos de notas intermedias.

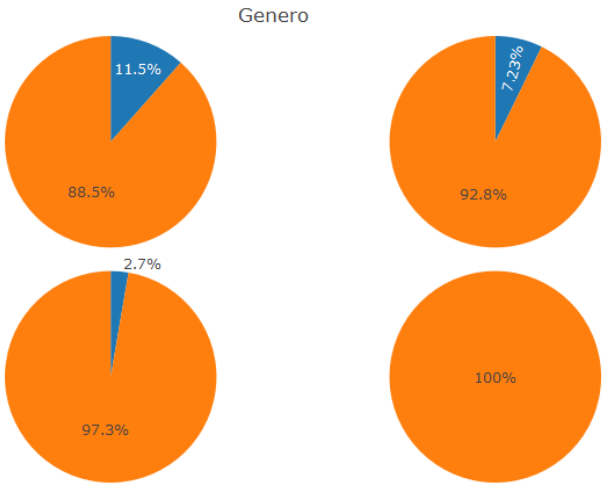


Ilustración 90 Gráficas de orígenes de clusters de alumnos de 2011

Colores: naranja: hombres, azul: mujeres.

Separando por sexo se comprueba que las alumnas que entraron en el año 2011 tienen un nivel bastante inferior con respecto a las que se matricularon en 2010, pues mientras estas su mayor porcentaje era en el grupo con segunda mejor media, las del año estudiado ahora se concentran en el grupo menos productivo, con presencia testimonial en el de segunda mayor nota y sin alcanzar el mejor.

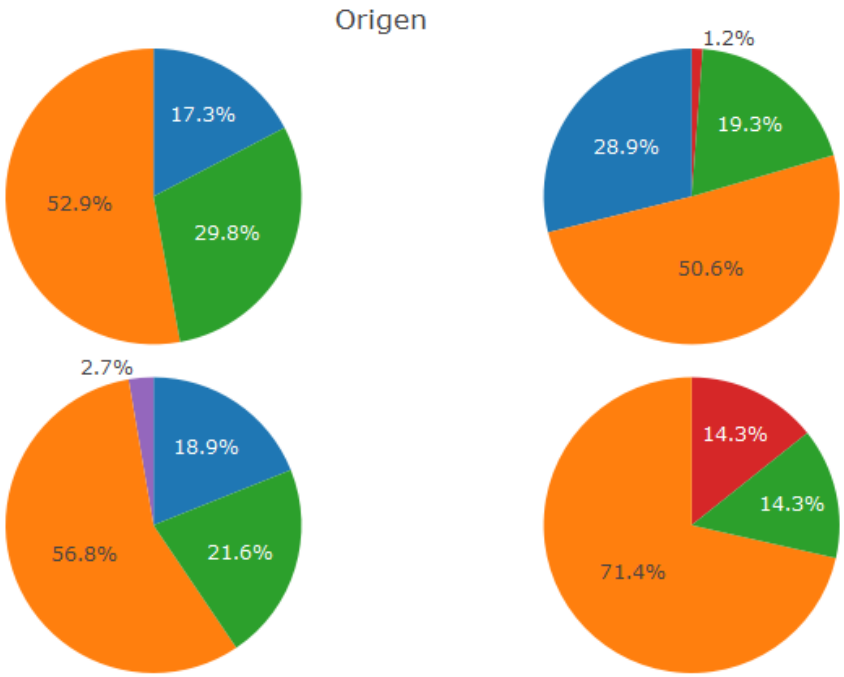


Ilustración 91 Gráficas de orígenes de clusters de alumnos de 2011

Colores: naranja: PAU, azul: ciclo formativo, verde: desconocido, rojo: mayores 25 años, morado: extranjeros de fuera de la UE.

En este caso la comparación por orígenes tiene un factor que no estaba en el anterior grupo, datos desconocidos y en un porcentaje no despreciable. De nuevo los alumnos de PAU son la mayoría en todos los grupos, sin embargo, los de ciclo dejan de aparecer en el grupo de mejores notas, donde un único alumno de "mayores de 25 años" y otro de origen desconocido completan el gráfico.

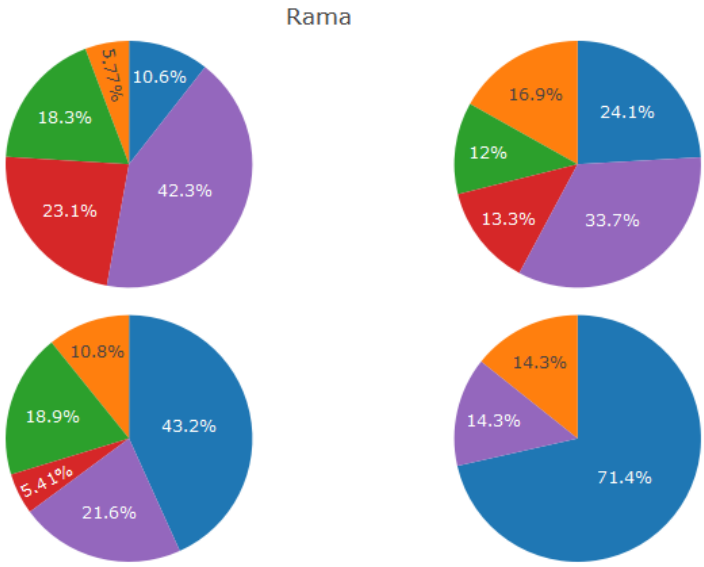


Ilustración 92 Gráficas de ramas de clusters de alumnos de 2011

Colores: Azul: computación, morado: tecnologías de la información, naranja: ingeniería de computadores, verde: ingeniería del software, rojo: sistemas de la información.

Comparando por rama se observa que se ha disparado el número de alumnos matriculados en tecnologías de la información, relegando al puesto de menos concurrida a ingeniería de computadores. Computación se mantiene como la opción predilecta de los alumnos más destacados. Sistemas de la información es más común cuanto menor es la nota, desapareciendo junto a ingeniería del software del grupo con mejor nota.

Predicciones

La pestaña de predicciones es un poco diferente a las demás, pues no sirve para analizar los datos de los alumnos ya matriculados, sino que los utiliza para predecir a que rama irán los nuevos alumnos.

Para estudiar este módulo se ha encontrado cuales son algunas las asignaturas más destacadas de cada rama.

Ilustración 93 Predicción con alta probabilidad de Computación

Como se puede observar en computación las asignaturas más destacadas son las de programación, es decir: IIP, PRG, EDA, IPC, LTP, FSO y CSD.

Ilustración 94 Predicción con alta probabilidad de Ingeniería de computadores

En ingeniería de computadores las asignaturas más influyentes son las de hardware: FFI, TCO y ETC.

Ilustración 95 Predicción con alta probabilidad de Ingeniería del software

Para conseguir una alta probabilidad en ingeniería del software se han utilizado asignaturas de campos variados: EST, IIP, PRG, ALG, EDA, y FSO.

Ilustración 96 Predicción con alta probabilidad de Sistemas de la información

Sistemas de la información destaca en las asignaturas matemáticas, como son: EST, AMA, ALG, MAD y TAL, además de DYP.

Ilustración 97 Predicción con alta probabilidad de Tecnologías de la información

También de diversas naturalezas son las asignaturas en las que destaca un alumno de Tecnologías de la información: FOE, TCO, MAD, DYP e IPC.

Por último, se realizaron otras dos predicciones, una con todo a cinco y otra con todo a diez.

Ilustración 98 Predicción de notas bajas

En el caso de los cincos lo más probable es que el alumno se decante por Tecnologías de la información, aunque tampoco sería raro que acabase en Ingeniería del software o Sistemas de la información.

Ilustración 99 Predicción de notas altas

En el supuesto completamente opuesto parece ser que computación es la opción indiscutida.

Capítulo 7: Conclusiones

Como se ha podido comprobar a lo largo de este proyecto se han cumplido todos los objetivos propuestos al inicio.

Conocer la evolución de los egresados de una titulación en relación a su rendimiento a lo largo de los estudios según distintos parámetros configurables (sexo, edad, cohorte, etc.)
Este objetivo se ha cumplido gracias a la posibilidad de poder analizar el rendimiento de los estudiantes tanto por asignaturas como por la comparación de las notas medias respecto a la nota de PAU.

Realizar agrupamientos y correlaciones bajo diversos parámetros.
La aplicación permite agrupar por diferentes campos del alumnado y realizar comparaciones entre ellos o agrupar según su rendimiento. Por otro lado, se permite realizar correlaciones entre asignaturas de todos los cursos y ramas.

Realizar predicciones sobre rendimientos futuros o posibles itinerarios de los estudiantes.
Las predicciones de rama también se realizan correctamente, tanto introduciendo los parámetros a mano como subiendo ficheros csv.

Impacto esperado

Espero que esta aplicación sea de utilidad para poder realizar diversos análisis estadísticos sobre los alumnos del Grado de ingeniería informática de la ETSINF, descubriendo patrones y relaciones útiles en un futuro.

Opciones de ampliación

Existen varias posibilidades de ampliación interesantes. Una de ellas es poder realizar más tipos de predicciones en base a los datos de los alumnos, como por ejemplo el número de años en los que un alumno dado tardará en acabar el grado, o que media de notas obtendrá un curso en base a las del anterior.

Otra opción es la de añadir funcionalidad para introducir nuevos datos o dar soporte a varias titulaciones, ya que tal y como está actualmente es preciso cambiar los ficheros de datos (ubicados en la subcarpeta “datos”) por otros con los nuevos registros a utilizar.

Por último, seria de interés crear una serie de informes automatizados para poder comprobar cierta información de un modo sencillo, como podría ser obtener las asignaturas con mayor correlación, por ejemplo.

Bibliografía

Hadley Wickham (2016). R for Data Science.

Hadley Wickham (2014). Advanced R.

Carson Sievert (2019). Interactive web-based data visualization with R, plotly, and shiny.

shinyapps.io team (2019). Shinyapps.io user guide.

Max Kuhn (2019). The caret Package.

Páginas web:

https://bbvaopen4u.com/es/actualidad/ventajas-e-inconvenientes-de-python-y-r-para-la-ciencia-de-datos

http://armillary-geomatica.blogspot.com/2015/04/comparativo-entre-sas-r-y-python.html

https://blogs.deusto.es/bigdata/r-vs-python-para-el-analisis-de-datos/

https://es.wikipedia.org/wiki/Ciencia_de_datos