



UNIVERSIDAD DE JAÉN
Nombre del Centro

Trabajo Fin de Grado

FRAMEWORK PARA EL DESARROLLO DE SISTEMAS DE RECOMENDACIÓN ATENTOS AL CONTEXTO

Alumno: Enrique Núñez Soto

Tutores: Prof. D. Luis Martínez López
Dña. Rosa María Rodríguez Domínguez.

Dpto: Departamento de Informática

Septiembre, 2021



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

D. Luis Martínez López y Dña. Rosa María Rodríguez Domínguez, tutores del Proyecto Fin de Carrera titulado: Framework para el desarrollo de sistemas de recomendación atentos al contexto, que presenta Enrique Núñez Soto, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Septiembre de 2021

El alumno:

Los tutores:

Enrique Núñez Soto

Luis Martínez López Rosa María Rodríguez Domínguez

Contenido

1. INTRODUCCION	10
1.1. Motivación	11
1.2. Propósito	11
1.3. Objetivos	12
1.4. Metodología de desarrollo	13
1.5. Planificación temporal	14
1.5.1. Diagrama de Gantt.....	14
1.6. Planificación de costes	15
1.6.1. Costes de desarrollo	15
1.6.1.1. Costes hardware.....	15
1.6.1.2. Costes software	16
1.6.1.3. Costes de personal.....	16
1.7. Estructura del TFG	18
2. SISTEMAS DE RECOMENDACIÓN.....	20
2.1. Introducción.....	21
2.2. Tareas básicas de los sistemas de recomendación	22
2.3. Elementos básicos de los sistemas de recomendación	24
2.3.1. Ítems	24
2.3.2. Usuarios.....	24
2.3.3. Valoraciones	24
2.4. Clasificación de los Sistemas de Recomendación	25
2.4.1. No personalizadas	25
2.4.2. Personalizadas	26
2.4.2.1. Recomendación basada en conocimiento.	26
2.4.2.2. Recomendación basada en contenido.....	27

2.4.2.3. Recomendación con filtrado colaborativo.	29
2.4.2.3.1. Algoritmo SVD.....	30
2.4.2.3.2. Algoritmo SVD aplicado al filtrado colaborativo	31
2.4.2.4. Recomendación híbrida.	32
2.4.2.5. Recomendaciones sensibles al contexto.	33
3. TECNOLOGÍAS UTILIZADAS	35
3.1. Introducción.....	36
3.2. ¿Por qué Python?.....	36
3.3. Herramienta de desarrollo: Spyder IDE.....	37
3.3.1. Características	37
3.3.2. Visualización de la interfaz	38
3.4. Sistemas de Información	41
3.5. ¿Qué son los SIG?	41
3.5.1. Estructura de los SIG	42
3.5.2. Formato KML.....	43
4. INGENIERÍA DEL SOFTWARE	46
4.1. Introducción a la Ingeniería del Software	47
4.2. Análisis del Sistema	49
4.2.1. Requisitos Funcionales.....	49
4.2.2. Requisitos No Funcionales	50
4.2.3. Casos de uso	51
4.2.3.1. Actores.....	51
4.2.3.2. Diagrama frontera	52
4.2.3.3. Caso de Uso 1: Crear archivos	53
4.2.3.4. Caso de Uso 2: Estudio de los datos	54
4.2.3.5. Caso de Uso 3: Realizar cross-validation	55
4.2.3.6. Caso de Uso 4: Introducir datos.....	56
4.2.3.7. Caso de Uso 5: Generar Recomendación.	57

4.2.3.8. Escenarios	58
4.3. Diseño del Sistema.....	63
4.3.1. Diagrama de Clases	63
4.4. Diseño de la interfaz.....	64
4.4.1. Metáforas	65
4.4.2. Storyboards	66
4.4.2.1. Storyboard Creación de los datos.....	67
4.4.2.2. Storyboard Análisis de los datos.....	68
4.4.2.3. Storyboard Cross-validation.....	69
4.4.2.4. Storyboard Recomendación	70
4.5. Implementación	71
4.5.1. Lenguaje del servidor.	71
4.5.2. Lenguaje del cliente.....	72
4.6. Pruebas y Validación.....	73
4.6.1. Casos de prueba	73
4.6.2. Resultados obtenidos	75
4.7. Análisis exploratorio de los datos	76
5. CONCLUSIONES	80
ANEXO I. Manual de Instalación.....	83
ANEXO II. Manual de Usuario	90
Bibliografía	99

Índice de Imágenes

Ilustración 1.1: Metodología SCRUM.

Ilustración 1.2: Diagrama de Gantt.

Ilustración 2.1: Esquema simplificado de SR.

Ilustración 2.2: Recomendación basada en contenido.

Ilustración 2.3: Recomendación con filtrado colaborativo basada similitud entre usuarios (izquierda), basada en similitud entre ítems (derecha).

Ilustración 2.4: Matriz S de valores singulares.

Ilustración 2.5: Paradigmas para la incorporación de contexto en sistemas de recomendación.

Ilustración 3.1: Gráfica de los lenguajes de programación más usados.

Ilustración 3.2: Logo de Spyder IDE.

Ilustración 3.3: Interfaz de Spyder.

Ilustración 3.4: Explorador de variables de Spyder.

Ilustración 3.5: Explorador de archivos de Spyder.

Ilustración 3.6: Ayuda de Spyder.

Ilustración 3.7: Termina de IPython en Spyder.

Ilustración 3.8: Historial de comandos en Spyder.

Ilustración 3.9: Captura del error cuadrático medio tras la ejecución de la validación de cruce.

Ilustración 3.10: Distribución de ítems por valoraciones.

Ilustración 3.11: Recuento valoraciones totales por lugar. (Top 10)

Ilustración 3.12: Recuento valoraciones totales por lugar. (Last 10)

Ilustración 3.13: Recuento valoraciones totales por usuario. (Top 10)

Ilustración 3.14: Recuento valoraciones totales por usuario. (Last 10)

Ilustración 3.9: Fichero POIJaen.kml

Ilustración 3.10: POI marcados en la provincia de Jaén desde Google MyMaps.

Ilustración 4.1: Diagrama frontera del sistema.

Ilustración 4.2: Diagrama de clases.

Ilustración 4.3: Diseño de la interfaz.

Ilustración 4.4: Salida de la ejecución del programa.

Ilustración 4.5: Metáfora de los POI.

Ilustración 4.6: Storyboard Creación de datos.

Ilustración 4.7: Storyboard Análisis de los datos.

Ilustración 4.8: Storyboard Cross-validation.

Ilustración 4.9: Storyboard Recomendación.

Ilustración 4.10: Fichero .csv importado a Excel.

Ilustración 4.11: Distribución de ítems por valoraciones.

Ilustración 4.12: Recuento valoraciones totales por lugar. (Top 10)

Ilustración 4.13: Recuento valoraciones totales por lugar. (Last 10)

Ilustración 4.14: Recuento valoraciones totales por usuario. (Top 10)

Ilustración 4.15: Recuento valoraciones totales por usuario. (Last 10)

Ilustración I.1: Ficheros de descarga de Python.

Ilustración I.2: Ventana de instalación de Python

Ilustración I.3: Ventana de instalación correcta de Python

Ilustración I.4: Visualización de los paquetes y versión instalados de Microsoft C++
Build Tools.

Ilustración I.5: Visualización de instalación de librería pykml desde el terminal.

Ilustración I.6: Visualización de instalación de librería pandas desde el terminal.

Ilustración I.7: Visualización de actualización del paquete setuptools desde el terminal.

Ilustración I.8: Visualización de instalación de librería surprise desde el terminal.

Ilustración I.9: Visualización de instalación de librería geopy desde el terminal.

Ilustración I.10: Visualización de instalación de librería folium desde el terminal.

Ilustración I.11: Visualización de instalación de librería Jinja2 desde el terminal.

Ilustración II.1: Visualización de la interfaz del programa.

Ilustración II.2: Visualización de las opciones de administrador.

Ilustración II.3: Visualización del fichero datosjaen.csv.

Ilustración II.4: Visualización del fichero ratings.csv.

Ilustración II.5: Visualización del fichero EstudioDatos.csv.

Ilustración II.6: Visualización del fichero Errores_cross_validation.csv.

Ilustración II.7: Ventana de la opción de usuario para realizar la recomendación.

Ilustración II.8: Visualización de la salida del programa al realizar la recomendación.

Índice de Tablas

Tabla 1.1: Tabla de costes de hardware.

Tabla 1.2: Tabla de costes de software.

Tabla 1.3: Tabla de costes de personal.

Tabla 1.4: Tabla total de los costes.

Tabla 2.1: Tabla de métodos de hibridación.

Tabla 4.1: Requisitos funcionales.

Tabla 4.2: Requisitos no funcionales.

1. INTRODUCCION

En este capítulo se redactará una introducción al Trabajo Final de Grado, que constará de una breve introducción a los Sistemas de Recomendación y al lenguaje de programación Python. En siguiente lugar se presentará el propósito y los objetivos del proyecto y finalmente se podrá visualizar la estructura del TFG y una planificación temporal del proyecto.

1.1. Motivación

Los sistemas de recomendación empezaron a utilizarse a partir de los años 90 y se utilizan para satisfacer las necesidades de un usuario y para ello se les proporcionará contenido que sea de su interés. En la actualidad cantidad de plataformas implementan este servicio, tanto en redes sociales como Twitter, plataformas de servicios multimedia como Netflix, Spotify o de e-commerce como Amazon.

El motivo de desarrollar este proyecto se debe a la necesidad que presentan hoy en día las empresas dedicadas a la venta de servicios turísticos y/o paquetes turísticos para encontrar un conjunto de elementos adecuados a las características de sus clientes de una forma simple, rápida y precisa. Este problema presenta una, importante complejidad debido a la gran cantidad de opciones posibles que proporcionar a los usuarios/clientes. Por lo que, la necesidad de sistemas inteligentes que automaticen y realicen la búsqueda de dichos elementos es una demanda en las empresas turísticas actuales.

1.2. Propósito

El propósito de este trabajo se centra en encontrar los puntos de interés (POIs) que más se ajusten a los intereses de un usuario, y poder así proporcionarle un listado de los n mejores lugares geográficos como recomendación de sus próximos destinos a visitar de acuerdo a sus necesidades, de manera que pueda visualizar en el mapa los lugares recomendados para cada usuario.

1.3. Objetivos

El propósito general anterior, se divide en los objetivos para poder crear un marco en el que se recomienden distintos lugares a un usuario incluyendo tanto la creación de los conjuntos de datos que vamos a utilizar y los distintos algoritmos para obtener la mejor recomendación.

- Obtención de los distintos datasets mediante el servicio web de Mymaps¹ de Google.
- Estudio de los datos para conocer las distribuciones de los mismos.
- Estudio de la precisión del sistema de recomendación mediante métricas adecuadas.
- Georreferenciación de las recomendaciones obtenidas para un usuario en concreto.

¹ Google MyMaps: servicio puesto en marcha por Google en abril del 2007, que permite a los usuarios crear mapas personalizados para uso propio o para compartir.

1.4. Metodología de desarrollo

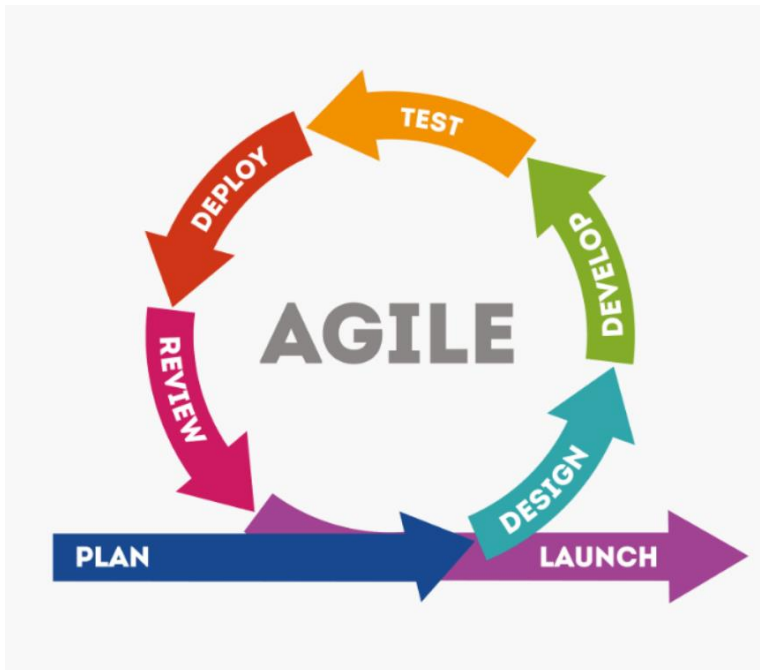


Ilustración 1.1: Metodología SCRUM.

Fuente = [<https://www.cognodata.com/blog/principios-metodologia-agile-desarrollo-proyectos/>]

Para elaborar el proyecto he utilizado la metodología SCRUM (Ilustración 1.1), que reduce el margen de error y los riesgos del proyecto, y consiste en los siguientes pasos:

1. **Definir el producto:** Definir los objetivos y los alcances del producto a desarrollar priorizando de acuerdo a la importancia y al valor que genere con respecto al coste y a las necesidades.
2. **Sprints:** El proyecto se divide en Sprints, es decir, cada una de las fases del proyecto donde se presentan los avances para prueba. A estos cambios se les asignan una prioridad y se decide si se ejecutan en el momento o se mandan a una lista de pendientes para ejecutarlo más adelante.
3. **Entrega del producto:** Una vez acabados todos los Sprints se hace la entrega del producto final.

1.5. Planificación temporal

1.5.1. Diagrama de Gantt

A continuación, se expone el tiempo y dedicación establecido en cada una de las diferentes tareas que está dividido el trabajo (Ilustración 1.2):

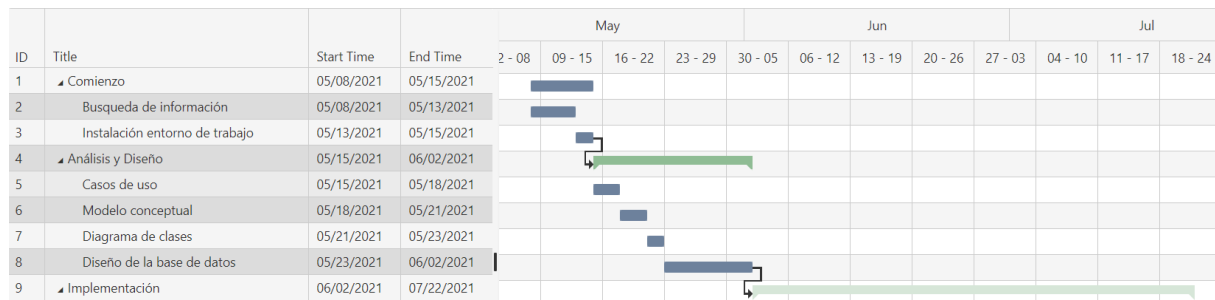


Ilustración 1.2: Diagrama de Gantt (1/2).

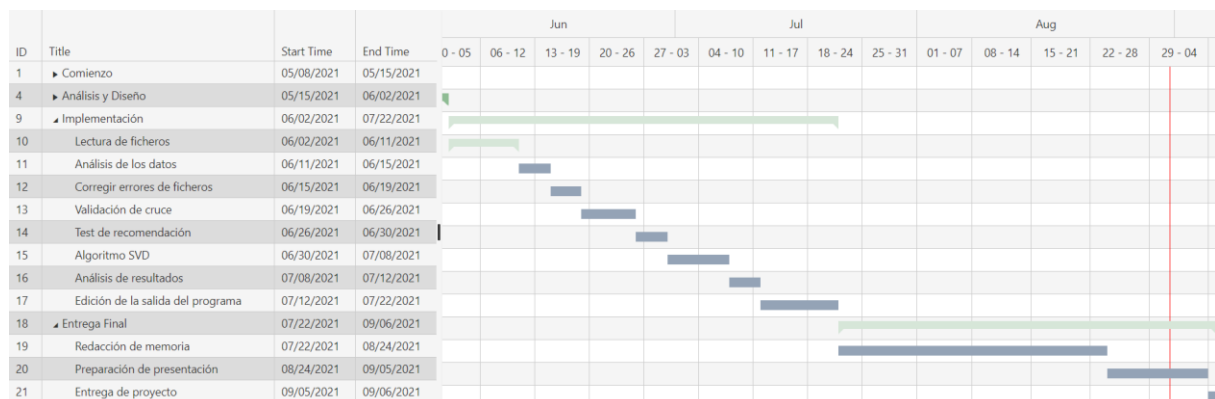


Ilustración 1.2: Diagrama de Gantt (2/2).

1.6. Planificación de costes

1.6.1. Costes de desarrollo

En este punto recogeremos los elementos físicos, de software, de personal necesarios para cumplir los objetivos, de manera que cuantificaremos cada elemento que se haya utilizado.

En este caso se realizará la amortización lineal del recurso y cuantificaré el coste de estos recursos dividiendo el precio de adquisición entre la vida útil y a su vez entre las horas de trabajo al año. Se calculará con la siguiente fórmula:

$$\text{Cuota} = ((\text{Precio} / \text{Vida útil})) / (\text{Horas utilizado})$$

1.6.1.1. Costes hardware

Se utiliza un ordenador portátil para el desarrollo, implementación y las pruebas necesarias (Tabla 1.1).

Concepto	Adquirido	Cuota	Horas	Coste
MSI PRESTIGE 15	1100 €	0.12 €/h	332	39,84 €
Total:				39,84 €

Tabla 1.1: Tabla de costes de hardware.

1.6.1.2. Costes software

El software utilizado para el desarrollo es el IDE Spyder y es gratuito, y para el diseño de los diagramas el Visual Paradigm², utilizo una versión de prueba, pero haré una estimación de costes con la versión Modeler con un precio de 99 \$ = 83,72 € (Tabla 1.2).

Concepto	Adquirido	Cuota	Horas	Coste
Visual Paradigm Modeler	83,72 €	0.25 €/h	332	21,38 €
Spyder IDE	0 €	0 €/h	332	0 €
Total:				21,38 €

Tabla 1.2: Tabla de costes de software.

1.6.1.3. Costes de personal

Al tratarse de un TFG, está elaborado por un solo alumno, en este caso programador Junior, y se requerirán los servicios de un analista de datos. (Tabla 1.3) Para calcular los salarios medios de nuestro personal nos apoyaremos en una página web que calcula el sueldo medio mediante estadísticas³.

Concepto	Adquirido	Cuota	Horas	Coste
Programador Jr	24.074 €/año	9,06 €/h	332	3.009,25 €
Analista de Datos	32.916 €/año	12,39 €/h	332	4.114,5 €
Cuota SS		31,05 %		2.211,92 €
Total:				9.335,67 €

Tabla 1.3: Tabla de costes de personal

² Visual Paradigm: herramienta proporciona un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación.

³ <https://es.indeed.com/?from=gnav-title-webapp> .

En la Tabla 1.4 se resume el costo de los 3 puntos anteriores descritos y la suma total de los costes para el desarrollo del proyecto.

Concepto	Coste
Costes de Personal	9.335,67 €
Software	21,38 €
Hardware	39,84 €
Total	9396,89 €

Tabla 1.4: Tabla total de los costes.

1.7. Estructura del TFG

A continuación, hare una introducción de los diferentes capítulos a tratar a lo largo del proyecto y de que consta cada uno de ellos.

En el primer capítulo se trata de la introducción del proyecto, la motivación de los Sistemas de Recomendación, el propósito que persigue, los objetivos a lograr y la metodología utilizada para realizar el proyecto. También se expondrá la planificación temporal mediante el Diagrama de Gantt y la planificación de los costes del proyecto.

En el capítulo 2 nos centraremos en los sistemas de recomendación y explicaremos todos los tipos que hay de sistemas de recomendación; las tareas básicas que realizan, los elementos básicos y los clasificaremos.

El capítulo 3 se dedicará a las tecnologías empleadas en el proyecto y comenzaremos explicando porque hemos escogido el lenguaje de programación de Python, que herramienta hemos utilizado para el desarrollo, sus características y la visualización de su interfaz. También desarrollaremos las pruebas y la validación realizadas y el análisis exploratorio de los datos que realizo en el proyecto. Concluiremos el capítulo con una breve descripción de los Sistemas de Información SI, los Sistemas de Información Geográfica SIG, y explicaré el formato kml en el proyecto para la georreferenciación.

En el capítulo 4 hablaré acerca de la ingeniería del software del proyecto, y comenzaré con una introducción a la ingeniería del software, y comenzare el análisis del sistema con los requisitos funcionales y no funcionales del sistema, y explicaré los casos de uso que presenta el sistema, junto con los escenarios de los mismos. También se hablará del diseño del sistema con el diagrama de clases y el diseño de la interfaz, comentando las metáforas que utilizo y los distintos storyboards del proyecto.

La memoria concluye con el capítulo 5 que consta de las conclusiones que obtengo al realizar el proyecto.

2. SISTEMAS DE RECOMENDACIÓN

En este capítulo se centrará en los Sistemas de recomendación (SR), comenzando por una breve introducción, comentaremos la clasificación y elementos básicos de los SR.

2.1. Introducción

Existen gran cantidad de dispositivos móviles como, por ejemplo, smartphones, tablets, relojes inteligentes; que proporcionan gran cantidad de información acerca de los usuarios, y puede ser aprovechada para conocer mejor al usuario, sus gustos, ideas y demás.

En la actualidad estos dispositivos toman lugar en nuestro día a día y la cantidad de datos creados en todo el mundo en 2018 alcanzó los 33 zettabytes, 1 zettabyte equivale a 1.000 millones de terabytes. Como podemos observar es muy grande la magnitud del problema que se le presenta al usuario a la hora de elegir un producto, y los SR se encargan de utilizar esta gran cantidad de datos para ayudar y guiar a usuario.

Un sistema de recomendación se puede definir (Michael D. Ekstrand) como un conjunto de herramientas y de software que se utiliza para ofrecer sugerencias y se adapta a los intereses de los usuarios.

Estos sistemas seleccionan directa o indirectamente los datos proporcionados por los usuarios y continúan analizando y procesando información del historial del usuario y convirtiendo estos datos en conocimientos recomendados. Los sistemas de recomendación son actualmente muy eficientes porque pueden correlacionar elementos de nuestro perfil de consumo para hacer recomendaciones.

De manera sencilla una recomendación se presenta como una lista ordenada de ítems. Para que se realice esta tarea el SR está basada en las preferencias, restricciones y conocimientos acerca del usuario, y estos deben quedar reflejados explícitamente para que puedan procesarse computacionalmente y obtener la recomendación. (Ilustración 2.1)

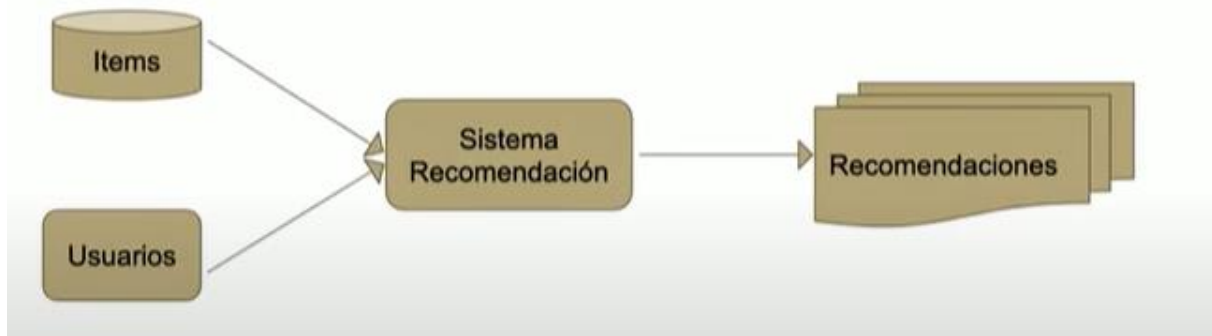


Ilustración 2.1: Esquema simplificado SR.

2.2. Tareas básicas de los sistemas de recomendación

A continuación, comentaré sobre todas las tareas que debe realizar un sistema de recomendación, cabe destacar que los sistemas de recomendación no realizan todas las tareas, solo algunas de ellas dependiendo de la clasificación y/o los objetivos del mismo.

- Encontrar algunos ítems buenos: Recomendar para un usuario, ítems que podrían gustarle al usuario. Como por ejemplo una lista ordenada con las recomendaciones ordenadas por puntuación.
- Encontrar todos los ítems buenos: Es básicamente lo mismo que el punto anterior solo que se recomiendan todos los ítems que le podrían interesar al usuario, ya que hay en algunos casos que el número de ítems es relativamente pequeño o el sistema cubre tareas críticas.
- Anotación en el contexto: Dada una lista de ítems, se destacan algunos de ellos dependiendo de las preferencias del usuario, por ejemplo, un sistema de recomendación de películas que puede resaltar aquellos géneros que merezcan la pena ver al usuario.
- Recomendar una secuencia: No se centra en un solo ítem para la recomendación sino en una secuencia de ítems.
- Recomendar un paquete: Se sugiere un grupo de ítems que se ajustan bien juntos, por ejemplo al recomendar una saga completa de películas.
- Simplemente explorar: El usuario explora el conjunto total de los ítems, pero únicamente para contemplar las opciones que hay.

- Encontrar creíble el recomendador: Debido a la desconfianza de algunos usuarios en los sistemas de recomendación, algunos de estos ofrecen funciones para que vean el comportamiento del sistema.
- Mejorar el perfil: Sirve para que el usuario proporcione información al sistema que sea relevante para que el sistema se ajuste a sus preferencias de la forma más precisa posible.
- Expresarse: Algunos usuarios no están interesados en la recomendación, lo que les importa es que puedan contribuir con valoraciones y opiniones.
- Ayudar a otros: Algunos usuarios buscan simplemente ayudar a la comunidad con sus contribuciones.
- Influnciar a otros: Ha usuarios que buscan influenciar a otros usuarios a la hora de elegir ciertos productos, tanto negativamente para que no se escojan como positivamente.

2.3. Elementos básicos de los sistemas de recomendación

En este punto hablaremos sobre los distintos elementos básicos que presentan los sistemas de recomendación.

2.3.1. Ítems

Se caracterizan por su complejidad y su valor o utilidad. Si un artículo es útil para el usuario, su valor puede ser positivo, si es inadecuado y el usuario toma una decisión equivocada al elegirlo, su valor puede ser negativo. Tenga en cuenta que cuando los usuarios compran productos, siempre incurren en costos, incluido el costo cognitivo de buscar el producto y el costo monetario real que eventualmente pagarán por el producto.

Por ejemplo, el diseñador de una RS de noticias debe considerar la complejidad de la noticia, como su estructura, representación textual y la importancia temporal de las noticias relacionadas. Pero al mismo tiempo, los diseñadores de SR deben comprender que, ya sea que los usuarios paguen por leer noticias o no, siempre existe un costo cognitivo en la búsqueda y lectura de noticias.

2.3.2. Usuarios

Los usuarios de SR pueden tener diferentes objetivos y características. Para la recomendación personalizada y la interacción persona-computadora, SR utiliza diversa información sobre el usuario. Esta información se puede construir de varias formas y, nuevamente, la elección de la información a modelar depende de la técnica de recomendación.

2.3.3. Valoraciones

Las valoraciones son las transacciones que se almacenan en el sistema de recomendación, estas son implícitas cuando el sistema infiere las opiniones de los usuarios y se basa en sus acciones, o explícitas cuando se le solicita al usuario una opinión en una escala de valoración.

2.4. Clasificación de los Sistemas de Recomendación

Existen diferentes tipos de SR y en esta sección consistirá en la clasificación según recomendación personalizados y no personalizados y luego clasificaremos los personalizados en recomendaciones basadas en conocimiento, contenido, con filtrado colaborativo, atentos a contexto o híbridos.

2.4.1. No personalizadas

Las recomendaciones no personalizadas se utilizan para resolver el problema de los SR a la hora de la entrada de un nuevo usuario, no existen datos de valoraciones acerca de su perfil para definir sus gustos y recomendarle ítems en base a eso, y este problema es el llamado 'Arranque en frío' (Cold Start).

El problema tiene una solución, que consistiría en obtener valoraciones implícitas del usuario en redes sociales, para tener alguna base sobre la que trabajar.

Las ventajas de este tipo es que son muy fáciles de implementar ya que recogen los ítems más populares de la comunidad, pero como desventaja es que todos los usuarios reciben la misma recomendación.

2.4.2. Personalizadas

Este tipo de recomendaciones tiene en cuenta gustos y opiniones de los usuarios, que gracias a ello realizan una recomendación específica para cada usuario. A continuación, veremos que estos sistemas de recomendaciones personalizadas se clasifican en los siguientes tipos.

2.4.2.1. Recomendación basada en conocimiento.

Estos sistemas tratan de sugerir objetos de aprendizaje basados en inferencias acerca de las necesidades del usuario y sus preferencias. Se basan en el historial del usuario, anteriores elecciones (Vekariya, 2012) .

Estos sistemas son llamados de sistemas de preferencias implícitas, es decir, que deducen las preferencias a raíz del comportamiento y la historial del usuario, y esto permite que en la mayoría de los casos no sea necesario solicitar mucha información al usuario para que pueda ser recomendado.

Utilizan bases de conocimiento de los usuarios y los ítems para localizar cuál de ellos satisface las necesidades del usuario en concreto. En este tipo de SR se almacenan las necesidades de los usuarios.

Estos sistemas trabajan mejor al comienzo que los de filtrado colaborativo, ya que estos precisan de una gran cantidad de valoraciones de muchos usuarios, ya que la precisión se ve afectada siendo bastante baja.

2.4.2.2. Recomendación basada en contenido.

Las recomendaciones que se le hacen a los usuarios están directamente basadas en información que se puede extraer del contenido, es decir, características del documento que se extraen y se utilizan para sugerir a los usuarios otro contenido que puede estar relacionado. (Aggarwal., 2006) Podemos observar un esquema de la recomendación basada en contenido en la Ilustración 2.2.



Ilustración 2.2: Recomendación basada en contenido.

Las ventajas de este tipo de sistemas son las siguientes:

- Independencia del usuario: Utiliza los ratings únicamente realizados por el usuario, para construir su recomendación, es decir, no necesita valoraciones de otros usuarios.
- Transparencia: Poder conocer las características de los ítems.
- Ítems nuevos: Permite valorar y recomendar ítems introducidos tras desarrollar el sistema y que no tienen valoración alguna.

Las desventajas de este tipo de sistemas son las siguientes:

- Análisis limitado del contenido: Los datos de los ítems que se pueden analizar de manera automática es reducido.

- Sobre especialización: Tendencia para recomendar ítems del mismo tipo que los más valorados del usuario, por tanto, si el usuario solo ha valorado cafeterías, solo recomendará cafeterías.
- Usuarios nuevos: Las recomendaciones para usuarios nuevos no tiene mucha confianza, ya que el sistema necesita recoger una gran cantidad de valoraciones para que pueda definir al usuario.

2.4.2.3. Recomendación con filtrado colaborativo.

Las recomendaciones con filtrado colaborativo (Ilustración 2.3) son las que utilizaremos para el desarrollo del proyecto y es la técnica de recomendación más popular. Su funcionamiento básico es recomendar al usuario ítems que usuarios que tienen gustos similares valoraron positivamente. (Aggarwal., 2006).

Nosotros realizaremos esta recomendación con el algoritmo SVD que explicaremos en el siguiente punto.

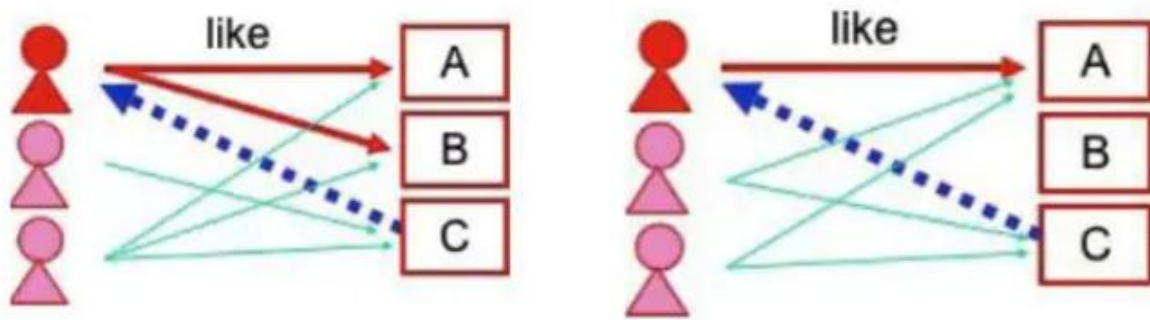


Ilustración 2.3: Recomendación con filtrado colaborativo basada en similitud entre usuarios (izquierda), basada en similitud entre ítems (derecha).

Este tipo de SR presenta problemas de escalabilidad cuando la base de datos aumenta, ya que la búsqueda de los vecinos para un usuario presenta complejidad $O(N)$ o peor, siendo N el conjunto de usuarios.

Para solucionar esto existe otra aproximación basada entre ítems, en lugar que, entre usuarios, que consiste en calcular teniendo en cuenta las valoraciones de los ítems. Esta técnica es muy parecida a la recomendación basada en contenido, pero en vez de calcular la similitud mediante las características de los ítems, se calcula con los patrones de los usuarios.

2.4.2.3.1. Algoritmo SVD

La Descomposición Singular de Valores (Singular Value Decomposition) es una técnica que permite la factorización de matrices y descompone las matrices tal que:

$$\text{SVD}(A) = U \times S \times V^t$$

Donde:

A = Matriz con dimensiones n x m.

U = Matriz ortogonal con dimensiones n x n.

V = Matriz ortogonal de dimensiones m x m.

S = Matriz diagonal de dimensiones m x n.

(Fórmula 2.1: Algoritmo SVD)

La matriz S tendrá en su diagonal valores singulares, en orden decreciente y con valor igual o superior a 0. Se puede ver su estructura en la Ilustración 2.4

$$S = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \dots & 0 & \lambda_n \end{bmatrix}$$

Ilustración 2.4: Matriz S de valores singulares.

2.4.2.3.2. Algoritmo SVD aplicado al filtrado colaborativo

En el filtrado colaborativo es necesario el manejo de la matriz de valoraciones de los usuarios y los ítems, con lo que precisaremos de esta matriz como base para aplicar el algoritmo SVD.

$$A = U \times S \times V^t$$

Donde:

A = Matriz con dimensiones n (usuarios) $\times$ m (ítems).

U = Matriz ortogonal con dimensiones $n(\text{usuarios}) \times m$.

V = Matriz ortogonal de dimensiones $m \times m$.

S = Matriz diagonal de dimensiones $m \times n$ (ítems).

(Fórmula 2.2: SVD aplicado a filtrado colaborativo.)

2.4.2.4. Recomendación híbrida.

Como hemos observado, todos los tipos de SR presentan sus ventajas e inconvenientes, por ello, los SR híbridos, combinan 2 o más técnicas de recomendación para obtener un mejor rendimiento con menos inconvenientes de cualquiera de ellas por separado. En la Tabla 2.1 se muestran algunos de los métodos de combinación que se han empleado.

Método de hibridación	Descripción
Ponderado	Las puntuaciones (o votos) de varias técnicas de recomendación son combinados para producir una única recomendación.
Conmutación	El sistema cambia entre las técnicas de recomendación dependiendo de la situación actual.
Mixto	Las recomendaciones de varios recomendadores son presentado al mismo tiempo
Combinación de características	Se lanzan características de diferentes fuentes de datos de recomendación en un único algoritmo de recomendación.
Cascada	Un recomendador perfecciona las recomendaciones dadas por otro.
Aumento de las características	El resultado de una técnica se utiliza como característica de entrada para otra.
Meta-nivel	El modelo aprendido por un recomendador se utiliza como entrada para otro.

Tabla 2.1: Tabla de Métodos de hibridación.

2.4.2.5. Recomendaciones sensibles al contexto.

Hay muchas veces que la recomendación depende de si es de noche, de día, llueve, o estoy en otro país, y con esto nace la necesidad de agregarle contexto a los SR para poder realizar las recomendaciones. Se precisan de 3 modelos para incorporar contexto:

- Pre-filtrado contextual: Se utiliza la información contextual de la actualidad para escoger los datos relevantes para utilizar un SR de los vistos anteriormente. Esto hace que la matriz esté más dispersa y afectará a la precisión de la recomendación.
- Post-filtrado contextual: La información contextual se obvia al principio y las valoraciones se predicen como en los anteriores SR con todos los datos y finalmente se ajustan a los datos contextuales.
- Modelado contextual: La información del contexto se utiliza en el modelo de estimación de las valoraciones.

Podemos visualizar estos 3 modelos en la ilustración 2.5.

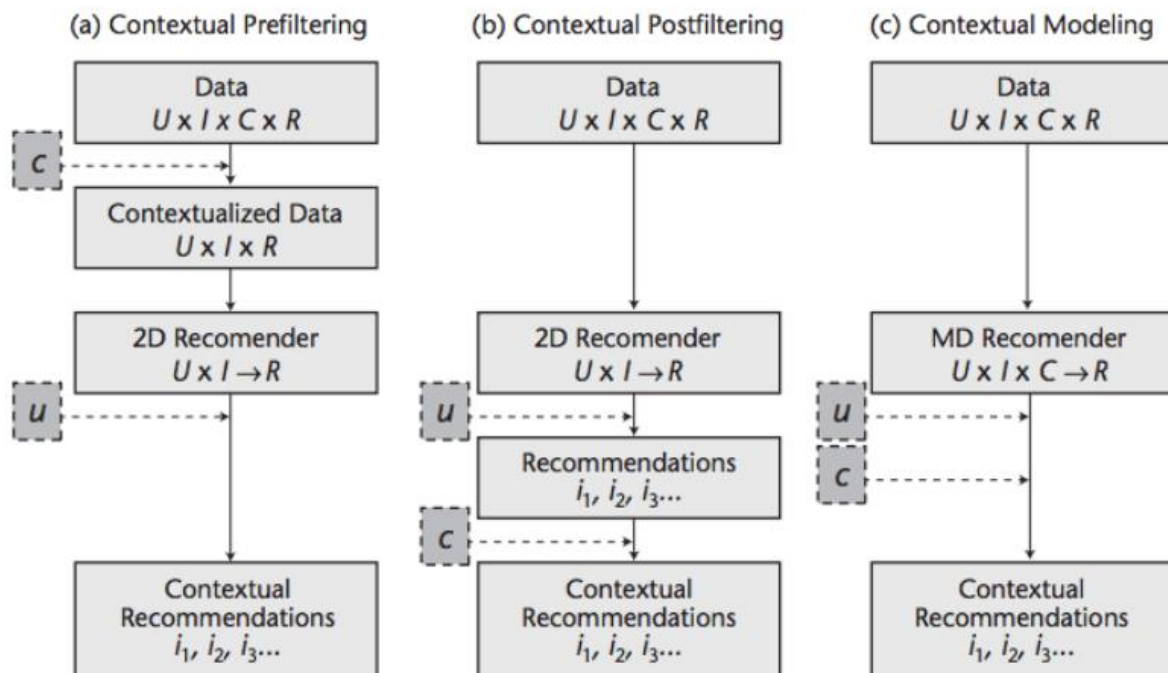


Ilustración 2.5: Paradigmas para la incorporación de contexto en sistemas de recomendación.

Fuente: [https://miro.medium.com/max/874/1*uMMkoeZ39i7oYiM0UtfN2w.png]

3. TECNOLOGÍAS UTILIZADAS

3.1. Introducción

Python es un lenguaje de programación interpretado, orientado a objetos de alto nivel y con semántica dinámica; su sintaxis destaca en la legibilidad del código, lo que facilita operaciones como la depuración y favorece la productividad. Ofrece la potencia y la flexibilidad de los lenguajes compilados con una curva de aprendizaje suave.

3.2. ¿Por qué Python?

Aunque Python fue creado como lenguaje de uso general, cuenta con librerías y entornos de desarrollo para cada una de las fases del proceso de Data Science⁴. Además, tiene el carácter de open source y la facilidad de aprendizaje que le ha llevado a tomar la delantera a otros lenguajes como por ejemplo R.

Como características también cabe destacar que no requiere de compilación ya que es un código interpretado.

En la Ilustración 3.1 observamos que es el lenguaje más usado en el mundo frente a Java y JavaScript.



Ilustración 3.1: Gráfica de los lenguajes de programación más usados

⁴ Data Science es la ciencia centrada en el estudio de los datos.

3.3. Herramienta de desarrollo: Spyder IDE

Spyder es un entorno de desarrollo integrado (IDE) multiplataforma de código abierto para programación científica en el lenguaje Python. Podemos observar el logo en la Ilustración 3.2.



Ilustración 3.2: Logo de Spyder IDE

Fuente = [[https://en.wikipedia.org/wiki/Spyder_\(software\)#/media/File:Spyder_logo.svg](https://en.wikipedia.org/wiki/Spyder_(software)#/media/File:Spyder_logo.svg)]

Es un lenguaje extensible con complementos propios y de terceros, incluye soporte para herramientas interactivas e integra instrumentos de introspección y aseguramiento de calidad del código.

3.3.1. Características

Algunas de las características fundamentales de Spyder son las siguientes:

- Plataforma cruzada, se encuentra disponible para Linux, Windows y macOS.
- Código gratuito y de código abierto.
- Resaltado de sintaxis.
- Soporte en varios idiomas.
- Consola interactiva.
- Explorador de variables.
- Visor de documentación, visualización de gráficos y datos
- Soporte para extensiones de funciones a través de complementos y API.

3.3.2. Visualización de la interfaz

Visualmente el entorno se encuentra dividido (Ilustración 3.3).

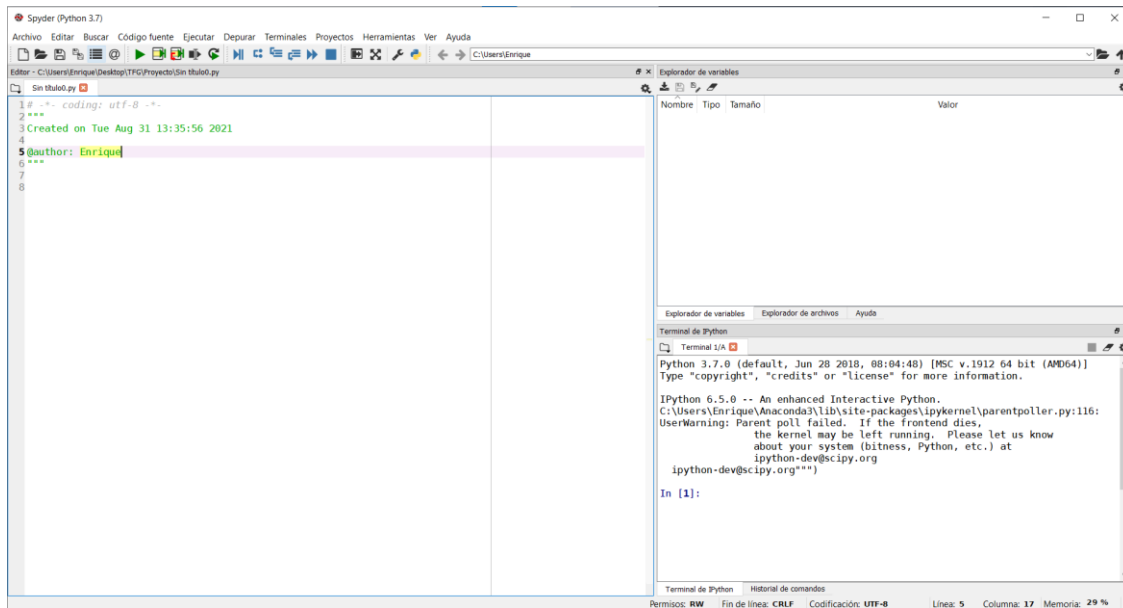


Ilustración 3.3: Interfaz de Spyder.

En la parte izquierda se localiza el editor donde se desarrolla el código, y en la derecha se localizan en la parte superior el explorador de Variables (Ilustración 3.4), Explorador de archivos (Ilustración 3.5) y la Ayuda (Ilustración 3.6), mientras que en la parte inferior se localiza el terminal de Ipython (Ilustración 3.7) y el Historial de comandos (Ilustración 3.8).

Nombre	Tipo	Tamaño	Valor
dataset	dict	5	{'tiempo': ['nublado', 'nublado', 'nublado', 'nublado', 'lluvioso', ...]}
df	DataFrame	(14, 5)	Column names: tiempo, temp, humedad, viento, rol
eps	float64	1	2.220446049250313e-16
humedad	list	14	['alta', 'baja', 'normal', 'alta', 'alta', 'baja', 'baja', 'baja', 'no ...]
rol	list	14	['yes', 'yes', 'yes', 'yes', 'yes', 'yes', 'no', 'yes', 'no', 'no', ...]
t	dict	1	{'tiempo': {'lluvioso': {...}, 'nublado': 'yes', 'soleado': {...}}}
temp	list	14	['alta', 'baja', 'normal', 'alta', 'normal', 'baja', 'baja', 'normal', ...]
tiempo	list	14	['nublado', 'nublado', 'nublado', 'nublado', 'lluvioso', 'lluvioso', ...]
viento	list	14	['FALSE', 'TRUE', 'TRUE', 'FALSE', 'FALSE', 'FALSE', 'TRUE', 'FALSE', ...]

Ilustración 3.4: Explorador de variables de Spyder

Nombre	Tamaño	Tipo	Última modificación
 dataset.dat	476 bytes	Archivo dat	26/05/2021 16:31
 FuzzyID3ens00005.pdf	207 KiB	Archivo pdf	27/05/2021 14:41
 Fuzzyid3ens00005.py	3 KiB	Archivo py	27/05/2021 14:28
 FuzzyID3ens00005.rar	190 KiB	Archivo rar	27/05/2021 14:41

Explorador de variables

Explorador de archivos

Ayuda

Ilustración 3.5: Explorador de archivos de Spyder.

Uso

En este panel es posible obtener la ayuda de cualquier objeto al oprimir **Ctrl+I** estando al frente del mismo, bien sea en el Editor o en la Terminal.

Esta ayuda también se puede mostrar automáticamente después de escribir un paréntesis junto a un objeto. Este comportamiento puede activarse en *Preferencias > Ayuda*.

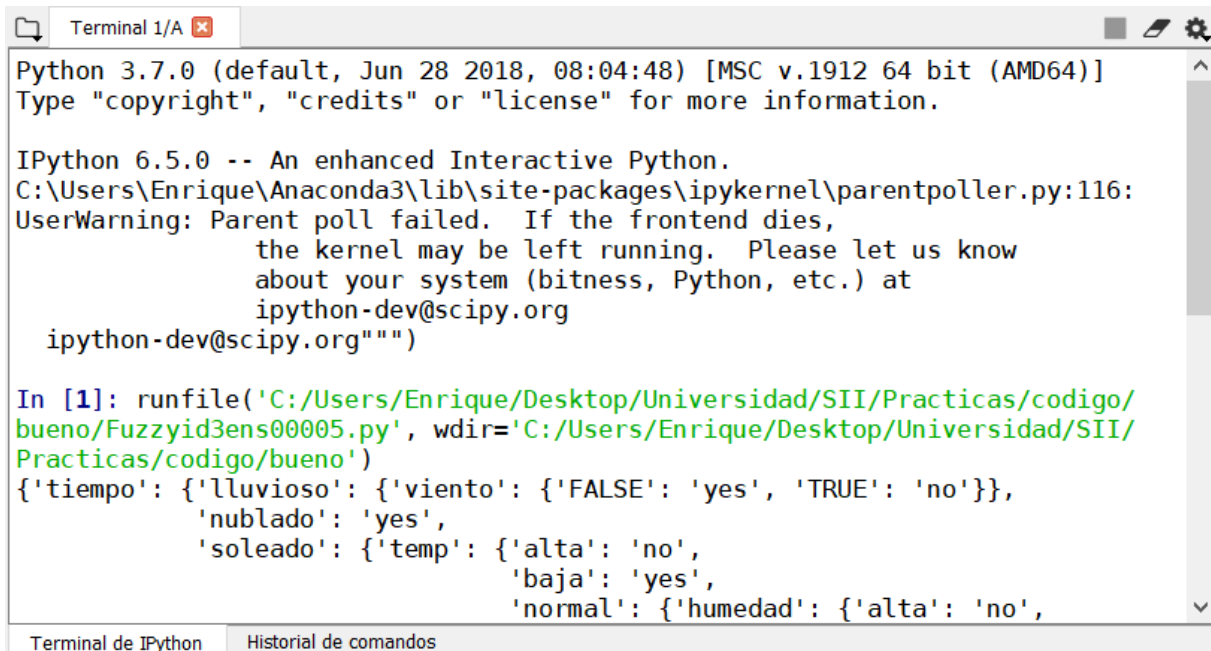
Nuevo en Spyder? Lee nuestro [tutorial](#)

Explorador de variables

Explorador de archivos

Ayuda

Ilustración 3.6 Ayuda de Spyder.



```

Python 3.7.0 (default, Jun 28 2018, 08:04:48) [MSC v.1912 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 6.5.0 -- An enhanced Interactive Python.
C:\Users\Enrique\Anaconda3\lib\site-packages\ipykernel\parentpoller.py:116:
UserWarning: Parent poll failed. If the frontend dies,
               the kernel may be left running. Please let us know
               about your system (bitness, Python, etc.) at
               ipython-dev@scipy.org
ipython-dev@scipy.org"""

In [1]: runfile('C:/Users/Enrique/Desktop/Universidad/SII/Practicas/codigo/
bueno/Fuzzyid3ens00005.py', wdir='C:/Users/Enrique/Desktop/Universidad/SII/
Practicas/codigo/bueno')
{'tiempo': {'lluvioso': {'viento': {'FALSE': 'yes', 'TRUE': 'no'}}},
 'nublado': 'yes',
 'soleado': {'temp': {'alta': 'no',
                      'baja': 'yes',
                      'normal': {'humedad': {'alta': 'no',

```

Ilustración 3.7: Terminal de IPython en Spyder.



```

## --- (Tue Aug 24 16:39:55 2021) ---
runfile('C:/Users/Enrique/Desktop/TFG/Proyecto/programa.py', wdir='C:/Users/
Enrique/Desktop/TFG/Proyecto')
runfile('C:/Users/Enrique/Desktop/TFG/Proyecto/programa.py', wdir='C:/Users/
Enrique/Desktop/TFG/Proyecto')
runfile('C:/Users/Enrique/Desktop/TFG/Proyecto/programa.py', wdir='C:/Users/
Enrique/Desktop/TFG/Proyecto')
runfile('C:/Users/Enrique/Desktop/TFG/Proyecto/programa.py', wdir='C:/Users/
Enrique/Desktop/TFG/Proyecto')

## --- (Wed Aug 25 11:01:47 2021) ---
runfile('C:/Users/Enrique/Desktop/TFG/Proyecto/programa.py', wdir='C:/Users/
Enrique/Desktop/TFG/Proyecto')

## --- (Tue Aug 31 13:35:51 2021) ---
runfile('C:/Users/Enrique/Desktop/Universidad/SII/Practicas/codigo/bueno/
Fuzzyid3ens00005.py', wdir='C:/Users/Enrique/Desktop/Universidad/SII/
Practicas/codigo/bueno')

```

Ilustración 3.8: Historial de comandos en Spyder.

3.4. Sistemas de Información

Un Sistema de Información (SI) consiste en la unión de información en formato digital y herramientas informáticas (programas), que permiten su análisis con unos objetivos concretos.

Cuando se habla de SI, se piensa en sistemas informáticos que prestan servicio a empresas u organismos; y este servicio implica:

- Almacenar la información relativa al capital de la empresa y a las transacciones.
- Permitir la consulta de datos en particular con facilidad y desde distintos puntos.
- Analizar estos datos para obtener un conocimiento de las transformaciones que atraviesa la empresa.
- Ayudar en cuanto a toma de decisiones importantes.

3.5. ¿Qué son los SIG?

Un sistema de información geográfica (SIG) es un marco para recopilar, administrar y analizar datos. El SIG tiene sus raíces en la ciencia geográfica e integra varios tipos de datos. Utiliza mapas y escenas 3D para analizar ubicaciones espaciales y organizar capas de información para su visualización. Con esta característica única, GIS revela conocimientos más profundos ocultos en los datos, como patrones, relaciones y situaciones, para ayudar a los usuarios a tomar decisiones más informadas.

Básicamente los SIG son herramientas para trabajar con información georreferenciada, que es aquella que viene acompañada de una posición (coordenadas en este caso), geográficas.

Nosotros utilizamos el formato .kml basado en .xml para georreferenciar la información de nuestro proyecto, lo veremos a continuación.

3.5.1. Estructura de los SIG

En sentido amplio, los SIG están constituidos por:

- Bases de datos espaciales: Se codifica mediante unos modelos de datos específicos.
- Bases de datos temáticas: La vinculación con la base de datos cartográfica le permite asignar a cada elemento unos valores temáticos (nombre, identificador).
- Conjunto de programas: Permiten manejar estas bases de datos de manera útil para lograr multitud de propósitos en investigación, docencia o gestión.
- Conjunto de ordenadores y periféricos: Componen el soporte físico de los SIG e incluyen tanto el programa de gestión de SIG como otros programas de apoyo.
- Comunidad de usuarios: Demandantes de información espacial.
- Administradores del sistema: Se encargan de resolver requerimientos de los usuarios utilizando las herramientas disponibles o proporcionando herramientas nuevas.

3.5.2. Formato KML

El formato KML (Keyhole Markup Language) (Emmanuel Stefanakis and Kostas Patroutas, 2008, págs. 143-152) , como hemos comentado anteriormente, se basa en XML y almacena datos geográficos y su contenido relacionado. Está basado en los estándares del Consorcio Geoespacial abierto.

Como observamos en la Ilustración 3.15 esta es la estructura del fichero POIJaen.kml

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>POI Jaén</name>
    <description/>
    <Style id="icon-1899-0288D1-nodesc-normal">
      <IconStyle>
        <color>ffdi8802</color>
        <scale>1</scale>
        <Icon>
          <href>https://www.gstatic.com/mapspro/images/stock/503-whit-blank_maps.png</href>
        </Icon>
        <hotSpot x="32" xunits="pixels" y="64" yunits="insetPixels"/>
      </IconStyle>
      <LabelStyle>
        <scale>0</scale>
      </LabelStyle>
      <BalloonStyle>
        <text><![CDATA[<h3>{name}</h3>]]></text>
      </BalloonStyle>
    </Style>
    <Style id="icon-1899-0288D1-nodesc-highlight">
      <IconStyle>
        <color>ffdi8802</color>
        <scale>1</scale>
        <Icon>
          <href>https://www.gstatic.com/mapspro/images/stock/503-whit-blank_maps.png</href>
        </Icon>
        <hotSpot x="32" xunits="pixels" y="64" yunits="insetPixels"/>
      </IconStyle>
      <LabelStyle>
        <scale>1</scale>
      </LabelStyle>
      <BalloonStyle>
        <text><![CDATA[<h3>{name}</h3>]]></text>
      </BalloonStyle>
    </Style>
    <StyleMap id="icon-1899-0288D1-nodesc">
      <Pair>
        <key>normal</key>
        <styleUrl>#icon-1899-0288D1-nodesc-normal</styleUrl>
      </Pair>
      <Pair>
        <key>highlight</key>
        <styleUrl>#icon-1899-0288D1-nodesc-highlight</styleUrl>
      </Pair>
    </StyleMap>
    <Folder>
      <name>Restaurantes</name>
      <Placemark>
        <name>Restaurante Kasler</name>
        <styleUrl>#icon-1899-0288D1-nodesc</styleUrl>
        <Point>
          <coordinates>
```

Ilustración 3.9: Fichero POIJaen.kml

KML soporta una serie de primitivas geométricas como puntos, líneas, polígonos entre otros. Estas primitivas se pueden asignar a los elementos del lenguaje que contienen información como pueden ser los Placemark⁵.

Para la obtención de los distintos POI⁶ situados en Jaén (Ilustración 4.1) he realizado desde la plataforma de MyMaps de Google, la selección de los mismos y luego los he exportado a un fichero .kml para parsearlos mediante Python y obtener los datos de los distintos lugares de interés georreferenciados por las coordenadas.



Ilustración 3.10: POI marcados en la provincia de Jaén desde Google MyMaps.

⁵ PlaceMark: Marca de Posición.

⁶ POI: Points of interest (Puntos de interés).

4. INGENIERÍA DEL SOFTWARE

4.1. Introducción a la Ingeniería del Software

Hay muchas definiciones para ingeniería del software pero aquí menciono algunas:

“La ingeniería del software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación hasta el mantenimiento de éste después de que se utiliza.” (Sommerville, 2005)

“La Ingeniería del Software es la disciplina que estudia el desarrollo, la operación y el mantenimiento del software.” (Rinaudo)

La ingeniería del software es una especialidad que consiste en sistemas, instrumentos y técnicas que se emplean en el desarrollo de programas informáticos. Va unido al ‘ciclo de vida del software’ que consiste en cuatro etapas: (Sommerville, 2005)

- Concepción: Encargada de determinar la repercusión del proyecto y diseña el modelo del negocio.
- Elaboración: Contiene la planificación del proyecto, especificando sus características.
- Construcción: Elaboración del producto.
- Transición: Entrega del producto al cliente.

Tras este ciclo, comienza la fase del mantenimiento del software, que consiste en que el software ofrece soluciones por reportes de error de los usuarios, y se incorporan a actualizaciones.

La ingeniería de software manifiesta la actividad del programa, que es la labor más importante para el momento de la creación de un software. El ingeniero de software se encarga de gestionar el proyecto para que evolucione en un periodo de tiempo determinado y con los recursos previstos para el proyecto.

La ingeniería de Software (Sommerville, 2005) consta de 7 etapas que se desarrollan a continuación:

- **Análisis:** Se investiga acerca de un problema al que se le encuentra una solución; éste se debe definir con claridad identificando los elementos principales que conformarán el producto.
- **Diseño:** Utiliza la información acumulada en la etapa anterior. Su función principal es crear un modelo o las características precisas para el producto.
- **Desarrollo:** Consiste en emplear los diseños creados en el diseño para desarrollar los elementos a usar en el sistema.
- **Pruebas:** En esta etapa se garantiza que los elementos individuales que componen el sistema o el producto, presentan las características de la etapa de diseño.
- **Implementación:** En esta etapa se distribuye el producto al cliente.
- **Mantenimiento:** Se aplican soluciones frente a problemas que pueda presentar el producto, ofreciendo una nueva versión con los problemas solventados.
- **Final EOL (End-of-Life):** Se ejecutarán las labores que garanticen a los clientes y a los empleados de que el producto no estará más a disposición.

4.2. Análisis del Sistema

Para el análisis del sistema utilizaremos los requisitos funcionales y los no funcionales, los casos de uso con sus distintos escenarios.

4.2.1. Requisitos Funcionales

Los requisitos funcionales de un sistema describen lo que el sistema debe hacer, y dependen del tipo de software que se desarrolle, de los posibles usuarios del software y del enfoque general que decida darle la organización. (Sommerville, 2005)

Los requisitos funcionales de este proyecto se observarán mediante la Tabla 4.1.

ID	NOMBRE	DESCRIPCION
RF1	Creación de los datos	El administrador podrá crear los conjuntos de los datos que se utilizarán
RF2	Estudio de los datos	El administrador podrá obtener un fichero de datos con los resultados del análisis de los datos.
RF3	Obtención de error cuadrático medio (RMSE) de los datos	El administrador podrá obtener un fichero con el valor del RMSE de los datos.
RF4	Generar recomendación	El usuario podrá escribir el id de un usuario y la distancia máxima de búsqueda para obtener la recomendación.

Tabla 4.1: Requisitos funcionales.

4.2.2. Requisitos No Funcionales

Los requisitos no funcionales, son aquellos que no se refieren a las funciones específicas que proporciona el sistema directamente, sino al conjunto de las propiedades emergentes como pueden ser la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. (Sommerville, 2005)

Los requisitos no funcionales se relacionan con las restricciones que hay que tener en cuenta en relación con los requerimientos del sistema (Tabla 4.2).

ID	NOMBRE	DESCRIPCION
RNF1	Interfaz sencilla	La aplicación debe tener una interfaz fácil de utilizar.
RNF2	Integridad	La aplicación debe mantener la integridad de los datos y no poder ser modificados por nadie que no sea el administrador.
RNF3	Rendimiento	La aplicación debe ser rápida en el motor de recomendación.
RNF4	Tecnología a utilizar	Para el desarrollo de la aplicación utilizaré Python.
RNF5	Volumen de los datos	La aplicación y el motor de recomendación trabajara con un conjunto de datos de hasta 40000 ítems

Tabla 4.2: Requisitos no funcionales

4.2.3. Casos de uso

Los casos de uso son una técnica que se basa en escenarios para la obtención de requisitos que se introducen por primera vez. Se han convertido en una característica fundamental de la notación UML, que se usa para describir los modelos de sistemas orientados a objetos. (Sommerville, 2005)

Se forma de:

- Nombre del caso de uso.
- Actores Participantes.
- Condiciones de entrada.
- Flujo de eventos.
- Condiciones.

4.2.3.1. Actores

Un actor modela una entidad externa que interacciona con el sistema, esto quiere decir que es un tipo de usuario del sistema, al igual que ocurre con un caso de uso, debe tener un nombre único y puede ir acompañado de una descripción del mismo. Los actores en este sistema serán:

- Usuario: Será el usuario que entra en el sistema y que solicita la recomendación del mismo.
- Administrador: Se encarga de editar y proporcionar los datos del sistema y de los análisis del sistema y de errores.

4.2.3.2. Diagrama frontera

En este diagrama se describe completamente toda la funcionalidad de la aplicación desarrollada, para explicar el sistema completo (Ilustración 4.1).

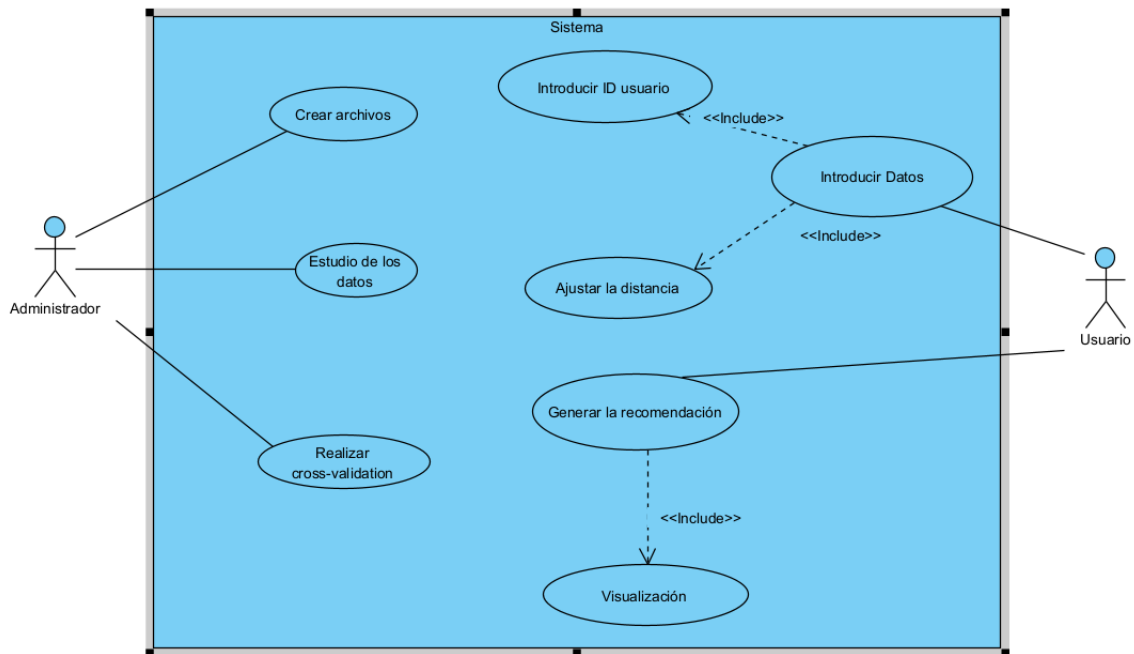


Ilustración 4.1: Diagrama frontera del sistema.

4.2.3.3. Caso de Uso 1: Crear archivos

Actores participantes: Administrador.

Condición de entrada:

- El administrador debe de tener el fichero con extensión .kml y el fichero 'usuarios.csv' en la carpeta de ejecución.
- El administrador debe quitar el comentario de la línea 280.

Flujos de eventos:

- 1.- El administrador ejecuta la aplicación.
- 2.- El sistema lee el fichero .kml.
- 3.- Se obtienen los datos de los lugares del fichero .kml y se crean las valoraciones de esos lugares para los usuarios del fichero usuarios.csv.

Excepciones:

E1: Se informa que no se dispone de alguno o todos los ficheros correspondientes.

Condición de salida: Se obtiene los ficheros datosjaen.csv y ratings.csv.

4.2.3.4. Caso de Uso 2: Estudio de los datos

Actores participantes: Administrador.

Condición de entrada:

- El administrador debe de tener los ficheros 'usuarios.csv' y 'ratings.csv' en la carpeta de ejecución.
- El administrador debe quitar el comentario de la línea 281.

Flujos de eventos:

- 1.- El administrador ejecuta la aplicación.
- 2.- El sistema lee los ficheros mencionados anteriormente.
- 3.- Se obtienen los análisis de datos de los lugares, usuarios y las valoraciones.

Excepciones:

E1: Se informa que no se dispone de alguno o todos los ficheros correspondientes.

Condición de salida: Se obtienen unas tablas de análisis de los ficheros.

4.2.3.5. Caso de Uso 3: Realizar cross-validation

Actores participantes: Administrador.

Condición de entrada:

- El administrador debe de tener el fichero 'ratings.csv' en la carpeta de ejecución.
- El administrador debe quitar el comentario de la línea 282.

Flujos de eventos:

- 1.- El administrador ejecuta la aplicación.
- 2.- El sistema lee el fichero 'ratings.csv'.
- 3.- Se ejecuta la valoración de cruce con el algoritmo SVD.

Excepciones:

E1: Se informa que no se dispone de alguno o todos los ficheros correspondientes.

Condición de salida: Se obtienen los errores cuadráticos medios de los subconjuntos de test y train por consola.

4.2.3.6. Caso de Uso 4: Introducir datos

Actores participantes: Usuario.

Condición de entrada:

Debe ejecutarse la interfaz del sistema.

Flujos de eventos:

- 1.- El usuario ejecuta la aplicación.
- 2.- El usuario introduce los valores de identificador de usuario y de distancia máxima.
- 3.- Se pulsa el botón de generar recomendación.

Excepciones:

E1: Se informa que el valor de identificador de usuario no está entre los rangos y debe de introducir los datos de nuevo.

Condición de salida: Se genera la recomendación (Caso de uso 5)

4.2.3.7. Caso de Uso 5: Generar Recomendación.

Actores participantes: Usuario.

Condición de entrada:

El usuario debe de introducir valores válidos para el identificador de usuario.

Flujos de eventos:

- 1.- Se pulsa el botón de generar recomendación.
- 2.- Se ejecuta el algoritmo SVD para encontrar la recomendación para el usuario designado para la distancia que haya indicado.

Condición de salida: Se abre una ventana en el navegador con los distintos puntos en el mapa georreferenciados.

4.2.3.8. Escenarios

Un escenario es una instancia de un caso de uso en el que se fijan las condiciones relativas a los distintos eventos, por tanto, a la hora del desarrollo no hay alternativas. (Heyde, 2016)

Estos escenarios están formados por los siguientes elementos:

- Nombre.
- Descripción.
- Actores.
- Flujo de eventos.

Estas pequeñas historias inventadas son muy útiles para los desarrolladores, ya que, al describir una interacción con la interfaz, permiten anticiparse a los problemas. Estas historias deben desarrollarse con alto nivel de detalles, para que los diseñadores encuentren más fácil debatir que discutir sobre una situación abstracta.

A continuación, propongo los escenarios que presenta mi proyecto y los desarrollaré más adelante:

- Escenario 1: Crear ficheros de datos.
- Escenario 2: Estudiar los datos creados.
- Escenario 3: Realizar la validación de cruce de los datos.
- Escenario 4: Generar la recomendación.

Escenario 1: Crear ficheros de datos

Nombre: Enrique (administrador) crea los ficheros de datos.

Descripción: Enrique crea los ficheros de los datos que necesitará para acceder a las funcionalidades del programa y los visualiza.

Actores: Enrique.

Flujo de eventos:

1. Enrique examina que existe el fichero *POIJaen.kml* y *usuarios.csv*.
2. Enrique inicia la aplicación.
3. La aplicación le pregunta si quiere acceder como usuario o administrador.
4. Enrique pulsa el botón de Administrador, ya que necesita crear los datos para su utilización.
5. El sistema le despliega las 3 opciones de administrador que tiene.
6. Enrique escoge la opción de **Crear ficheros de datos**.
7. El sistema crea los ficheros *datosjaen.csv* y *ratings.csv*
8. Enrique accede a la carpeta del programa abre los ficheros y los visualiza.

Escenario 2: Estudiar los datos creados

Nombre: María José (administradora) estudia los ficheros de los datos.

Descripción: María José estudia la distribución de los datos obtenidos en los ficheros *datosjaen.csv*, *ratings.csv* y *usuarios.csv*.

Actores: María José.

Flujo de eventos:

1. María José ejecuta la aplicación.
2. La aplicación le pregunta si quiere acceder como usuario o administrador.
3. María José pulsa el botón de Administrador, ya que necesita saber si los datos que tiene están balanceados.
4. El sistema le despliega las 3 opciones de administrador que tiene.
5. María José escoge la opción de **Estudiar los datos**.
6. El sistema crea el fichero *EstudioDatos.csv*.
7. María José accede a la carpeta del programa y visualiza el fichero *EstudioDatos.csv*.
8. María José visualiza que la distribución de los datos está balanceada.

Escenario 3: Realizar la validación de cruce de los datos

Nombre: Antonio (administrador) realiza la validación de cruce de los datos.

Descripción: Antonio realiza la validación de cruce de los datos que tiene con el algoritmo SVD para obtener el Error Cuadrático Medio (RMSE) y visualizarlo.

Actores: Antonio.

Flujo de eventos:

1. Antonio inicia la aplicación.
2. La aplicación le pregunta si quiere acceder como usuario o administrador.
3. Antonio pulsa el botón de Administrador, ya que necesita saber el RMSE y saber que los datos son correctos para el funcionamiento.
4. El sistema le despliega las 3 opciones de administrador que tiene.
5. Antonio escoge la opción de **Obtener error (RMSE) mediante validación de cruce**.
6. El sistema crea el fichero *Errores_cross_validation.csv*
7. Antonio accede a la carpeta del programa abre el fichero.
8. Antonio visualiza el RMSE.

Escenario 4: Generar la recomendación

Nombre: Isabel (usuario) genera la recomendación.

Descripción: Isabel genera una recomendación para el usuario con el identificador '123' en un rango de distancia de '20 km' y visualiza en una pestaña del navegador las cinco recomendaciones con mayor puntuación.

Actores: Isabel.

Flujo de eventos:

1. Isabel inicia la aplicación.
2. La aplicación le pregunta si quiere acceder como usuario o administrador.
3. Isabel pulsa el botón de Usuario.
4. El sistema le despliega los campos a rellenar (identificador de usuario y distancia máxima de búsqueda.)
5. Isabel rellena el campo de identificador de usuario con "123", y el campo de distancia máxima de búsqueda "20".
6. Isabel pulsa el botón de generar recomendación.
7. El sistema crea el fichero "*Mirecomendacion.html*" con las cinco recomendaciones con mayor puntuación para el usuario y la distancia máxima de búsqueda y lo abre en una pestaña del navegador.
8. Isabel visualiza la pestaña con los 5 puntos de interés marcados en el mapa de la provincia Jaén.

4.3. Diseño del Sistema

4.3.1. Diagrama de Clases

El diagrama de clases contiene los atributos, métodos y asociaciones de los objetos. Este diagrama es esencial para modelar el sistema mediante objetos, y de todos los diagramas UML⁷, éste es el único obligatorio para este modelado. (Heyde, 2016)

En la Ilustración 4.2 podemos observar el diagrama de clases que presenta nuestro sistema.

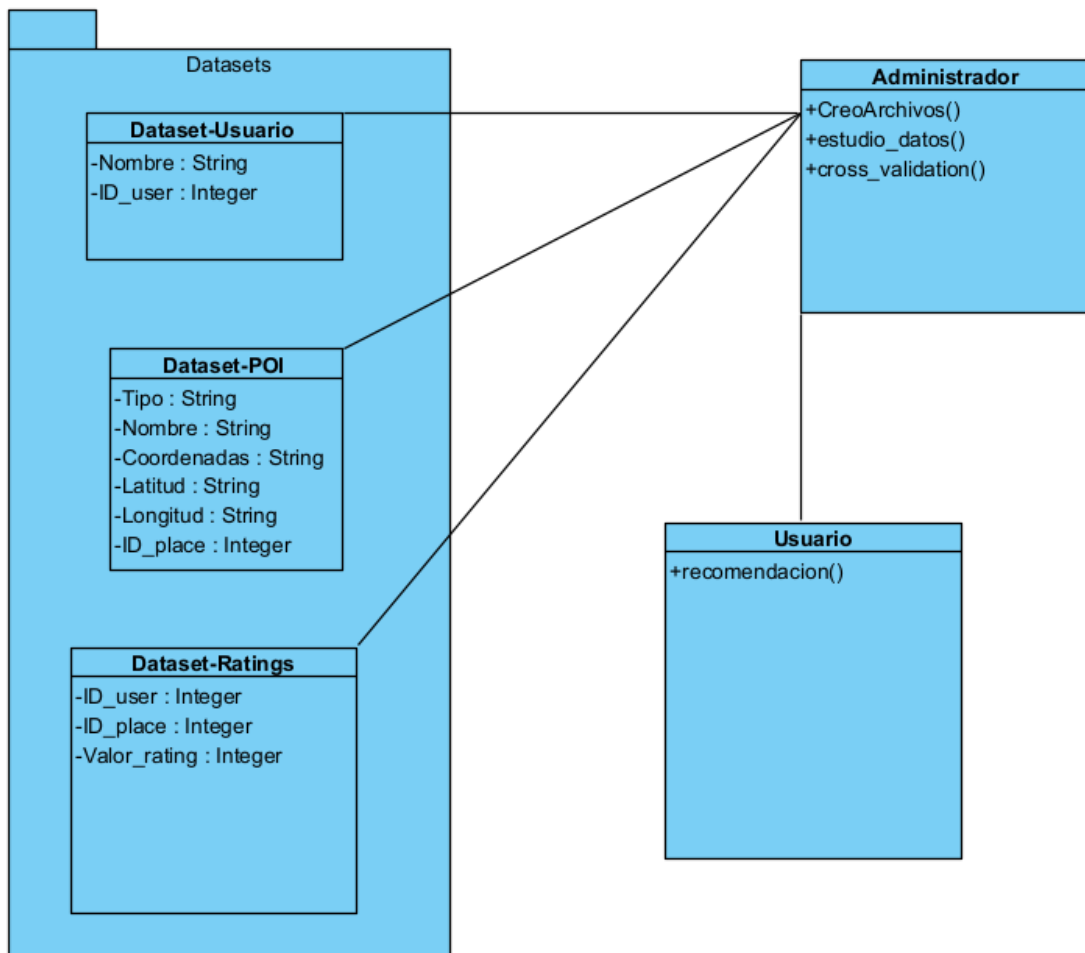


Ilustración 4.2: Diagrama de clases.

⁷ UML: lenguaje unificado de modelado (UML, por sus siglas en inglés, Unified Modeling Language.)

4.4. Diseño de la interfaz

La interfaz gráfica se dispondrá en una serie de ventana desplegadas desde la librería de tkinter⁸ de Python. En la Ilustración 4.3 se puede observar un ejemplos de cómo se ve la interfaz.

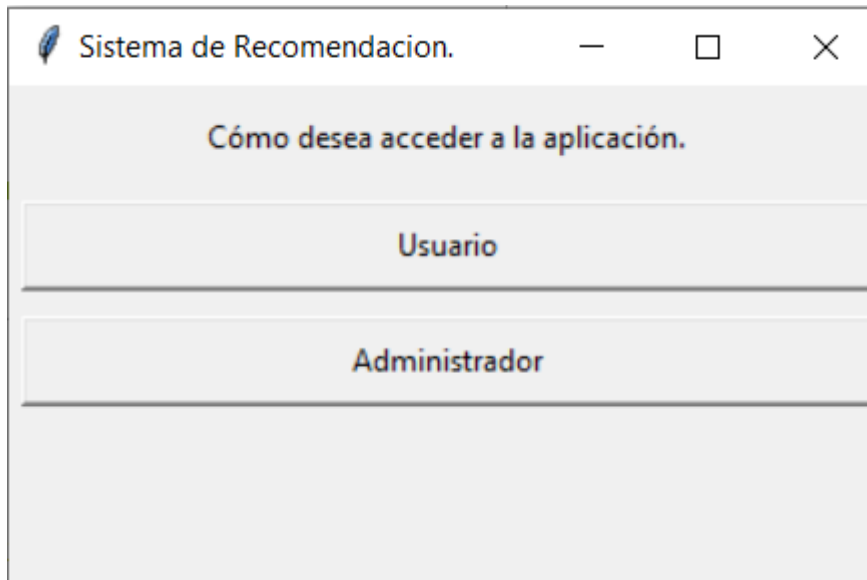


Ilustración 4.3: Diseño de la interfaz.

⁸ Tkinter: Tkinter es un binding de la biblioteca gráfica Tcl/Tk para el lenguaje de programación Python.

La salida del programa serán un fichero .html que se crea automáticamente donde tengamos el programa guardado y se abre de forma automática de la forma que observamos en la Ilustración 4.4.

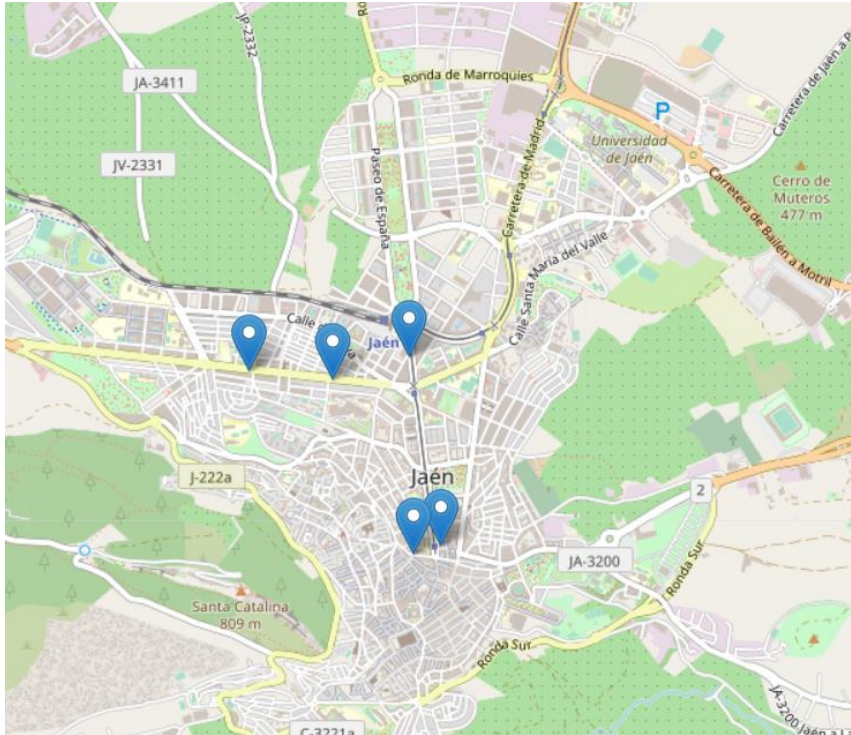


Ilustración 4.4: Salida de la ejecución del programa.

4.4.1. Metáforas

Las metáforas se utilizan para referirnos a conceptos abstractos en la utilización del lenguaje, en nuestro campo, la informática, se usan las metáforas para referirnos a tareas que la interfaz permite desarrollar y así facilitar el uso de las aplicaciones.

En nuestro caso utilizaré solo una metáfora que es la de los POI y utilizaremos el icono que observamos en la Ilustración 4.5.



Ilustración 4.5: Metáfora de los POI

4.4.2. Storyboards

La totalidad del diseño de la interfaz representa su visión estática, pero la interacción del usuario con la interfaz concurre de manera más fluida y presenta unos caminos para la navegación.

Para estudiar la navegación, se utilizan los Storyboards, que son herramientas que narra secuencialmente las pantallas que el usuario visualiza hasta llegar a la acción concreta. Este prototipo se utiliza para que el usuario pruebe y valide la interfaz para realizar correcciones.

Para la realización de los Storyboards se pondrán capturas de pantalla secuenciales y en el orden de aparición al usuario.

Presentará las siguientes tanto para el usuario como para el administrador:

- Storyboard creación de los datos.
- Storyboard análisis de los datos.
- Storyboard cross-validation.
- Storyboard recomendación.

4.4.2.1. Storyboard Creación de los datos

Para crear los datos (Ilustración):

- El usuario pulsa el botón Administrador.
- El sistema muestra las opciones que tiene como Administrador.
- El usuario pulsa el botón Crear ficheros de datos.
- El sistema crea los datos y emite un mensaje por pantalla para saber que ha terminado.

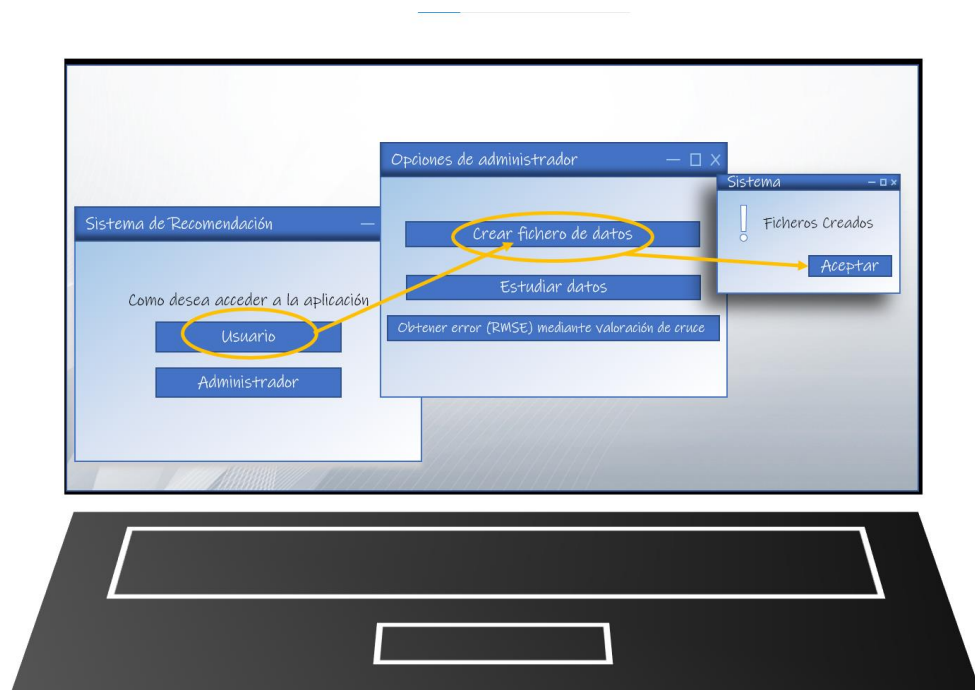


Ilustración 4.6: Storyboard Creación de datos

4.4.2.2. Storyboard Análisis de los datos

Para analizar los datos (Ilustración):

- El usuario pulsa el botón Administrador.
- El sistema muestra las opciones que tiene como Administrador.
- El usuario pulsa el botón Estudiar los datos.
- El sistema analiza los datos y emite un mensaje por pantalla para saber que ha terminado.



Ilustración 4.7: Storyboard Análisis de los datos

4.4.2.3. Storyboard Cross-validation

Para realizar la validación de cruce (Ilustración):

- El usuario pulsa el botón Administrador.
- El sistema muestra las opciones que tiene como Administrador.
- El usuario pulsa el botón Obtener error (RMSE) mediante validación de cruce.
- El sistema realiza la validación de cruce y emite un mensaje por pantalla para saber que la tarea se está realizando.
- El sistema emite otro mensaje para que el usuario sepa que ha finalizado la tarea.
-

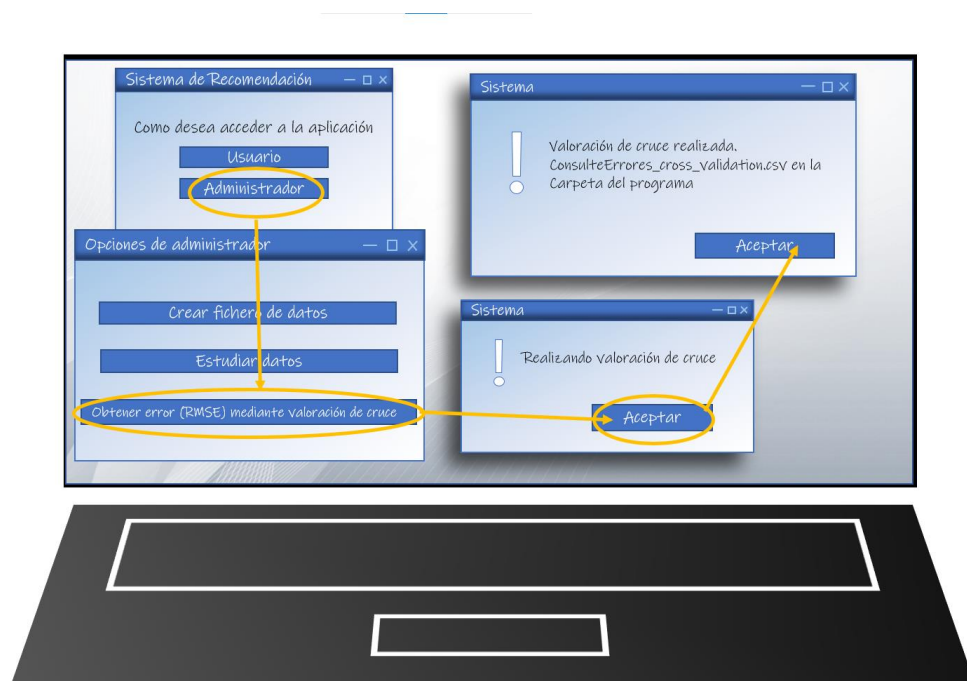


Ilustración 4.8: Storyboard Cross-validation

4.4.2.4. Storyboard Recomendación

Para realizar la recomendación (Ilustración):

- El usuario pulsa el botón Usuario.
- El usuario introduce los datos correctamente para los campos disponibles.
- El usuario pulsa el botón calcular la recomendación.
- El sistema emite un mensaje para saber que se está realizando el proceso de recomendación.
- El sistema emite un nuevo mensaje para que el usuario sepa que se ha terminado el proceso.

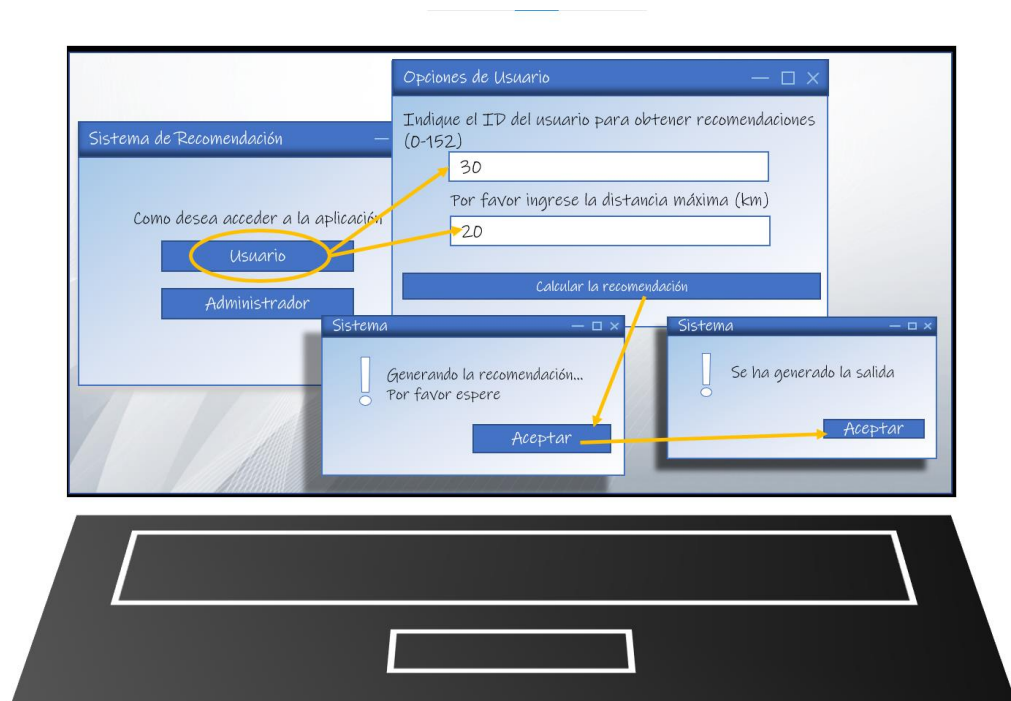


Ilustración 4.9: Storyboard Recomendación

4.5. Implementación

Para la implementación del sistema, no utilizo base de datos, sino ficheros de texto con formato .csv y .kml y esta se comunica con el programa mediante lecturas de los mismos. Pasaremos a explicar lo que es el formato CSV, y como se comunica con el sistema.

4.5.1. Lenguaje del servidor.

Para el lenguaje del servidor utilizaremos los ficheros CSV, proviene del inglés y significa “*Comma Separated Values*” y esto significa valores separados por comas. Un archivo CSV es cualquier fichero de texto cuyos caracteres estén separados por comas, haciendo así una tabla en filas y columnas. En la Ilustración podemos ver un ejemplo de este formato y como se vería en Excel al exportarlo.

Column1	Column2	Column3
43	452	2
143	626	2
49	179	2
112	312	0
42	526	6
106	13	3
16	58	2
89	341	3
105	89	7
71	18	7
82	181	3
130	238	2
142	425	6
135	154	6
79	95	3
129	455	4
31	617	2
79	374	3
94	214	5
63	85	10
126	34	6
89	462	3
5	253	0
79	377	3
103	86	0
141	226	8
78	301	8
105	47	1

Ilustración 4.10: Fichero .csv importado a Excel.

Permite visualizar los datos, modificarlos y eliminarlos muy sencillamente y mediante Excel, por ejemplo, se pueden estudiar los datos y ver sus distribuciones.

Estos ficheros se comunicarán con el cliente mediante la lectura de ficheros y se estructurarán mediante la librería ‘pandas’⁹ de Python.

⁹ Pandas: https://pandas.pydata.org/docs/user_guide/index.html#user-guide

4.5.2. Lenguaje del cliente.

El lenguaje de programación del cliente es Python junto con el uso de módulos o librerías en C/C++.

- Python: Es un lenguaje de programación multiparadigma, esto significa que permite varios estilos, como la programación orientada a objetos, imperativa y funcional. Utiliza tipado dinámico y conteo de referencias para administrar la memoria.

Otro objetivo del diseño del lenguaje es la facilidad de su extensión; ya que se pueden escribir nuevos módulos fácilmente en C/C++.

- C/C++: Es un lenguaje de programación que permite la manipulación de objetos. Estas son algunas de sus características:
 - La sintaxis de C++ es heredada de C.
 - Programación orientada a objetos (POO).
 - Permite agrupar instrucciones.
 - Lenguaje muy didáctico.
 - Es portátil y tiene gran número de compiladores en distintas plataformas y Sistemas Operativos.
 - Permite separar un programa en módulos.
 - Lenguaje de alto nivel.

4.6. Pruebas y Validación

En esta sección del proyecto se realizarán pruebas de caja negra para garantizar la calidad del software desarrollado. Su propósito es encontrar un conjunto de entradas que puedan satisfacer completamente los requisitos funcionales. (En estas pruebas no se tiene en cuenta la estructura de control.)

Para realizar el test, se enumerarán unos casos de test que se corresponden con los requisitos funcionales que defino anteriormente.

4.6.1. Casos de prueba

A continuación, propongo los casos de prueba diseñados:

Test 1: Creación de los datos por parte de administrador. [Requisito Funcional 1]

Pre-condiciones:	El administrador de comprobar que están en la carpeta del programa los ficheros 'usuarios.csv' y 'POIJaen.kml'
Acción:	El administrador pulsa el botón de crear datos.
Checkpoint 1:	El sistema genera los ficheros datosjaen.csv y 'ratings.csv' en la carpeta donde está ubicado el programa.

Test 2: Generar el análisis de los datos. [Requisito Funcional 2]

Pre-condiciones:	El administrador de comprobar que están en la carpeta del programa los ficheros 'usuarios.csv', 'datosjaen.csv' y 'ratings.csv'.
Acción:	El administrador pulsa el botón de analizar los datos.
Checkpoint 1:	El sistema genera el fichero 'EstudioDatos.csv' en la carpeta donde está ubicado el programa.

Test 3: Generar el RMSE de los datos. [Requisito Funcional 3]

Pre-condiciones:	El administrador de comprobar que está en la carpeta del programa el fichero 'ratings.csv'.
Acción:	El administrador pulsa el botón de obtener RMSE de los datos.
Checkpoint 1:	El sistema genera el fichero 'Error_cross_validation.csv' en la carpeta donde está ubicado el programa.

Test 4: Creación de recomendación en una lista. [Requisito Funcional 4]

Pre-condiciones:	El usuario debe introducir correctamente los datos que se le solicitan.
Acción:	El usuario pulsa el botón de generar recomendación
Checkpoint 1:	El sistema genera el fichero 'Recomendación.csv' en la carpeta donde está ubicado el programa.

Test 5: Visualización de recomendación en el mapa. [Requisito Funcional 4]

Pre-condiciones:	El usuario debe introducir correctamente los datos que se le solicitan.
Acción:	El usuario pulsa el botón de generar recomendación
Checkpoint 1:	El sistema genera el fichero 'Mirecomendación.html' en la carpeta donde está ubicado el programa y lo muestra.

4.6.2. Resultados obtenidos

Test 1	
Checkpoint 1	Ok
Test 2	
Checkpoint 1	Ok
Test 3	
Checkpoint 1	Ok
Test 4	
Checkpoint 1	Ok
Test 5	
Checkpoint 1	Ok

4.7. Análisis exploratorio de los datos

Como se puede ver la Ilustración 3.10 nos muestra de cuantas valoraciones de ítems, en este caso ubicaciones, disponemos para cada puntuación. Observamos que están todas entre 3500 y 3700 con lo que podemos ver que el sistema esta balanceado. También se puede observar que en total hay 39999 valoraciones en total.

```
10    3603
9     3709
8     3632
7     3576
6     3594
5     3659
4     3627
3     3634
2     3644
1     3625
0     3696
Name: placeRating, dtype: int64
Int64Index([10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0], dtype='int64')
[3603 3709 3632 3576 3594 3659 3627 3634 3644 3625 3696]
39999
```

Ilustración 4.11: Distribución de ítems por valoraciones.

Podemos observar el ranking del número de valoraciones totales de las que disponemos para cada lugar, y observamos que las primeras están en torno a 80 y 100 valoraciones por lugar Ilustración 3.11 y en torno a 40 y 50 para las últimas Ilustración 3.12. También se observa que disponemos de 643 POI distintos.

```

Recuento Lugar-Puntuacion (Top-10)
      placeID placeRating
138      138          96
257      257          89
161      161          88
184      184          87
514      514          87
35        35          86
612      612          86
69        69          82
350      350          82
566      566          81
418      418          80
320      320          80
224      224          80
176      176          80
641      641          79

```

Ilustración 4.12: Recuento valoraciones totales por lugar. (Top 10)

```

86        86          46
378       378          46
90        90          46
525       525          46
553       553          45
356       356          45
387       387          45
447       447          45
72        72          45
513       513          45
328       328          44
481       481          43
377       377          40
225       225          40

```

```

[643 rows x 2 columns]
Numero de lugares: 643

```

Ilustración 4.13: Recuento valoraciones totales por lugar. (Last 10)

En la Ilustración 3.13 podemos visualizar de cuantas valoraciones disponemos por usuarios y recogemos que los usuarios que más han valorado están en torno a 280 y 300 valoraciones y en la Ilustración 3.14 podemos observar que los usuarios que menos han valorado están en torno a 220 y 240 valoraciones, y que disponemos de 152 usuarios distintos.

Recuento Usuario-Puntuacion (Top 10)		
	userID	placeRating
97	97	301
110	110	297
20	20	294
122	122	294
133	133	293
112	112	291
66	66	291
129	129	289
51	51	288
131	131	285
9	9	284
35	35	284
47	47	283
75	75	282
144	144	281
137	137	281
67	67	280

Ilustración 4.14: Recuento valoraciones totales por usuario. (Top 10)

151	151	243
105	105	243
81	81	243
64	64	243
146	146	242
100	100	240
30	30	239
86	86	237
61	61	237
136	136	236
134	134	234
128	128	234
21	21	229
46	46	228
0	0	228
39	39	226

[152 rows x 2 columns]
Numero de usuarios: 152

Ilustración 4.15: Recuento valoraciones totales por usuario. (Last 10)

5. CONCLUSIONES

El desarrollo de este TFG me ha dado a conocer los Sistemas de Recomendación y a profundizar en el aspecto del IOT que es un campo muy innovador, aparte de aprender a utilizar mejor el lenguaje de programación Python. También he aprendido a utilizar programas como Visual Paradigm y Jupyter Notebook.

Como conclusión respecto de los SR, tras los experimentos realizados puedo concluir que son muy útiles para el día a día y como ya he comentado anteriormente está implementada como algoritmos en plataformas muy famosas como Netflix o Amazon, y esto hace que el usuario tenga muchas más facilidades al satisfacer una necesidad.

Como conclusión final personal, he descubierto un nuevo campo donde me gustaría seguir trabajando e investigando ya que me parece una temática muy útil e interesante, y que a mi parecer le faltan muchas más funcionalidades posibles de implementación.

ANEXO I. Manual de Instalación

Antes de comenzar con el manual de instalación del sistema, procederé a enumerar el contenido del TFG:

- PDF con la documentación del Trabajo Fin de Grado.
- Fichero programa.py con el código del programa.
- Fichero POIJaen.kml con los datos de los lugares de la provincia de Jaén.
- Fichero usuarios.csv con los usuarios registrados en la aplicación.

Dado que el lenguaje está realizado en Python 3.6, deberá tener instalado ese lenguaje en su ordenador.

I.1 Instalación de Python 3.6.

Para instalar Python 3.6 deberá acceder al siguiente enlace <https://www.python.org/downloads/release/python-360/> y descargar el fichero que corresponda de la lista de la Ilustración I.1.

Files					
Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		3f7062ccf8be76491884d0e47ac8b251	22256403	SIG
XZ compressed source tarball	Source release		82b143ebbf4514d7e05876bed7a6b1f5	16805836	SIG
Mac OS X 64-bit/32-bit installer	macOS	for Mac OS X 10.6 and later	72acb0175e7622dec7e1b160a43b8c42	27442222	SIG
Windows help file	Windows		6a842a15ab3b4aa316c91a9779db82ec	7940890	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	0ec0caeea75bae5d2771cf619917c71f	6925798	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	71c9d30c1110abf7f80a428970ab8ec2	31505640	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	25b8b6c93a098dfade3b014630f9508e	1312376	SIG
Windows x86 embeddable zip file	Windows		1adf2fb735c5000af32d42c39136727c	6315855	SIG
Windows x86 executable installer	Windows		38d9b036b25725f6acb553d4aece4db4	30566536	SIG
Windows x86 web-based installer	Windows		f71f4590be2cc5cdc43069594d4ea98d	1286984	SIG

Ilustración I.1: Ficheros de descarga de Python.

Al descargarla ejecutar el fichero descargado, que será un ejecutable con extensión .exe y debe instalar Python 3.6 mediante ese fichero Ilustración I.2. Sabrá que lo ha instalado correctamente cuando aparezca como en la Ilustración I.3.



Ilustración I.2: Ventana de instalación de Python

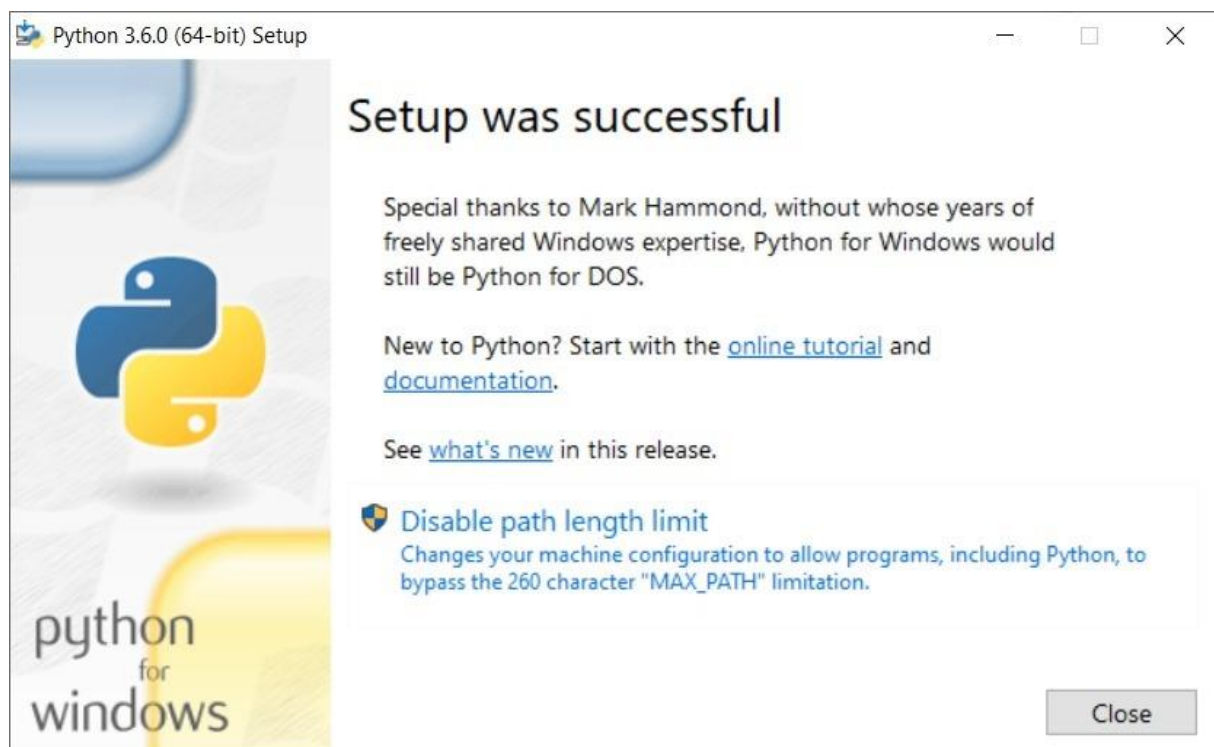


Ilustración I.3: Ventana de instalación correcta de Python

I.2 Instalación de las librerías de Python

Para poder ejecutar el programa deberá también instalar las librerías que se utilizan en el programa, pero antes de instalarlas, necesitareé instalar también Microsoft C++ Build Tools 14 o superior, puede descargarlo haciendo click en el link <https://visualstudio.microsoft.com/visual-cpp-b>. Estará instalado cuando se observe como la Ilustración I.4.

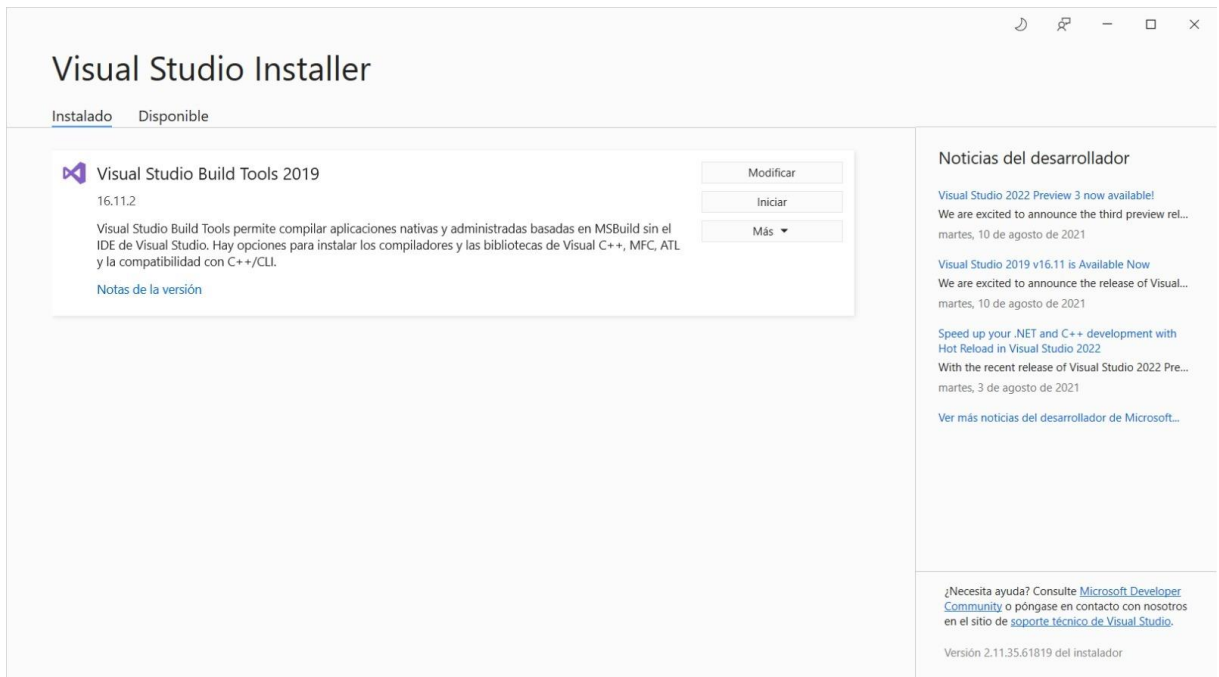


Ilustración I.4: Visualización de los paquetes y versión instalados de Microsoft C++ Build Tools.

A continuación, pongo todas las instrucciones necesarias para instalar las librerías necesarias y para saber que se han instalado correctamente debería verse como se verá a continuación.

- `pip install pykml` (Ilustración I.5)
- `pip install pandas` (Ilustración I.6)
- `pip install --upgrade setuptools` (Ilustración I.7)
- `pip install surprise` (Ilustración I.8)
- `pip install geopy` (Ilustración I.9)
- `pip install folium` (Ilustración I.10)
- `pip install Jinja2==3.0.0rc2` (Ilustración I.11)

```
C:\Users\jtmer\OneDrive\Escritorio\Proyecto>pip install pykml
Collecting pykml
  Downloading https://files.pythonhosted.org/packages/b8/22/8b3e7aec303a3d11bc62de04d863cf2092d7a722ade35809f7f6232df164/pykml-0.2.0-py3-none-any.whl (41kB)
    100% |#####| 51kB 743kB/s
Collecting lxml>=3.3.6 (from pykml)
  Downloading https://files.pythonhosted.org/packages/a0/e6/653cddfc52f252869be6db7cdd8140a3a6f7c9c2a4cd58a869dcc37efb9/lxml-4.6.3-cp36-cp36m-win_amd64.whl (3.5MB)
    100% |#####| 3.5MB 277kB/s
Installing collected packages: lxml, pykml
Successfully installed lxml-4.6.3 pykml-0.2.0
```

Ilustración I.5: Visualización de instalación de librería pykml desde el terminal.

```
C:\Users\jtmer\OneDrive\Escritorio\Proyecto>pip install pandas
Collecting pandas
  Downloading https://files.pythonhosted.org/packages/f0/ac/92c3d2f0b627efbd1a7b2156faa697f9c2bbd7b0fe83ba8a9d36f982156f/pandas-0.25.3-cp36-cp36m-win_amd64.whl (9.0MB)
    100% |#####| 9.0MB 106kB/s
Collecting pytz>=2017.2 (from pandas)
  Downloading https://files.pythonhosted.org/packages/70/94/784178ca5dd892a98f113cdd923372024dc04b8d40abe77ca76b5fb90ca6/pytz-2021.1-py2.py3-none-any.whl (510kB)
    100% |#####| 512kB 1.6MB/s
Collecting numpy>=1.13.3 (from pandas)
  Downloading https://files.pythonhosted.org/packages/ea/bc/da526221bc111857c7ef39c3af670bbcf5e69c247b0d22e51986f6d0c5c2/numpy-1.19.5-cp36-cp36m-win_amd64.whl (13.2MB)
    100% |#####| 13.2MB 73kB/s
Collecting python-dateutil>=2.6.1 (from pandas)
  Downloading https://files.pythonhosted.org/packages/36/7a/87837f39d0296e723bb9b62bbb257d0355c7f6128853c78955f57342a56d/python_dateutil-2.8.2-py2.py3-none-any.whl (247kB)
    100% |#####| 256kB 1.2MB/s
Collecting six>=1.5 (from python-dateutil>=2.6.1->pandas)
  Downloading https://files.pythonhosted.org/packages/d9/5a/e7c31adbe875f2abbb91bd84cf2dc52d792b5a01506781dbcf25c91daf11/six-1.16.0-py2.py3-none-any.whl
Installing collected packages: pytz, numpy, six, python-dateutil, pandas
Successfully installed numpy-1.19.5 pandas-0.25.3 python-dateutil-2.8.2 pytz-2021.1 six-1.16.0
```

Ilustración I.6: Visualización de instalación de librería pandas desde el terminal.

```
C:\Users\jtmer\OneDrive\Escritorio\Proyecto>pip install --upgrade setuptools
Requirement already satisfied: setuptools in c:\users\jtmer\appdata\local\programs\python\python36\lib\site-packages (28.8.0)
Collecting setuptools
  Downloading setuptools-58.0.3-py3-none-any.whl (816 kB)
    ##### 816 kB 3.3 MB/s
Installing collected packages: setuptools
  Attempting uninstall: setuptools
    Found existing installation: setuptools 28.8.0
    Uninstalling setuptools-28.8.0:
      Successfully uninstalled setuptools-28.8.0
Successfully installed setuptools-58.0.3
```

Ilustración I.7: Visualización de actualización del paquete setuptools desde el terminal.

```
C:\Users\jtmer\OneDrive\Escritorio\Proyecto>pip install surprise
Collecting surprise
  Using cached surprise-0.1-py2.py3-none-any.whl (1.8 kB)
Collecting scikit-surprise
  Using cached scikit-surprise-1.1.1.tar.gz (11.8 MB)
Requirement already satisfied: joblib>=0.11 in c:\users\jtmer\appdata\local\programs\python\python36\lib\site-packages (from scikit-surprise->surprise) (1.0.1)
Requirement already satisfied: numpy>=1.11.2 in c:\users\jtmer\appdata\local\programs\python\python36\lib\site-packages (from scikit-surprise->surprise) (1.19.5)
Requirement already satisfied: scipy>=1.0.0 in c:\users\jtmer\appdata\local\programs\python\python36\lib\site-packages (from scikit-surprise->surprise) (1.5.4)
Requirement already satisfied: six>=1.10.0 in c:\users\jtmer\appdata\local\programs\python\python36\lib\site-packages (from scikit-surprise->surprise) (1.16.0)
Using legacy 'setup.py install' for scikit-surprise, since package 'wheel' is not installed.
Installing collected packages: scikit-surprise, surprise
  Running setup.py install for scikit-surprise ... done
Successfully installed scikit-surprise-1.1.1 surprise-0.1
```

Ilustración I.8: Visualización de instalación de librería surprise desde el terminal.

```
C:\Users\jtmer\OneDrive\Escritorio\Proyecto>pip install geopy
Collecting geopy
  Downloading geopy-2.2.0-py3-none-any.whl (118 kB)
    | 118 kB 2.2 MB/s
Collecting geographiclib<2,>=1.49
  Downloading geographiclib-1.52-py3-none-any.whl (38 kB)
Installing collected packages: geographiclib, geopy
Successfully installed geographiclib-1.52 geopy-2.2.0
```

Ilustración I.9: Visualización de instalación de librería geopy desde el terminal.

```
C:\Users\jtmer\OneDrive\Escritorio\Proyecto>pip install folium
Collecting folium
  Downloading folium-0.12.1-py2.py3-none-any.whl (94 kB)
    | 94 kB 3.3 MB/s
Requirement already satisfied: numpy in c:\users\jtmer\appdata\local\programs\python\python36\lib\site-packages (from folium) (1.19.5)
Collecting branca>=0.3.0
  Downloading branca-0.4.2-py3-none-any.whl (24 kB)
Collecting Jinja2>=2.9
  Downloading Jinja2-3.0.1-py3-none-any.whl (133 kB)
    | 133 kB ...
Collecting requests
  Downloading requests-2.26.0-py2.py3-none-any.whl (62 kB)
    | 62 kB 4.8 MB/s
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.0.1-cp36-cp36m-win_amd64.whl (14 kB)
Collecting charset-normalizer~2.0.0
  Downloading charset-normalizer-2.0.4-py3-none-any.whl (36 kB)
Collecting urllib3<1.27,>=1.21.1
  Downloading urllib3-1.26.6-py2.py3-none-any.whl (138 kB)
    | 138 kB ...
Collecting certifi>=2017.4.17
  Downloading certifi-2021.5.30-py2.py3-none-any.whl (145 kB)
    | 145 kB 6.8 MB/s
Collecting idna<4,>=2.5
  Downloading idna-3.2-py3-none-any.whl (59 kB)
    | 59 kB ...
Installing collected packages: MarkupSafe, urllib3, Jinja2, idna, charset-normalizer, certifi, requests, branca, folium
Successfully installed MarkupSafe-2.0.1 branca-0.4.2 certifi-2021.5.30 charset-normalizer-2.0.4 folium-0.12.1 idna-3.2 j...
```

Ilustración I.10: Visualización de instalación de librería folium desde el terminal.

```
C:\Users\jtmer\OneDrive\Escritorio\Proyecto>pip install Jinja2==3.0.0rc2
Collecting Jinja2==3.0.0rc2
  Downloading Jinja2-3.0.0rc2-py3-none-any.whl (125 kB)
    | 125 kB 1.7 MB/s
Requirement already satisfied: MarkupSafe>=2.0.0rc2 in c:\users\jtmer\appdata\local\programs\python\python36\lib\site-packages (from Jinja2==3.0.0rc2) (2.0.1)
Installing collected packages: Jinja2
  Attempting uninstall: Jinja2
    Found existing installation: Jinja2 3.0.1
    Uninstalling Jinja2-3.0.1:
      Successfully uninstalled Jinja2-3.0.1
Successfully installed Jinja2-3.0.0rc2
```

Ilustración I.11: Visualización de instalación de librería Jinja2 desde el terminal.

ANEXO II. Manual de Usuario

En este manual vamos a describir cómo acceder a toda la funcionalidad de la aplicación para permitir realizar un uso correcto de la misma. Para ello, seguiré un orden de ejecución de acuerdo al orden que se han colocado las distintas funciones dentro de la pantalla.

II.1 Ejecución del programa.

Para la ejecución del programa se necesitan los siguientes ficheros como mínimo en la carpeta donde se ubica el programa:

- programa.py
- POIJaen.kml
- Usuarios.csv

Una vez nos aseguramos que tenemos los ficheros, procederemos a ir a ejecutar el terminal e introducir las siguientes instrucciones.

- `cd ubicacion_carpeta_proyecto.`
- `python programa.py.`

Cuando hemos ejecutado las instrucciones, se abrirá la interfaz del programa como en la Ilustración II.1.

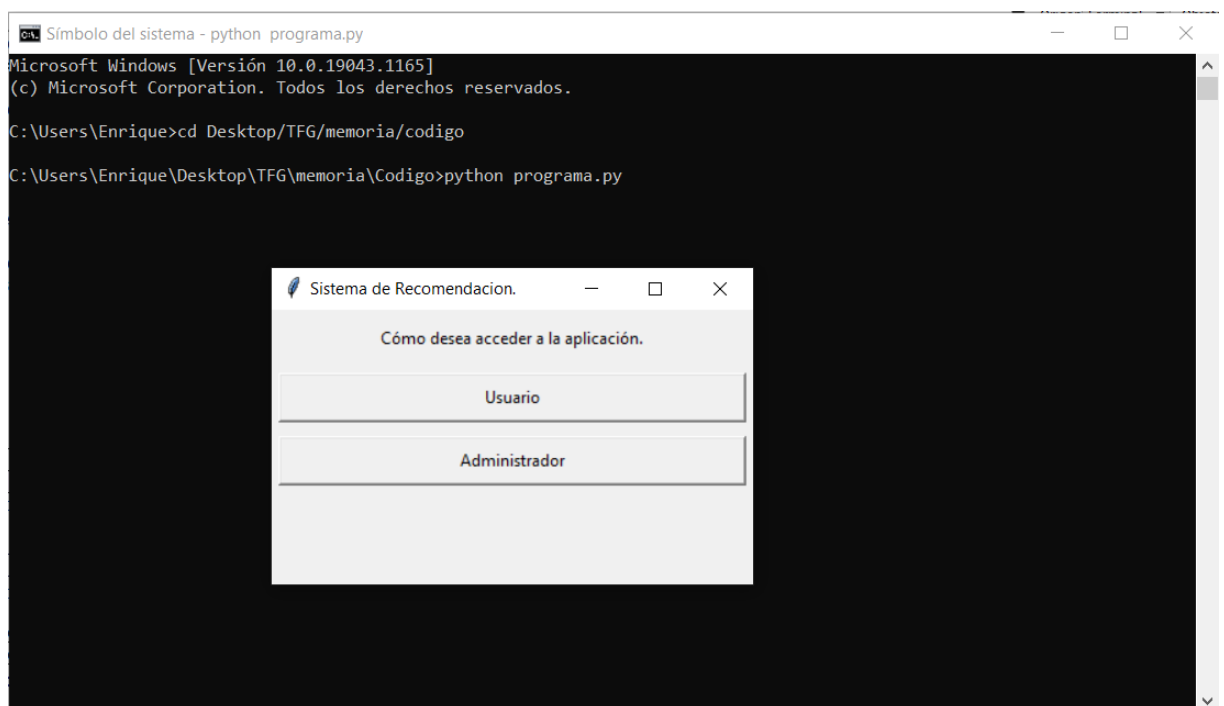


Ilustración II.1: Visualización de la interfaz del programa.

II.2 Menú Administrador

Para acceder al menú con las opciones del administrador (Ilustración II.2), al ejecutarse la aplicación se pulsará el botón de ‘*Administrador*’ y se abrirá otra ventana para ver de las que dispone. La primera “*Crear ficheros de datos*” donde se podrán crear los ficheros ‘datosjaen.csv’ y ‘ratings.csv’, que utilizaremos en el sistema de recomendación; la segunda opción “*Estudiar los datos*” crea el fichero ‘EstudioDatos.csv’ con el análisis de los datos creados y por último la opción “*Obtener error (RMSE) mediante validación de cruce*”, crea el fichero ‘Errores_cross_validation.csv’ con los errores de los sets de prueba y de entrenamiento.

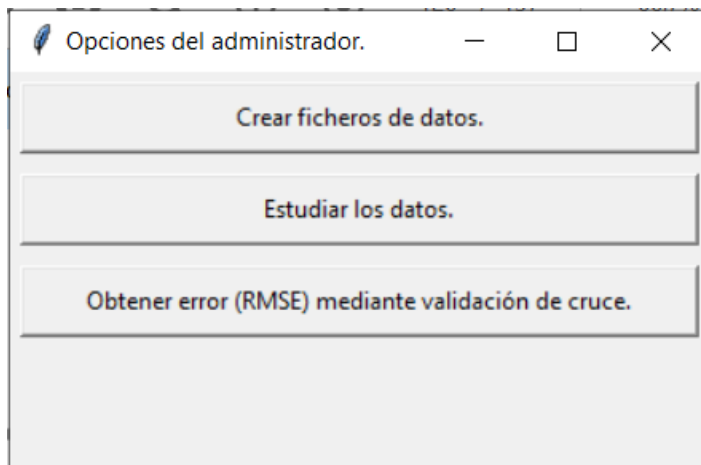


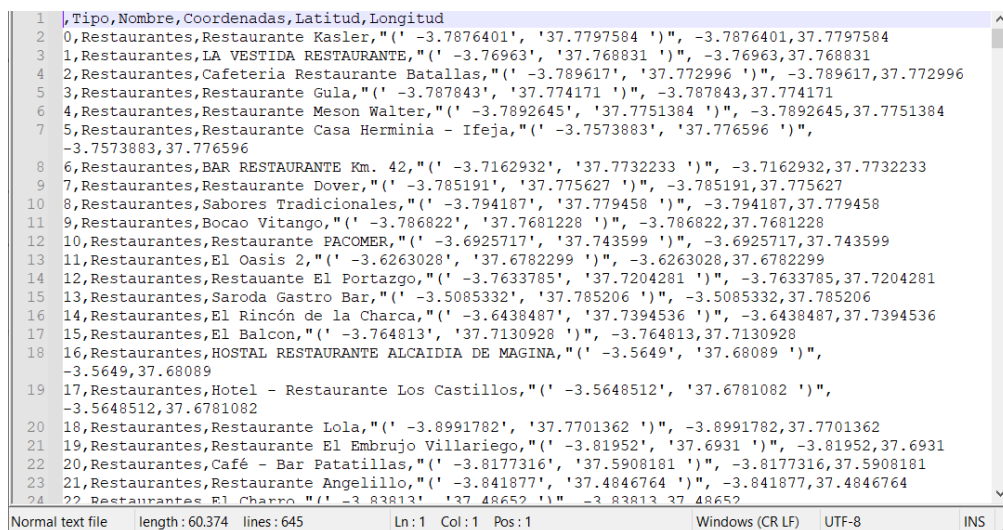
Ilustración II.2: Visualización de las opciones de administrador.

II.2.1 Crear ficheros de datos

Para crear los ficheros de los datos que vamos a utilizar en el programa, hay que pulsar en el menú de Administrador la opción “Crear ficheros de datos.”

Al pulsarlo el sistema emitirá un mensaje de que se han creado los ficheros en la carpeta y serán de la siguiente forma.

Datosjaen.csv (Ilustración II.3)



```

1 ,Tipo,Nombre,Coordenadas,Latitud,Longitud
2 0,Restaurantes,Restaurante Kasler,"(' -3.7876401', '37.7797584 ')", -3.7876401,37.7797584
3 1,Restaurantes,LA VESTIDA RESTAURANTE,"(' -3.76963', '37.768831 ')", -3.76963,37.768831
4 2,Restaurantes,Cafeteria Restaurante Batallas,"(' -3.789617', '37.772996 ')", -3.789617,37.772996
5 3,Restaurantes,Restaurante Gula,"(' -3.787843', '37.774171 ')", -3.787843,37.774171
6 4,Restaurantes,Restaurante Meson Walter,"(' -3.7892645', '37.7751384 ')", -3.7892645,37.7751384
7 5,Restaurantes,Restaurante Casa Herminia - Ifeja,"(' -3.7573883', '37.776596 ')",
  -3.7573883,37.776596
8 6,Restaurantes,BAR RESTAURANTE Km. 42,"(' -3.7162932', '37.7732233 ')", -3.7162932,37.7732233
9 7,Restaurantes,Restaurante Dover,"(' -3.785191', '37.775627 ')", -3.785191,37.775627
10 8,Restaurantes,Sabores Tradicionales,"(' -3.794187', '37.779458 ')", -3.794187,37.779458
11 9,Restaurantes,Bocao Vitango,"(' -3.786822', '37.7681228 ')", -3.786822,37.7681228
12 10,Restaurantes,Restaurante PACOMER,"(' -3.6925717', '37.743599 ')", -3.6925717,37.743599
13 11,Restaurantes,El Oasis 2,"(' -3.6263028', '37.6782299 ')", -3.6263028,37.6782299
14 12,Restaurantes,Restaurante El Portazgo,"(' -3.7633785', '37.7204281 ')", -3.7633785,37.7204281
15 13,Restaurantes,Saroda Gastro Bar,"(' -3.5085332', '37.785206 ')", -3.5085332,37.785206
16 14,Restaurantes,El Rincón de la Charca,"(' -3.6438487', '37.7394536 ')", -3.6438487,37.7394536
17 15,Restaurantes,El Balcon,"(' -3.764813', '37.7130928 ')", -3.764813,37.7130928
18 16,Restaurantes,HOSTAL RESTAURANTE ALCAIDIA DE MAGINA,"(' -3.5649', '37.68089 ')",
  -3.5649,37.68089
19 17,Restaurantes,Hotel - Restaurante Los Castillos,"(' -3.5648512', '37.6781082 ')",
  -3.5648512,37.6781082
20 18,Restaurantes,Restaurante Lola,"(' -3.8991782', '37.7701362 ')", -3.8991782,37.7701362
21 19,Restaurantes,Restaurante El Embrujo Villariego,"(' -3.81952', '37.6931 ')", -3.81952,37.6931
22 20,Restaurantes,Café - Bar Patatillas,"(' -3.8177316', '37.5908181 ')", -3.8177316,37.5908181
23 21,Restaurantes,Restaurante Angelillo,"(' -3.841877', '37.4846764 ')", -3.841877,37.4846764
24 22,Restaurantes,El Charro,"(' -3.83813', '37.48652 ')", -3.83813,37.48652

```

Normal text file | length: 60.374 | lines: 645 | Ln:1 | Col:1 | Pos:1 | Windows (CR LF) | UTF-8 | INS

Ilustración II.3: Visualización del fichero datosjaen.csv.

Ratings.csv (Ilustración II.4)



```

1 103,466,8
2 121,96,1
3 69,246,1
4 143,612,10
5 109,249,9
6 40,537,7
7 116,462,2
8 137,441,1
9 40,358,3
10 130,596,10
11 141,489,4
12 73,206,0
13 11,70,7
14 32,410,2
15 3,368,9
16 11,532,5
17 31,519,3
18 55,35,10
19 1,23,4
20 78,222,3
21 112,518,4
22 29,332,0
23 20,58,10
24 24,606,4
25 144,310,7
26 90,359,4
27 101,224,2

```

Normal text file | length: 407.864 | lines: 40.001 | Ln:1 | Col:1 | Pos:1 | Windows (CR LF) | UTF-8 | INS

Ilustración II.4: Visualización del fichero ratings.csv.

II.2.2 Estudiar los datos

Para estudiar los datos que vamos a utilizar en el programa, hay que pulsar en el menú de Administrador la opción “Estudiar los datos.”

Al pulsarlo el sistema emitirá un mensaje de que se han creado el fichero resultante al realizar el estudio en la carpeta y será de la forma de la Ilustración II.5.

Puede visualizar la distribución de los datos y ver si están balanceados para saber si la recomendación que hará con esos datos tendrá suficiente información o no.

EstudioDatos.csv

```

1 Total de valoraciones: 39999
2 Distribucion de valoraciones de items para cada puntuacion.
3 10 3628
4 9 3700
5 8 3679
6 7 3655
7 6 3686
8 5 3595
9 4 3571
10 3 3678
11 2 3593
12 1 3614
13 0 3600
14 Name: placeRating, dtype: int64
15
16
17
18
19
20 Numero total de POI: 643
21 Recuento POI-Puntuacion
22 | placeID placeRating
23 230 230 89
24 244 244 87
25 249 249 83
26 498 498 82
27 88 88 82
28 .. ... ...
29 390 390 44
30 617 617 43
31 295 295 43
32 186 186 42
33 461 461 42
34
35 [643 rows x 2 columns]
36
37
38
39
40
41 Numero total de usuarios: 152
42 Recuento Usuario-Puntuacion
43 | userID placeRating
44 148 148 321
45 125 125 304
46 32 32 301
47 43 43 299
48 54 54 296
49 .. ... ...
50 111 111 235
51 67 67 234
52 98 98 233
53 42 42 230
54 23 23 227
55

```

Ilustración II.5: Visualización del fichero EstudioDatos.csv.

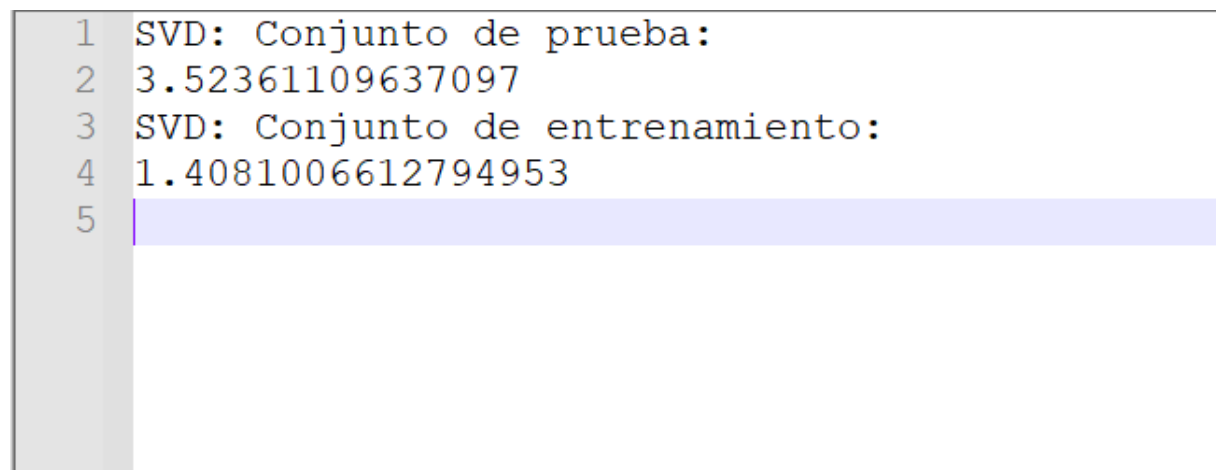
II.2.3 Obtener error (RMSE) mediante validación de cruce

Para realizar la validación de cruce con el algoritmo SVD y visualizar el error cuadrático medio (RMSE) con los datos que disponemos, hay que pulsar en el menú de Administrador la opción “Obtener error (RMSE) mediante validación de cruce.”

Al pulsarlo el sistema emitirá un mensaje de que se está realizando la validación de cruce y al acabar se generará otro y generará un fichero que será de la forma de la Ilustración II.6.

Puede visualizar la distribución de los datos y ver si están balanceados para saber si la recomendación que hará con esos datos tendrá suficiente información o no.

Errores_cross_validation.csv ()



```
1 SVD: Conjunto de prueba:
2 3.52361109637097
3 SVD: Conjunto de entrenamiento:
4 1.4081006612794953
5
```

Ilustración II.6: Visualización del fichero Errores_cross_validation.csv.

II.3 Menú Usuario

Para acceder al menú con las opciones del usuario, al ejecutarse la aplicación se pulsará el botón de '*Usuario*' y se abrirá otra ventana con la única opción del usuario (Ilustración II.7), que será para realizar la recomendación.

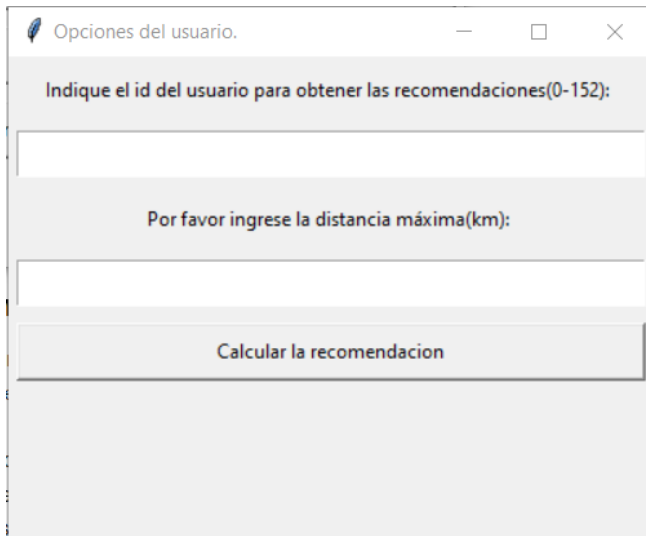


Ilustración II.7: Ventana de la opción de usuario para realizar la recomendación.

II.3.1 Obtener la recomendación

Para calcular la recomendación deberás insertar en el campo primero el identificador del usuario del que quieres obtener la recomendación, y deberá estar comprendido entre 0 y 152 incluidos ya que el identificador del usuario son enteros únicos del 0 al 152. En el segundo campo deberá ingresar es la distancia máxima que quiere utilizar, y deberá ser entre 0 y 200. Si el sistema detecta que los datos no están dentro de los rangos se lo dirá dispondrá en la ventana.

Una vez que ha introducido los datos correctamente al pulsar el botón “Calcular la recomendación” y el sistema generará un mensaje de que se está generando la recomendación. Debe pulsar “Aceptar” para continuar y esperar como le indica en el mensaje.

Al esperar unos segundos el sistema generará un segundo mensaje diciendo que ya se ha realizado la recomendación y debe pulsar “Aceptar” de nuevo.

Se generará el fichero “Mirecomendación.html” en la carpeta del programa, que se abrirá automáticamente en una pestaña del navegador del ordenador con la forma de la Ilustración II.8.

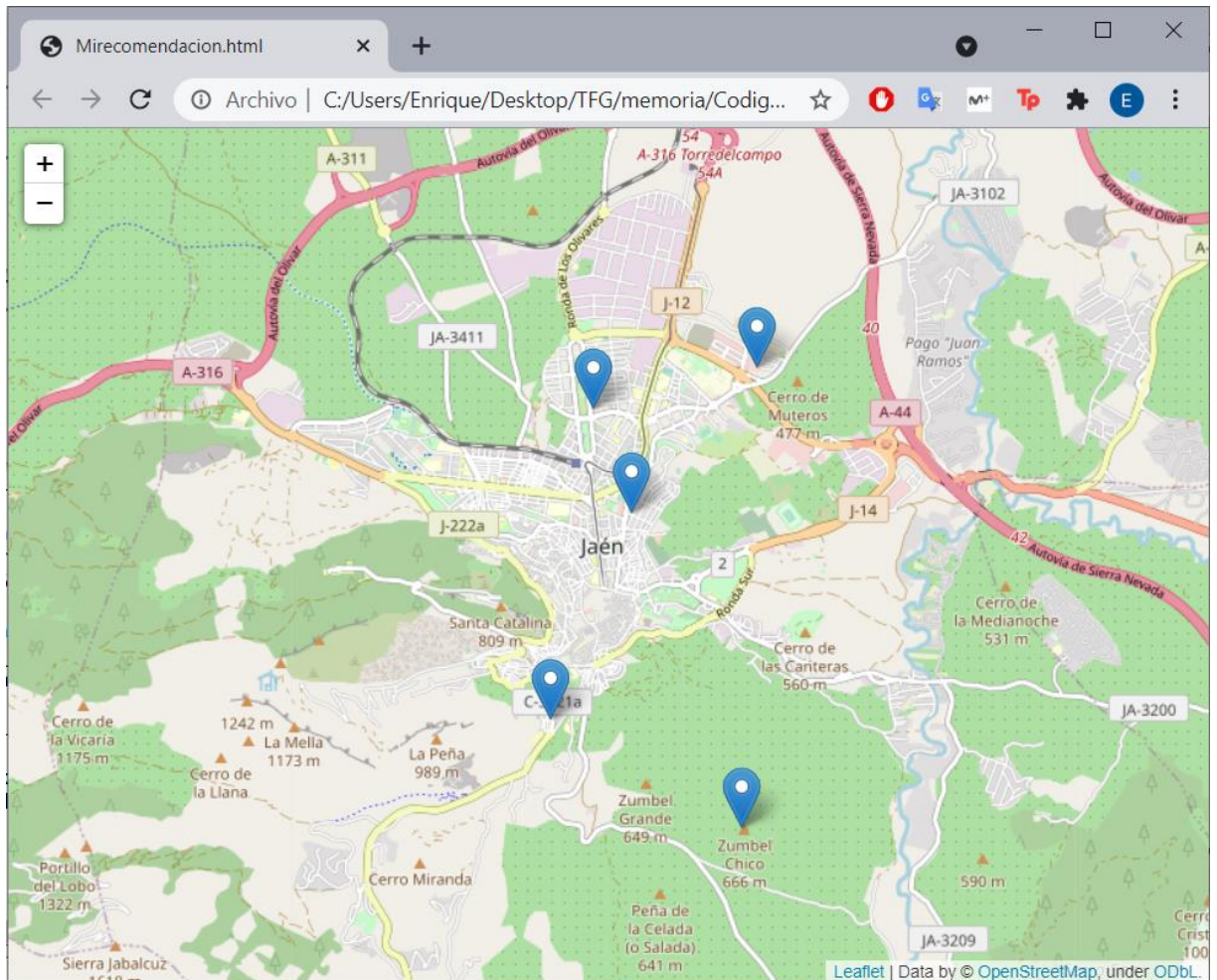


Ilustración II.8: Visualización de la salida del programa al realizar la recomendación.

Bibliografía

- Aggarwal., C. C. (2006). Recommender Systems: The Textbook. Springer International Publishing,.
- Burke, R. (2004). Hybrid recommender systems: Algorithms and evaluation. ACM Transactions on information Systems (TOIS).
- Emmanuel Stefanakis and Kostas Patroumpas. (2008). Google Earth and XML: Advanced Visualization and Publishing of Geographic Information.
- Goldberg, D. e. (1992). Using collaborative filtering to weave an information tapestry. Interacting with Computers of the ACM, 1992.
- Herlocker, J. e. (2004.). Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems.
- Heyde, L. D. (2016). UML 2.5 iniciación, ejemplos y ejercicios corregidos 4ª Edición.
- Kolodner, J. (1992). An introduction to case-based reasoning. Artificial Intelligence Review, .
- Michael D. Ekstrand, J. T. (s.f.). Collaborative Filtering Recommender Systems.
- Ricci F., R. L. (2011). Introduction to Recommender Systems Handbook.
- Rinaudo, G. P. (s.f.). Ingeniería del Software.
- Sommerville, I. (2005). Ingeniería del Software 7ª Edición. Madrid: Pearson Educacion S.A.
- Vekariya, V. &. (2012). Hybrid recommender systems: Survey and experiments. Second International Conference on Digital Information and Communication Technology and its Applications (DICTAP).
- ¿Qué es KML?—Ayuda | ArcGIS for Desktop. (s. f.). KML. Recuperado 25 de agosto de 2021, de <https://desktop.arcgis.com/es/arcmap/10.3/manage-data/kml/what-is-kml-.htm>
- Sistemas de Recomendación. (2020, 3 marzo). Aprende Machine Learning. <https://www.aprendemachinelearning.com/sistemas-de-recomendacion/>

- C. (s. f.-a). Datasets-for-Recommender-Systems/README.md at master · caserec/Datasets-for-Recommender-Systems. Public Datasets For Recommender Systems. Recuperado 25 de agosto de 2021, de <https://github.com/caserec/Datasets-for-Recommender-Systems/blob/master/README.md>
- C. (s. f.-b). GitHub - caserec/Datasets-for-Recommender-Systems: This is a repository of a topic-centric public data sources in high quality for Recommender Systems (RS). GitHub. Recuperado 25 de agosto de 2021, de <https://github.com/caserec/Datasets-for-Recommender-Systems>
- McAuley, J. (s. f.). Recommender Systems Datasets. Recommender Systems and Personalization Datasets. Recuperado 25 de agosto de 2021, de <https://cse-web.ucsd.edu/%7Ejmcauley/datasets.html>
- P. (2020, 22 agosto). Recommender Systems in Python. Kaggle. <https://www.kaggle.com/prashant111/recommender-systems-in-python>
- Choudhury, A. (2020, 26 abril). 10 Open-Source Datasets One Must Know To Build Recommender Systems. Analytics India Magazine. <https://analyticsindia-mag.com/10-open-source-datasets-one-must-know-to-build-recommender-systems/>
- F. (s. f.). Sistemas de recomendación en Python. Sistemas de Recomendación En Python. Recuperado 25 de agosto de 2021, de <https://blog.finde-mor.es/2018/02/sistemas-de-recomendacion-en-python/>
- MovieLens. (2021, 2 marzo). GroupLens. <https://grouplens.org/datasets/movielens/>
- Recommendation and Ratings Public Data Sets For Machine Learning. (s. f.). Gist. Recuperado 25 de agosto de 2021, de <https://gist.github.com/entaroa-dun/1653794>
- TMDB 5000 Movie Dataset. (2017, 28 septiembre). Kaggle. <https://www.kaggle.com/tmdb/tmdb-movie-metadata>
- ¿Qué es SIG? | Sistemas de Información Geográfica - Aeroterra S.A. (s. f.). ¿Que es SIG? Recuperado 26 de agosto de 2021, de <https://www.aeroterra.com/es-ar/que-es-gis/introduccion>
- Gonzalez, L. (2020, 20 agosto). Introducción al IDE Spyder. Aprende IA. <https://aprendeia.com/ide-spyder-para-python/>

- N. (2011, 25 abril). Migrating from MATLAB to Python | Greener Engineering. Python.
<https://web.archive.org.trans-late.goog/web/20141010221143/http://web.ics.purdue.edu:80/%7Es-mit1447/blog/?p=24& x tr sl=en& x tr tl=es& x tr hl=es& x tr pto=ajax,se,elem,sc>
- N. (2019). ¿Qué es Data Science? Data Science. <https://www.neoland.es/blog/que-es-data-science>
- Saavedra, J. (2017, 19 diciembre). Sistemas de recomendación, parte 5: modelos complejos. Medium. <https://medium.com/@eng.saavedra/sistemas-de-recomendaci%C3%B3n-parte-5-modelos-complejos-c8db70e8b7b>
- SpyderPlugins - spyderlib - Plugin development - Spyder is the Scientific PYthon Development EnviRonment - Google Project Hosting. (s. f.). Spyder Plugins. Recuperado 31 de agosto de 2021, de <https://web.archive.org.trans-late.goog/web/20131024165518/http://code.google.com/p/spyderlib/wiki/SpyderPlugins? x tr sl=en& x tr tl=es& x tr hl=es& x tr pto=ajax,se,elem,sc>
- Statista. (2019a, enero 9). Los lenguajes de programación más usados del mundo. Statista Infografías. <https://es.statista.com/grafico/16580/lenguajes-de-programacion-mas-usados-del-mundo/>
- Statista. (2019b, abril 17). A la espera de un Big Bang de datos. Statista Infografías. <https://es.statista.com/grafico/17734/cantidad-real-y-prevista-de-datos-generados-en-todo-el-mundo/>
- U. (2021, 31 agosto). Tabla comparativa de los lenguajes de programación. Lenguajes de programación. <http://desarrollowebydesarrolloweb.blogspot.com/2015/02/tabla-comparativa-de-los-lenguajes-de.html>
- Sistemas de recomendación con Surprise - Antonio David Pérez Morales y Rafael Haro Ramos. (2020, 15 febrero). [Vídeo]. YouTube. https://www.youtube.com/watch?v=aUZZ9AIMpx8&ab_channel=PythonEspa%C3%B1a
- M. (2021, 6 febrero). Ingeniería de software: Qué es, objetivos, características y más. Carreras Universitarias. <https://micarrerauniversitaria.com/c-ingenieria/ingenieria-de-software/>
- Martínez, Q. M. Á. (s. f.). Modelo de recomendación basado en conocimiento para el desarrollo del pensamiento del trabajo con objetos de aprendizaje. SRBC. Recuperado 6 de septiembre de 2021, de

http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1990-86442020000400111

Olguin, M. (s. f.). Sistemas de recomendacion. SR. Recuperado 6 de septiembre de 2021, de <https://www.slideshare.net/molguinm/sistemas-de-recomendacion-67731071>

Richard. (2017, 2 junio). ¿Que son los Sistemas de Recomendación? Jarroba. <https://jarroba.com/que-son-los-sistemas-de-recomendacion/>

Equipo editorial, Etecé. (2021, 6 agosto). Sistema de Información - Concepto, tipos, elementos y ejemplos. Concepto. <https://concepto.de/sistema-de-informacion/>

Garcia, P. (2021, 11 agosto). ¿Qué es un SIG, GIS o Sistema de Información Geográfica? Geoinnova. <https://geoinnova.org/blog-territorio/que-es-un-sig-gis-o-sistema-de-informacion-geografica/#:%7E:text=La%20definici%C3%B3n%20m%C3%A1s%20b%C3%A1sica%20es,acompa%C3%B1ada%20de%20una%20posici%C3%B3n%20geogr%C3%A1fica>.

¿Qué es un archivo CSV y para qué sirve? [Septiembre 2021]. (s. f.). GEEKNETIC. Recuperado 9 de septiembre de 2021, de <https://www.geeknetic.es/Archivo-CSV/que-es-y-para-que-sirve>