eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea
INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Grado en Ingeniería Informática
Computación

Trabajo de Fin de Grado

Desarrollo e implementación de un sistema inteligente de recomendación para salones de cómic

Autora
Enya Goñi Maganto

Director
Josu Ceberio Uribe

Agradecimientos

Dar las gracias a mi director, Josu, por haber confiado en mí. Por haber captado mi visión del proyecto y haberme guiado para que la convirtiera en una realidad de la mejor forma posible, y por haber sacado lo mejor de mí siempre. Además, agradecerle a él y a Anaje por haber estado dispuestos a ayudarme en lo que necesitara a lo largo del proceso. Si al entrar en la universidad me hubiesen dicho que iba a encontrar a estas dos personas por el camino no me lo hubiese creído. Han hecho que lo difícil pareciera fácil y he aprendido tanto de ellos. No habría podido tener a dos personas mejores conmigo. Muchísimas gracias, de verdad. Trabajar con vosotros ha sido una maravilla.

Y, por supuesto, agradecer a mi padre, a Maialen y a Julen por haber estado a mi lado y haberme apoyado en los mejores y en los peores días, por haber celebrado conmigo mis victorias y haberme animado en los peores momentos. Por haber sido mi roca. Ha sido un proceso largo, y no siempre ha sido todo lo fácil que hubiese deseado, así que haberles tenido junto a mí no ha tenido precio.

Prólogo

El presente documento describe el desarrollo del Trabajo de Fin de Grado titulado "Desarrollo e implementación de un sistema inteligente de recomendación para salones de cómic", presentado en la Facultad de Informática de San Sebastián (Universidad del País Vasco) como parte del Grado en Ingeniería Informática. La autora del proyecto es Enya Goñi.

La idea de este proyecto nació del Salón Internacional de Cómic y Manga de San Sebastián. Tengo el privilegio de poder decir que, en todas sus ediciones, he formado parte de la Organización del evento, inicialmente como Creadora de Contenidos, actualmente en calidad de Subdirectora. A partir de ahí, desde el instante en que empezamos a detectar los retos que se plantean en un Salón, empecé a gestar la idea de este Trabajo de Fin de Grado que se describe a continuación.

Índice de contenidos

Índice de figuras

1 Introducción

Con el fin de contextualizar y motivar el proyecto, en está sección introductoria se realiza una descripción del proyecto que se plantea.

1.1 Contexto del proyecto

Un Salón de cómic. Un lugar donde conviven, normalmente durante un fin de semana, miles de personas con un denominador común: descubrir los contenidos que la organización ha preparado para ellas. Exposiciones, charlas, actividades, concursos, juegos...contenidos que pueden ser continuos, puntuales o repetidos a lo largo del evento. Es importante tener en consideración que, bajo el paraguas de un título tan genérico como 'Salón de cómic', allí conviven todos aquellos universos que hoy son inseparables de las viñetas: álbumes, revistas, ilustración, juegos de mesa avanzados, rol, literatura, series y películas de fantasía, terror o ciencia ficción, coleccionismo, *merchandising* centrado en todo lo mencionado... El público paga su entrada y, a cambio, espera actividades, exposiciones, sorpresas. Espera, una vez accede a un recinto de miles de metros cuadrados donde se suceden acontecimientos de forma simultánea, orientación, no desperdiciar su tiempo.

Los salones de cómic son un proceso que ya se repite en el mundo entero, encabezado por eventos de tal magnitud que ya nada tienen de artesanal; San Diego, Tokio, Angoulême, México D.F, Milán, Londres, Berlín, Buenos Aires... Después de años, muchas ciudades repartidas por el planeta cuentan con su festival, más o menos exclusivo, pero todos ellos con el cómic como piedra angular.

Paradójicamente, no existía en Donostia, sede de eventos como el Festival de Cine, el Jazzaldia o Gastronomika, aun cuando el País Vasco ha sido y sigue siendo cuna de artistas muy destacados del llamado noveno arte, como Mata, Redondo, Holgado, Landa, Unzueta...

Hace casi tres décadas se produjo el primer intento de instaurar el primer festival en Donostia, pero tanto éste como los dos posteriores no vieron la luz por determinadas circunstancias financieras e institucionales que dejaron en barbecho el proyecto. Hace tres años, sin embargo, de la mano de otros promotores, un nuevo intento fructificaba bajo el prisma de intentar satisfacer todas las sensibilidades. Así, nacía en 2017 el primer Salón Internacional de Cómic y Manga de Donostia, cuya sede sería el Palacio de Congresos Kursaal, un edificio emblemático que intentaba enviar un mensaje de prestigio a todos los visitantes y profesionales.

Esa primera cita, celebrada del 3 al 5 de febrero, reunió de viernes a domingo a 7.000 visitantes, mientras que la segunda, celebrada 16 al 18 de marzo de 2018 concitó el beneplácito de 14.000 aficionados, cifra que fue ampliamente superada por la tercera, del 22 al 24 de marzo de 2019.

Alrededor del mundo, millones de personas (solo Angoulême, en Francia, reúne a más de 500.000 visitantes) esperaban la llegada de sus salones preferidos, aunque cada una lo hiciera a partir de sus propios intereses. Sin embargo, existía un común denominador en todos ellos: visitar durante horas un certamen de forma que se creaban flujos de personas que solo se podrían gestionar a través de la Informática. Empezó con la creación de páginas web cada vez más complejas y con

la, más tarde incorporada, venta de entradas online. Este tipo de venta constituía una herramienta no solo destinada a agilizar accesos multitudinarios, sino como indicador de la aceptación que el salón o el festival alcanzaría, reafirmando así la necesidad de la gestión de los flujos mencionados mediante la informática.

Tal y como nos hemos referido anteriormente con respecto a los miles de visitantes involucrados en un evento de dichas características que desean optimizar su tiempo, de inmediato surge otro argumento: ¿Dicho asesoramiento, no llevaría a la organización a conocer, de forma individualizada, los intereses y, por ende, el perfil de cada sujeto? Y ahí es donde la informática puede convertirse en un arma de vital importancia. No solo para el evento en curso, sino, a partir de él, en instrumento de proyección para los venideros. Una herramienta de incalculable valor para que los creadores de contenido de dichos eventos los ajusten a los gustos del asistente al salón.

Así, un tema en auge despierta especial interés: la Inteligencia Artificial. La inteligencia artificial es una herramienta de gran poder que permite realizar recomendaciones particulares a una persona sobre un producto (o elemento) basándose en las características de la persona y el producto. Amazon o Facebook son empresas que facturan miles de millones al año[1], y se valen de la IA y, más concretamente, de los Sistemas de Recomendación [1] para sugerir a sus usuarios compras que deberían realizar. Eso que muchos usuarios creen que ocurre por arte de magia, cuando su navegador les recomienda ciertos productos, no deja de ser un trabajo exhaustivo de recogida de datos por parte de estas empresas para entrenar modelos que generan predicciones de productos en los que los clientes pueden estar interesados. En este proyecto, se plantea utilizar estas mismas técnicas para dar respuesta a otro tipo de demanda. Los organizadores de los salones de cómic podrían ser capaces de predecir las preferencias de sus asistentes para ofrecerles un mejor asesoramiento en cuanto a asistencia a los eventos y contenidos que se organizan como parte del salón.

1.2 Descripción del proyecto

El propósito de este proyecto es construir un primer prototipo de un sistema inteligente que recomiende, a un visitante de un salón de cómic, un programa, para los días en los que se lleve a cabo el evento, en base a sus preferencias. Además, la idea es tener en cuenta las preferencias de asistentes con perfiles similares al protagonista para proponerle contenidos que quizás, en un principio, no serían necesariamente de su agrado pero que finalmente puedan gustarle.

En este proyecto, se utilizarán técnicas de *Machine Learning* para la predicción de preferencias y de optimización heurística para la optimización de un programa/*scheduling* en base a las preferencias predichas. Esto es necesario ya que sería desafortunado que los asistentes recibieran recomendaciones dispares a sus gustos y decidieran, en base a eso, no volver al salón.

En base al proyecto presentado y como resultado del mismo, en un futuro se

[1] Amazon triplica su beneficio en 2018:
https://elpais.com/economia/2019/02/01/actualidad/1549040550_020872.html

considerará mejorar el prototipo que se plantea aquí y desarrollar una aplicación para teléfono móvil que implemente este sistema inteligente para poder ser utilizada en diferentes ediciones del salón de cómic de Donostia. Además, se abriría la posibilidad de crear una aplicación genérica para cualquier salón de cómic.

2 Planteamiento inicial

En este capítulo se describen, por un lado, los objetivos y la estructura del proyecto y, por otro, se definen el alcance, la gestión, la planificación temporal, las herramientas, la gestión de riesgos y una evaluación económica relacionada con los gastos que conlleva la realización del proyecto.

2.1 Objetivos del proyecto

En este proyecto se plantean dos objetivos principalmente:

- Diseñar un sistema que solicite al asistente datos personales y sus preferencias en cuando a las temáticas que se tratan en los distintos contenidos de un salón del cómic.
- Aplicar técnicas de *Machine Learning* sobre la información recogida para ofrecer al asistente una recomendación. Esta recomendación consistirá en un programa o *scheduling* de actividades para los días del evento, con diferentes actividades que podrían ser del interés del asistente. Para generar dicha recomendación se tendrán en cuenta los contenidos que sean continuos, puntuales o repetidos a lo largo del evento.

2.2 Estructura del proyecto

Dada la naturaleza del proyecto, y los objetivos planteados, hemos identificado tres etapas en las que dividir el trabajo: (1) Preparación del conjunto de datos, (2) Predicción de preferencias y (3) Optimización del *scheduling*. Estas etapas, a su vez, tienen una serie de tareas principales que se presentan en este apartado. A continuación, en la figura 1, se presenta el diagrama de flujo del proyecto donde se diferencian las etapas, con sus respectivas tareas, y así facilitar su identificación a lo largo de esta memoria.

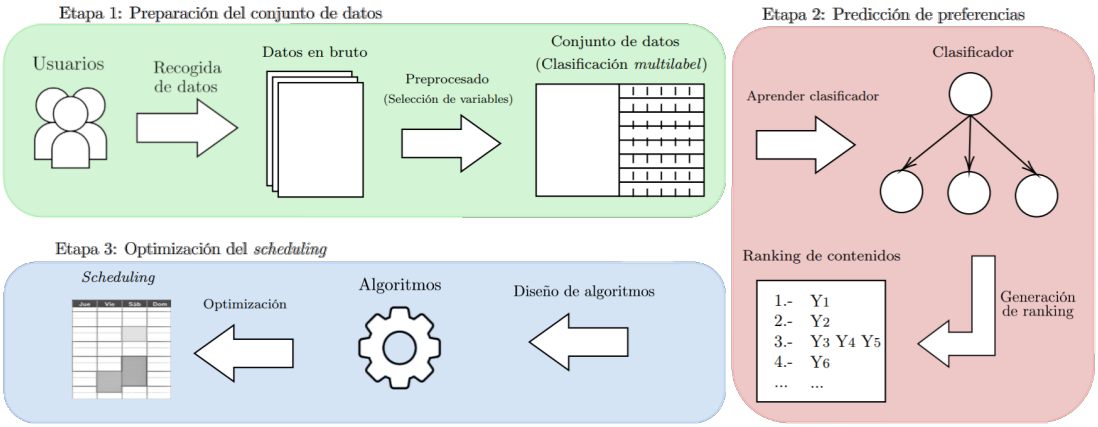


Figura 1: Diagrama de flujo del proyecto.

La **etapa 1** del proyecto, *"Preparación del conjunto de datos"*, agrupa las tareas referentes a la recogida y construcción del conjunto de datos que se va a

utilizar a lo largo de todo el proyecto. Las tareas principales de la etapa son la recogida de datos y su preprocesado. En la primera, la recogida de datos, se obtendrán los datos de asistentes reales y se limpiarán para que tengan el formato que se va a necesitar para poder trabajar con ellos. Por otro lado, conviene destacar que dada la naturaleza de los datos (como veremos más adelante) vamos a trabajar en un contexto de datos *multilabel* [2]. Esto es, además de tener un número importante de variables predictoras, el número de variables clase (los distintos contenidos a los que un asistente al salón podría acudir) es elevado. Con el fin de poder trabajar de manera eficiente con este conjunto de datos, se realizará una selección de variables, parte fundamental de la segunda tarea principal de esta etapa. Tras este proceso se obtendrá el conjunto de datos con el que se trabajará durante el resto del proyecto.

En la **etapa 2**, *"Predicción de preferencias"*, se decide trabajar con un algoritmo probabilístico, que permita, dadas las características del asistente y sus preferencias temáticas (variables predictoras), calcular la probabilidad de que le guste cada uno de los contenidos que se ofertan en el salón (las variables clase). Para poder realizar esta tarea, se ha elegido el algoritmo *Naïve Bayes*. Es importante destacar que, a pesar de que dicho algoritmo es un clasificador *baseline* en la minería de datos, en este proyecto no se realiza clasificación, sino que se utilizan las probabilidades computadas por *Naïve Bayes* para después proponer un *scheduling* de actividades al asistente. Por lo tanto, la primera tarea de esta esta etapa consiste en implementar un *Naïve Bayes* que, dadas las preferencias del asistente al salón, obtenga las probabilidades de que le guste cada uno de los contenidos que se ofrecen en el salón. La segunda tarea de la etapa consistirá en construir un ranking de contenidos en base a las probabilidades calculadas (de mayor a menor).

Por último, en la **etapa 3**, *"Optimización del scheduling"*, se parte del ranking previamente generado y el primer paso será formalizar matemáticamente el problema de optimización, y después, diseñar los algoritmos que se van a aplicar para construir el *scheduling*. En este proyecto se ha decidido implementar dos algoritmos de optimización: un algoritmo constructivo heurístico y un algoritmo genético. El primero aporta soluciones mediante un procedimiento que va incorporando elementos a cierta estructura, inicialmente vacía, que representa la solución. Para este problema en concreto se contará con una estructura vacía, que simulará un programa de actividades sin rellenar, que irá tomando forma a medida que se le añadan contenidos. El segundo, es un método adaptativo que puede usarse para resolver problemas de búsqueda y optimización. Trabaja con una población de individuos, y cada uno de ellos representa una solución factible a un problema dado.[2].

Una vez que se hayan realizado las tres etapas, se llevará a cabo una experimentación para poder evaluar la calidad del prototipo planteado en base al tipo de *schedulings* que ofrece.

[2]http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/temageneticos.pdf

2.3 Gestión del proyecto

Desde el primer momento del proyecto se optó por una metodología de trabajo ágil, Scrum, que permite ir progresando en la obtención de los resultados del proyecto de una forma incremental. Por ello se definieron 6 hitos a lo largo del tiempo estimado para dicho proyecto con la estimación de las tareas en horas en cada uno de los hitos. A continuación, se enumeran dichos hitos y se describen a grandes rasgos las tareas planificadas.

Hito 1: 14 de febrero

Para este primer hito se planificaron 25 horas en total. Se asignaron 8 horas de tareas de gestión, principalmente para definir el objetivo y las tareas del proyecto. Además, se asignaron 7 horas a formación, con objeto de aprender bases de datos y LaTeX, y 10 a documentación, para definir la estructura del documento y comenzar a redactar la memoria.

En este hito se emplearon un total de 25 horas, distribuidos en 8,5 en gestión, 2 en formación y 12,5 en documentación.

Hito 2: 7 de marzo

Para el segundo hito se planificaron 46 horas en total. Se asignaron 4 horas a tareas de gestión, principalmente para planificar las tareas del proyecto y realizar un seguimiento del mismo y 14 horas a formación, con objeto de aprender bases de datos y sistemas de recomendación. Por otro lado, se asignaron 5 horas a análisis, con el fin de definir las funcionalidades y seleccionar herramientas a emplear en el proyecto, y 13 horas a diseño, para diseñar la base de datos y comenzar con el diseño del sistema inteligente. Por último, se asignaron 5 horas a experimentación, para empezar a desarrollar un plan de experimentación para el proyecto, y 5 horas a documentación, para continuar redactando la memoria.

En este hito se emplearon un total de 31 horas, distribuidos en 4 en gestión, 14,5 en formación, 4 en análisis, 4 en diseño, y 4,5 en implementación.

Hito 3: 4 de abril

Para el tercer hito se planificaron 63 horas en total. Se asignaron 3 horas a tareas de gestión, principalmente para planificar las tareas del proyecto y realizar un seguimiento del mismo y 8 horas a formación, con objeto de aprender sistemas de recomendación. Por otro lado, se asignaron 5 horas a análisis, con el fin de revisar las funcionalidades definidas y las herramientas elegidas, y 7 horas a diseño, para revisar el diseño del sistema inteligente. Por último, se asignaron 25 horas a implementación, para crear la base de datos, 10 horas a experimentación, para revisar el plan de experimentación del proyecto, y 5 horas a documentación, para continuar redactando la memoria.

En este hito se emplearon un total de 38 horas, distribuidos en 4,5 en gestión, 11,5 en formación, 5 en análisis, 8 en diseño, 4 en experimentación y 5 en documentación.

Hito 4: 2 de mayo

Para el cuarto hito se planificaron 69 horas en total. Se asignaron 2 horas a tareas de gestión, principalmente para realizar un seguimiento del proyecto y 8 horas a formación, con objeto de aprender sistemas de recomendación. Por otro lado, se asignaron 4 horas a análisis, con el fin de revisar las funcionalidades definidas y las herramientas elegidas y 30 horas a implementación, para implementar el sistema inteligente. Por último, se asignaron 20 horas a experimentación, para realizar pruebas, y 5 horas a documentación, para continuar redactando la memoria.

En este hito se emplearon un total de 69 horas, distribuidos en 4 en gestión, 10 en formación, 4 en análisis, 32,5 en implementación, 2 en experimentación y 16,5 en documentación.

Hito 5: 4 de junio

Para el quinto hito se planificaron 77 horas en total. Se asignó 1 hora a gestión para realizar un seguimiento del proyecto, 5 horas a formación, con objeto de aprender sistemas de recomendación, y 1 hora a análisis, con el fin de revisar las funcionalidades definidas. Por otro lado, se asignaron 45 horas a implementación, para implementar el sistema inteligente, 15 horas a experimentación, para realizar pruebas, y 10 horas a documentación, para continuar redactando la memoria.

En este hito se emplearon un total de 83 horas, distribuidos en 4,5 en gestión, 7 en formación, 1 en análisis, 0,5 en diseño, 43,5 en implementación, 12,5 en experimentación y 14 en documentación.

Hito 6: 20 de junio

Para el último hito se planificaron 45 horas en total. Se asignaron 2 horas a tareas de gestión, principalmente para realizar un seguimiento del proyecto y 3 horas a formación, con objeto de aprender LaTeX. Por último, se asignaron 40 horas a documentación, para terminar de redactar la memoria y preparar la defensa del trabajo de fin de grado.

En este hito se emplearon un total de 90 horas, distribuidos en 3,5 en gestión, 6 en formación, 1 en diseño, 8 en implementación, 16 en experimentación y 55,5 en documentación.

Para representar estos hitos en el tiempo se utiliza el Diagrama de Gantt mostrado en la figura 2:

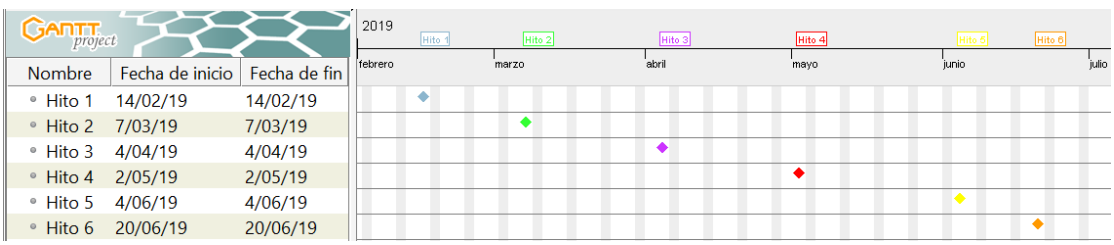


Figura 2: Diagrama de Gantt del proyecto.

2.4 Estimación vs. inversión temporal real

Durante la planificación del proyecto se estimaron cuántas horas duraría el proyecto y el número de semanas serían necesarias para acabarlo. En total se estimaron 19 semanas de trabajo, excluyendo fines de semana, con una carga de trabajo de algo más de 17 horas semanales. Es decir, 327 horas en total. Finalmente, se han invertido un total de 336 horas. En los hitos 2 y 3 se han invertido menos horas de las estimadas en un principio a causa de la atención que la organización y la ultimación de detalles que el salón de cómic requería en ese momento. No obstante, en los hitos 5 y 6 se ha decidido dedicarle más esfuerzos al proyecto para que el resultado final del trabajo sea de mayor interés, por lo que el recuento de horas finalmente invertidas ha quedado similar al estimado. Además, se estima que se dedicarán un total de 20 horas para la preparación de la defensa, tras la entrega de la memoria.

Tabla 1: Comparación de estimación e inversión temporal del proyecto.

Tareas	**Horas estimadas**	**Horas invertidas**
Hito 1	25	25
Hito 2	46	31
Hito 3	63	38
Hito 4	69	69
Hito 5	77	83
Hito 6	45	110
Total	**327**	**356**

A continuación, en la figura 3 se muestran estos datos en una gráfica para visualizarlos de una forma más clara:

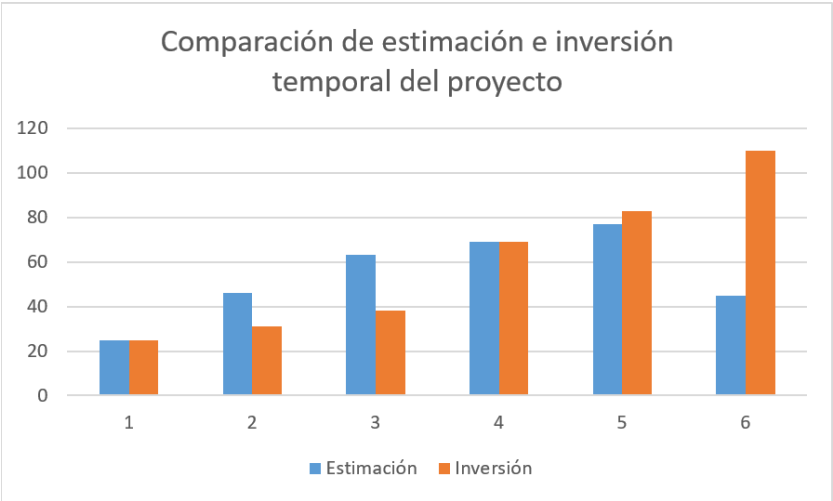


Figura 3: Comparación de estimación e inversión temporal del proyecto.

2.5 Herramientas

La aproximación inicial de las herramientas a utilizar a lo largo del proyecto es:

- LaTeX: Sistema de composición de textos, orientado a la creación de documentos escritos que presenten una alta calidad tipográfica. Utilizado, normalmente, para escribir artículos y libros científicos que incluyen, por ejemplo, expresiones matemáticas.

- Git: Software de control de versiones. Al tratarse de un almacenamiento en la nube, permite la recuperación de datos locales en caso de pérdida.

- Gantt Project: Programa que permite representar gráficamente la planificación temporal del proyecto. En él se añaden las distintas tareas previstas con sus fechas de inicio y duraciones y el propio programa crea la representación gráfica.

- Google Drive: Herramienta que permite almacenar y compartir archivos en Internet. Dado que es un servicio en la nube, los usuarios pueden acceder a sus archivos desde cualquier ordenador, dispositivos móviles o tablet identificándose mediante sus credenciales.

- Dropbox: Servicio de alojamiento de archivos multiplataforma en la nube. Permite a los usuarios almacenar y sincronizar archivos en línea y entre ordenadores, dispositivos móviles o tablets, y compartir archivos y carpetas con otros usuarios.

- Mathcha: Herramienta de diseño de diagramas y ecuaciones matemáticas. Permite exportar el trabajo en forma de imagen o código LaTeX.

- Microsoft Excel: programa de hojas de cálculo. Una herramienta de análisis y visualización de datos.

- Weka y Meka: Weka es una colección de algoritmos de *Machine Learning* utilizada para tareas de minería de datos. Contiene herramientas para la preparación, clasificación, regresión, *clustering*, asociación de reglas de minería y visualización de datos. Meka, por su parte, es una extensión de Weka que permite trabajar con conjuntos de datos *multilabel* [3].

2.6 Gestión de riesgos

El desarrollo de un proyecto conlleva una serie de riesgos que deben ser identificados con antelación para poder evitarlos o, en caso de que ocurran, poder actuar con rapidez para solventar cualquier problema. A continuación se describen los riesgos identificados:

Planificación incorrecta

- Descripción: Fallo en la estimación temporal, si alguna tarea requiere más tiempo del pensado inicialmente.
- Probabilidad: Media.
- Impacto: Medio.
- Consecuencias: Retraso en la entrega final del proyecto.
- Prevención: Detallar todas las tareas que se deberán realizar con el fin de que las estimaciones sean lo más ajustadas posibles, y si quedan dudas consultar con los tutores los tiempos que se suelen necesitar para completar algunas tareas.
- Plan de contingencia: Nueva estimación temporal dentro de ese marco temporal.

Problemas al desarrollar el sistema

- Descripción: Fallos en el diseño, en su desarrollo, falta de conocimiento.
- Probabilidad: Media.
- Impacto: Alto.
- Consecuencias: Retraso en la implementación y, por lo tanto, en la terminación del proyecto.
- Prevención: Buen análisis y diseño del sistema a desarrollar, con el fin de que luego sea más sencillo implementarlo.
- Plan de contingencia: Dedicación de más horas a la implementación y a las tareas que dependen de ella para no retrasar la fecha de terminación del proyecto.

Problemas de índole personal

- Descripción: Enfermedad común, lesión o accidente.
- Probabilidad: Baja.
- Impacto: Depende de la gravedad de la enfermedad/lesión. Un resfriado no afecta en absoluto al proyecto, pero un accidente de coche incapacitaría físicamente para continuar.
- Consecuencias: Inhabilidad para invertir tiempo en el proyecto.
- Prevención: Cuidado de la salud.
- Plan de contingencia: En caso de incapacidad para continuar, detener toda actividad hasta la recuperación.

Fallos en el sistema

- Descripción: Fallos en el equipo utilizado para trabajar.
- Probabilidad: Media.
- Impacto: Media.
- Consecuencias: Inhabilidad para seguir trabajando con ese equipo, al menos hasta su reparación.
- Prevención: Guardado de copias de seguridad de últimas versiones para evitar la pérdida de datos.
- Plan de contingencia: Uso de otro equipo hasta la reparación del original, mientras se sigue trabajando con la última versión guardada.

2.7 Evaluación económica

En esta sección se realiza un cálculo del coste total del proyecto teniendo en cuenta tanto la mano de obra, como los costes ocasionados por la adquisición o uso de software, hardware u otros gastos derivados del proyecto.

- Mano de obra: Teniendo en cuenta que el cálculo hipotético de la duración total de este proyecto es de 327 horas y que la hora de un analista de datos se cobra a 30 euros/hora, el **coste total de personal** asciende a **9.810 euros**.
- Software:
  - Licencia Windows 10, incluida en el precio del portátil.
  - Latex: licencia gratuita.
  - Git: uso de versión gratuita.
  - Gantt Project: licencia gratuita.
  - Google Drive: herramienta gratuita.
  - Dropbox: uso de plan gratuito.
  - Cacoo: versión de prueba gratuita.

  Gastos totales software: 0 euros.
- Hardware:
  - Ordenador portátil: valorado en 780 euros, con una vida media de 5 años y un uso de un 75% a lo largo del proyecto. Amortización anual: 780/5 = 156 euros. Gastos del portátil: ((156 x 19 semanas) x 0.75) / 52 semanas = 42,75 euros.

  Gastos totales hardware: 42,75 euros.

- Otros:
  - Libro "Fundamentos de los sistemas de ayuda a la decisión": **24 euros**.
  - Gastos indirectos: **493 euros**.

  Gastos totales otros: 517 euros.

Por lo tanto, la estimación de **coste total del proyecto** asciende a **10.369,75 euros**.

3 Gestión de datos

En este apartado se describe la gestión de datos que se ha realizado para la consecución del proyecto. Por un lado, se presenta la base de datos que va a almacenar los datos que se recojan. Por otro lado, se describe tanto el formato de los ficheros que van a almacenar la información exportada de la base de datos como el de los que va a recibir y gestionar el sistema inteligente.

3.1 Base de datos

Se decide hacer uso de una base de datos porque se considera la mejor opción para relacionar toda la información acerca de asistentes, contenidos del evento que posteriormente se va a manipular. El proyecto contará con una única base de datos relacional SQL, almacenada en un servidor creado con XAMPP, que estará formada por tablas que recogerán toda la información que se necesite gestionar.

Después de realizar un análisis general de los contenidos que se plantean en un salón del cómic[3] y de las características que pueden ser de interés recabar de los asistentes, se ha diseñado una base de datos de 22 tablas. Además, este análisis se ha realizado pensando en la ampliación del proyecto en el futuro. En el Anexo 1 se muestran las tablas, y las dependencias que habrá entre ellas, que recogerá la base de datos.

Las tablas almacenan los datos de los asistentes que van a utilizar el sistema, los de los salones y sus contenidos, los de las temáticas y subtemáticas que hay, los artistas que acuden al evento y las asociaciones que participan en el mismo. Además, existen una serie de tablas intermedias que se crean a fin de establecer conexiones entre tablas. Por ejemplo, la tabla 'contsubtem' recoge el título de un contenido, el nombre y la edición del salón al que pertenece, su temática y su subtemática[4], y la tabla 'usugustatem' contiene el email del asistente y el nombre de una temática que le gusta.

Cabe destacar que el elevado nivel de complejidad de la base de datos se justifica por la flexibilidad que proporciona a la gestión de datos y proporciona el sustento para poder realizar futuras extensiones o mejoras al sistema inteligente.

3.2 Fichero de datos ARFF

La piedra angular de este proyecto es el sistema inteligente. Por lo tanto, es importante, para realizar la gestión del mismo, conocer el formato de los datos con los que se va a trabajar a lo largo del proceso.

Para este problema se ha decidido que la extensión de fichero que el sistema va a recibir es ARFF (*Attribute-Relation File Format*), un formato de archivo que se puede importar desde la herramienta Weka. Los datos, antes de que los reciba el

[3]Se ha tomado el salón organizado en el Kursaal del 22 al 24 de marzo de 2019 a modo de inspiración.

[4]El contenido "Charla con David Benzal", por ejemplo, es una actividad de temática "Cómic" y subtemática "Ilustración", mientras que "Rol en vivo" es un contenido de temática "Juego" y subtemática "Rol".

sistema, se almacenan, primero, en la base de datos para, después, ser exportados y manejados. Esto es, se vuelcan en un fichero CSV, que contendrá estos datos y después será convertido a ARFF y, así, facilitar la tarea.

Se decide hacer uso de un fichero de extensión CSV para la transición indicada. Por lo tanto, toda la información que se va a manipular se va a gestionar desde un mismo fichero de este tipo. A diferencia de otros proyectos, la previsión realizada con respecto al tamaño de los datos que se van a manejar en este es de conjuntos de datos que ocupan KBs. Por lo tanto, no se prevé tener que dividir la información a gestionar en varios ficheros distintos. No obstante, si en un futuro el proyecto se ampliara y su uso se extendiera a diversos salones que acogieran, a su vez, a grandes multitudes, esta previsión podría variar.

4 Etapa 1: Preparación del conjunto de datos

En este capítulo se describe la **etapa 1** que se ilustra en la figura 1. El objetivo de esta etapa es, partiendo de la recogida datos en bruto, construir el conjunto de datos con el que el sistema trabaja más adelante. Para conseguir que el formato final del conjunto de datos sea el deseado, deben seguirse una serie de pasos que se explican a continuación.

4.1 Recogida de datos

La recogida de datos es un paso absolutamente esencial en el desarrollo del sistema. Tanto es así que sin datos que manejar no se podrían cumplir los objetivos planteados.

Pese a que en la mayoría de proyectos se parte de un conjunto de datos ya existente, en este caso se decidió trabajar con datos reales. Estos datos, además, serán los que se almacenarán en la base de datos del proyecto (descrito en el punto 3) para su posterior manejo y gestión.

Por todo ello, se generó una encuesta de satisfacción (ver Anexo 2) para el público que visitara la *III Edición del Salón Internacional de Cómic y Manga de San Sebastián*, que tuvo lugar los días 22, 23 y 24 de marzo de 2019. El objetivo de dicha encuesta era conocer los contenidos por los que el visitante tenía interés para, en base a ellos, ser capaz de generar predicciones acertadas para futuros asistentes. La encuesta contaba con tres módulos de preguntas a responder por el asistente: el primero constaba de preguntas acerca de datos personales del asistente, tales como lugar de origen, género o nivel de estudios. El segundo, por otro lado, pretendía conocer los gustos del asistente en cuanto a temáticas o subtemáticas de contenidos del evento (cómic, ilustración, vestuario de cine), y su nivel de satisfacción respecto al mismo evento. El tercero, y último, buscaba obtener información acerca de la experiencia del asistente en el evento para ofrecer mejores contenidos en posibles futuras ediciones.

La encuesta en cuestión se diseñó con la herramienta Google Forms y se recogieron 125 respuestas de asistentes al evento, que se exportaron a una tabla de un documento Excel. Cada respuesta contaba con 16 campos de pregunta, aunque, como se explica más adelante, en muchos casos los asistentes dejaron preguntas sin responder.

Por otro lado, había que tener en cuenta que la base de datos recoge información de carácter personal acerca de los asistentes, como sus direcciones de correo electrónico. Sin embargo, siguiendo lo establecido en el Reglamento General de Protección de Datos (RGPD), no era posible solicitar al visitante del evento que proporcionara este dato en la encuesta que debía completar. Por lo tanto, y para no prescindir de esta información, se ha utilizado un generador de datos aleatorios online[5] para generar los datos requeridos para cada asistente (ver Anexo 3). Esta herramienta es capaz de generar un fichero con el número de campos deseados y completarlos con instancias del tipo elegido de entre los disponibles. En este caso,

[5] generatedata.com

se genera un fichero CSV que cuenta con un campo, para el correo electrónico, y tantas instancias de tipo *email* como asistentes han completado la encuesta.

4.2 Limpieza del conjunto de datos

Una vez recogidos los datos se procede a su limpieza. Esto es vital, ya que los datos no llegan en el formato necesario para trabajar con ellos y se deben preparar para que la base de datos los recoja correctamente.

Los datos con los que se cuenta se encuentran divididos en dos conjuntos de datos. El primero contiene las 125 respuestas recogidas en la encuesta, mientras que el segundo tiene los datos generados con las herramientas online previamente mencionadas.

Parte de los datos que se solicitan en la encuesta no son necesarios para el diseño del prototipo que se plantea en el trabajo. Sin embargo, aprovechando la iniciativa, se decidió pedir información a los asistentes que pudiera ser de interés en futuras mejoras del prototipo. A continuación se muestran los campos que en un principio tienen ambos conjuntos de datos (figura 4) y, después, los que se conservarán y los que se desecharán (figura 5).

Marca temporal	Lugar de origen	Género
Edad	Nivel de estudios	Días asistidos
Temas de interés	Subtemas de interés	¿Fuiste a todas las charlas...?
¿A cuáles no fuiste...?	¿A cuáles sí fuiste...?	¿Recomendarías...?
¿Cuál/es?	¿Es el primer año...?	Si no es la primera vez...
¿Qué has echado de menos...?	Última pregunta	

Email

Figura 4: Campos iniciales de los conjuntos de datos.

~~Marca temporal~~	Lugar de origen	Género
Edad	Nivel de estudios	~~Días asistidos~~
Temas de interés	Subtemas de interés	~~¿Fuiste a todas las charlas...?~~
¿A cuáles no fuiste...?	¿A cuáles sí fuiste...?	~~¿Recomendarías...?~~
~~¿Cuál/es?~~	~~¿Es el primer año...?~~	~~Si no es la primera vez...~~
~~¿Qué has echado de menos...?~~	~~Última pregunta~~	

Email

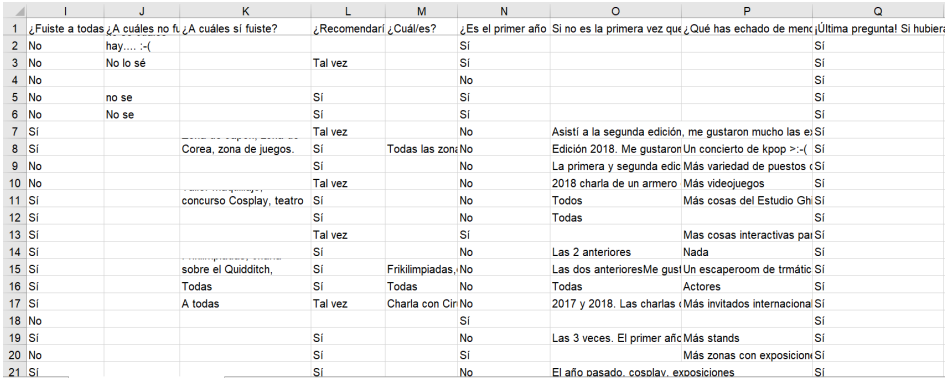
Figura 5: Campos a conservar o desechar de los conjuntos de datos.

Durante el proceso de limpieza se analiza qué datos de todos los recogidos merece la pena conservar y se llevan a cabo las modificaciones necesarias. En este caso, la tarea es examinar los dos conjuntos de datos que contienen la información y decidir qué datos deberán ser volcados a la base de datos y de qué manera. A continuación se explica el proceso que se va a seguir para completar esta tarea.

Antes de comenzar se muestran dos fragmentos (figuras 6 y 7) del conjunto de datos de respuestas de los asistentes de la tercera edición del Salón Internacional de Cómic y Manga de San Sebastián:

	A	B	C	D	E	F	G	H
1	Marca temporal	Lugar de origen	Género	Edad	Nivel de estudios	¿Qué días fuiste al evento	¿Cuáles de los siguientes t	¿Cuáles de los siguientes s
2	3/11/2019 17:15:03	Ondarroa	Hombre	33	Estudios superiores (Unive	Viernes (22 de marzo)	Cine, Cómic	Guion (Cine), Dibujo (Cómi
3	3/11/2019 19:39:53	Irura	Mujer	39	Estudios superiores (Unive	Viernes (22 de marzo)	Juegos	Juegos de mesa, Juegos d
4	3/12/2019 15:53:45	Donostia	Mujer	22	Soy estudiante	Viernes (22 de marzo), Sá	Cine, Cómic, Entretenimier	Actuación (Cine), Producci
5	3/12/2019 16:09:46	Irura	Mujer	39	Estudios superiores (Unive	Viernes (22 de marzo)	Entretenimiento	Me gustan todos los subte
6	3/12/2019 16:29:10	Irura	Mujer	39	Estudios superiores (Unive	Viernes (22 de marzo)	Entretenimiento	Juegos de mesa
7	3/22/2019 17:45:14	Gotas Delchev	Hombre	22	Soy estudiante	Viernes (22 de marzo), Sá	Cine, Cómic, Entretenimier	Actuación (Cine), Color (C
8	3/22/2019 17:45:23	Donostia	Hombre	21	Estudios superiores (Unive	Viernes (22 de marzo), Sá	Cine, Entretenimiento, Jue	Actuación (Cine), Vestuari
9	3/23/2019 16:01:02	Hernani	Hombre	36	Estudios superiores (Unive	Sábado (23 de marzo)	Cómic	Dibujo (Cómic), Guion (Cór
10	3/24/2019 13:40:05	Hondarribia	Hombre	24	Estudios secundarios (Bac	Sábado (23 de marzo), Do	Cine, Cómic, Juegos	Actuación (Cine), Producci
11	3/24/2019 19:00:34	Donosti	Mujer	47	Estudios secundarios (Bac	Domingo (24 de marzo)	Cine, Cómic, Entretenimier	Me gustan todos los subte
12	3/24/2019 21:13:21	San Sebastián	Mujer	23	Estudios superiores (Unive	Viernes (22 de marzo), Sá	Cine, Cómic, Juegos, Tele	Vestuario (Cine), Dibujo (C
13	3/24/2019 21:13:40	Donosti	Mujer	28	Estudios superiores (Unive	Viernes (22 de marzo), Sá	Entretenimiento, Juegos	Dibujo (Cómic), Ilustración
14	3/24/2019 21:14:16	Donostia	Hombre	29	Estudios superiores (Unive	Viernes (22 de marzo), Sá	Cine, Cómic, Entretenimier	Actuación (Cine), Dirección
15	3/24/2019 21:29:15	Donostia	Mujer	41	Estudios superiores (Unive	Viernes (22 de marzo), Sá	Cine, Cómic, Televisión	Actuación (Cine), Dibujo (C
16	3/24/2019 22:37:33	Irún	Mujer	25	Estudios superiores (Unive	Viernes (22 de marzo), Sá	Cine, Cómic, Entretenimier	Actuación (Cine), Vestuari
17	3/25/2019 8:49:33	San Sebastián	Hombre	55	Estudios superiores (Unive	Viernes (22 de marzo), Sá	Cine, Cómic, Entretenimier	Actuación (Cine), Color (C
18	3/27/2019 19:07:02	San Sebastián	Hombre	42	Estudios secundarios (Bac	Domingo (24 de marzo)	Cine, Entretenimiento, Tele	Videojuegos
19	3/27/2019 19:07:52	Renteria	Mujer	27	Prefiero no decirlo	Sábado (23 de marzo)	Cine, Cómic, Entretenimier	Me gustan todos los subte
20	3/27/2019 19:08:20	Catalunya	Hombre	49	Estudios superiores (Unive	Sábado (23 de marzo)	Cine, Cómic, Entretenimier	Ilustración (Cómic), Proyec
21	3/27/2019 19:09:22	Donostia	Mujer	47	Estudios superiores (Unive	Sábado (23 de marzo)	Cine, Cómic, Entretenimier	Actuación (Cine), Vestuari

Figura 6: Primer fragmento de respuestas de los asistentes del evento.



	I	J	K	L	M	N	O	P	Q
1	¿Fuiste a todas	¿A cuáles no fu	¿A cuáles sí fuiste?	¿Recomendarí	¿Cuál/es?	¿Es el primer año	Si no es la primera vez que	¿Qué has echado de menc	¡Última pregunta! Si hubier
2	No	hay.... :-(				Sí			Sí
3	No	No lo sé		Tal vez		Sí			Sí
4	No					No			Sí
5	No	no se		Sí		Sí			Sí
6	No	No se		Sí		Sí			Sí
7	Sí			Tal vez		No	Asistí a la segunda edición,	me gustaron mucho las e	Sí
8	Sí		Corea, zona de juegos.	Sí	Todas las zona	No	Edición 2018. Me gustaron	Un concierto de kpop >:-(	Sí
9	No			Sí		No	La primera y segunda edic	Más variedad de puestos (	Sí
10	No			Tal vez		No	2018 charla de un armero	Más videojuegos	Sí
11	Sí		concurso Cosplay, teatro	Sí		No	Todos	Más cosas del Estudio Gh	Sí
12	Sí			Sí		No	Todas		Sí
13	Sí			Tal vez		Sí		Mas cosas interactivas par	Sí
14	Sí			Sí		No	Las 2 anteriores	Nada	Sí
15	Sí		sobre el Quidditch,	Sí	Frikilimpiadas,	No	Las dos anterioresMe gust	Un escaperoom de trmátic	Sí
16	Sí		Todas	Sí	Todas	No	Todas	Actores	Sí
17	Sí		A todas	Tal vez	Charla con Cir	No	2017 y 2018. Las charlas (	Más invitados internacional	Sí
18	No					Sí			Sí
19	Sí			Sí		No	Las 3 veces. El primer año	Más stands	Sí
20	No			Sí		Sí		Más zonas con exposicion	Sí
21	Sí			Sí		No	El año pasado, cosplay,	exposiciones	Sí

Figura 7: Segundo fragmento de respuestas de los asistentes del evento.

Para empezar, el conjunto de datos que recoge las respuestas de la encuesta cuenta con una serie de campos que carecen de relevancia para el proyecto, como, por ejemplo, aquellas que preguntan al asistente por los días que asistió al evento o si era el primer año que lo visitaba. Por lo tanto, solo se conservarán aquellas que cuentan con la edad, el género, el lugar de origen del asistente y su nivel de estudios, indican qué temáticas y subtemáticas le gustan, o hacen referencia a los contenidos que se visitaron (o no) durante el evento y si se recomendarían. Además, la primera instancia, que contiene las preguntas de la encuesta, también es prescindible.

Por último, 7 de las 125 respuestas indican que no se recomendarían los contenidos que se visitaron. Se decide que el modelo se va a entrenar únicamente con los casos en los que el asistente sí recomendaría los contenidos visitados. Esto se debe a que, en el contexto de un salón de cómic, un asistente tiende a ir, únicamente, a los contenidos que sabe que le van a resultar de su agrado. Si un área no se encuentra entre sus intereses, no se va a sentir atraído por ella y no va a mostrar ningún interés por acudir a los contenidos de la misma. Por lo tanto, si aun yendo únicamente a lo que creía con seguridad que iba a resultar de su agrado no lo recomendaría, la organización del evento pierde la capacidad de guiarle; se convierte en un caso impredecible. Por todo esto, en este proyecto se decide no tener en cuenta las respuestas en las que el asistente indica que no recomendaría los contenidos del evento del conjunto de datos.

Es decir, que al principio había un conjunto de datos con 17 campos y 125 instancias y, tras la limpieza, uno de 10 campos y 118 instancias, reduciendo así la cantidad de datos a manejar hasta tener únicamente lo esencial. El conjunto de datos que contiene las direcciones de correo electrónico generados de manera aleatoria, sin embargo, se mantiene inalterado.

4.3 Preprocesado del conjunto de datos

Finalizada la limpieza, el siguiente paso consiste en el preprocesado de los datos. Los datos que sí se guardan se gestionan con la ayuda de un *script*. El objetivo, en este paso, es introducir los datos que se han conservado en la base de datos para, después, exportarlos a un nuevo fichero CSV (*Comma-Separated Values*) y

continuar con el proceso.

Los datos a gestionar contienen información sobre el asistente o sobre el evento. Los referentes al asistente, que después se volcarán a la base de datos, son: email, edad, género, nivel de estudios y ciudad de origen. El primer elemento se obtiene del conjunto de datos generados con la herramienta online previamente mencionada, mientras que los demás llegan del conjunto de datos de las respuestas de la encuesta. Aunque los datos de género no necesitan sufrir cambio alguno, los de edad se utilizarán para calcular el año de nacimiento del asistente y los de nivel de estudios se acortarán (se modificarán. Por ejemplo, "Estudios superiores (Universitarios / Formación Profesional Superior)" será "Estudios superiores", y así sucesivamente). Por otro lado, los de ciudad de origen se modificarán para que aunque varios asistentes hayan escrito el nombre de una ciudad o comunidad autónoma de forma distinta se almacenen con el mismo nombre. Por último, a partir del lugar de origen se asignará un código postal, intentando que sea lo más preciso posible. Algunos pueblos solo tendrán un código postal y se elegirá ese para el asistente, pero en aquellos casos en los que se haya escrito el nombre de una comunidad autónoma o país se escogerá un código postal de su capital. Teniendo la información de los dos conjuntos de datos se procede a su volcado en la base de datos.

No obstante, llegados a este punto sigue quedando la segunda parte de los datos: la asistencia a los contenidos (si/no) y el nivel de satisfacción (si se recomiendan). Puesto que las respuestas de texto que se reciben en las encuestas no siguen un formato estandarizado, deben ser sometidas a un proceso de adaptación para su posterior introducción en la base de datos.

Sin embargo, tampoco se puede utilizar un *script* para automatizar su modificación, puesto que cada asistente hace un uso distinto del lenguaje para plasmar sus respuestas. Por lo tanto, la conclusión a la que se llega es que hay que hacer una edición pormenorizada y manual de los datos. Por un lado, se entiende que si el visitante ha acudido a ciertos contenidos ha sido porque le interesaban, pero de la misma manera se intuye que si no ha podido ir a algunos y se ha quedado con las ganas es porque esos contenidos también le suscitaban interés.

Por desgracia, fueron muchos los visitantes que dejaron respuestas en blanco y no proporcionaron ningún tipo de información, así que se hace necesario llevar a cabo una investigación más concisa para eliminar estas incógnitas. Para ello, se presta atención a las temáticas y subtemáticas que indicara que le gustan, y a su respuesta a la pregunta "*Si no es la primera vez que vienes, ¿qué otras ediciones visitaste y cuáles fueron los contenidos que más te gustaron?*". Esto se debe a que esta respuesta también puede ofrecer pistas sobre qué le pudo haber gustado, aunque no lo dijera. Es una información muy valiosa que, en base a la experiencia previa que tengo en el dominio, soy capaz de extraer.

Con todo esto, ya se puede rellenar la base de datos indicando qué contenidos del evento le gustaron a un individuo y cuáles no, preparando el terreno para la siguiente tarea. Las tablas de la base de datos que se han utilizado han sido las siguientes (ver Figura 8): *usuario, salon, tematica, subtematica, contenido, usugustatem, cuando, usugustasub, acudeusucont, conttem* y *contsubtem*.

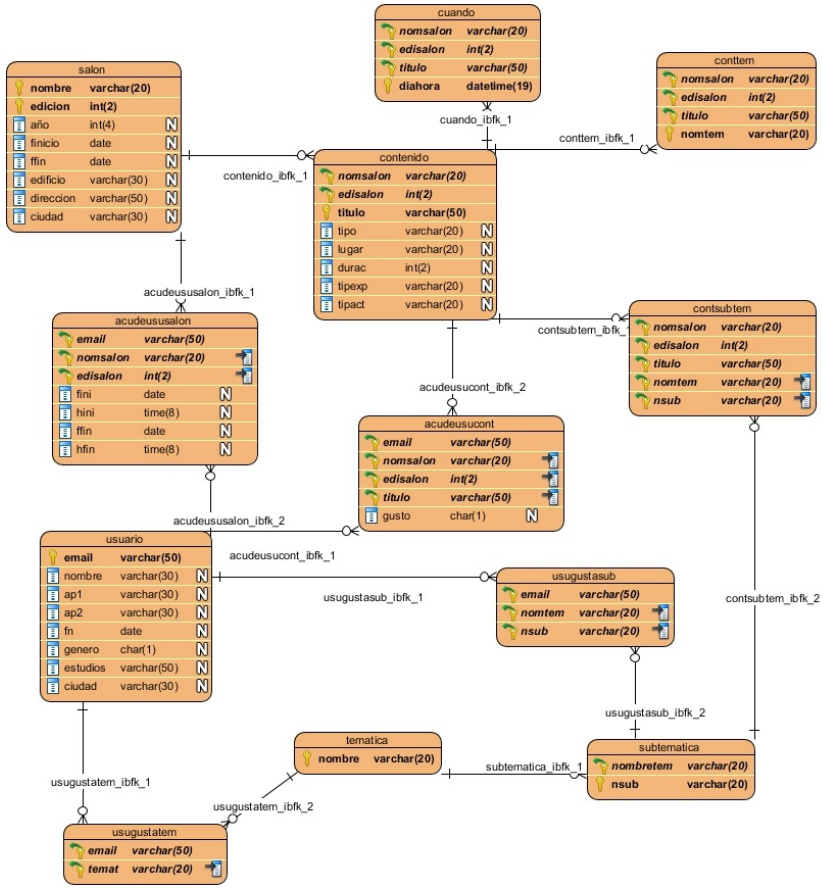


Figura 8: Tablas de la base de datos utilizadas para este proyecto.

4.4 Edición del conjunto de datos

Tal y como se menciona en el punto 3.2, la extensión del fichero que va a recibir el modelo va a ser ARFF. Este fichero será el producto de convertir el fichero CSV con los datos elegidos, exportados desde la base de datos, al formato deseado para facilitar la tarea a Weka y Meka. El fichero contendrá la siguiente información, en este orden:

- Datos del asistente: email, fecha de nacimiento, lugar de origen, código postal, género y nivel de estudios. Estos datos no serán binarios, sino que cada uno tendrá un valor para cada asistente. Cada uno ocupará un campo del fichero.

- Temáticas y subtemáticas: Cada subtemática existente en la base de datos tendrá asociada un campo en el conjunto de datos. Es decir, que en el conjunto de datos habrá tantos campos dedicadas a estos datos como subtemáticas haya en la base de datos. No obstante, puesto que cada subtemática S pertenece a una temática T, el formato que tendrá este apartado en el conjunto de datos será el siguiente: *T1-S1, T1-S2, T1-S3, T2-S1...* Por último,

los datos de estos campos serán binarios: un 1 si al asistente al que se hace referencia le gusta esa temática/subtemática y un 0 si no.

- Contenidos: Cada contenido que se haya registrado en la base de datos tendrá asociado un campo en el conjunto de datos. Los datos de estos campos serán binarios: un 1 si al asistente al que se hace referencia le gusta ese contenido y un 0 si no. En caso de desconocer si al asistente le ha gustado cierto contenido o no, se supondrá que no.

En conclusión, el conjunto de datos tendrá un formato como el de la figura 9:

Datos del usuario						**Temáticas y subtemáticas**				**Contenidos**			
Email	Fecha nacimiento	Lugar de origen	Código postal	Género	Nivel de estudios	T1-S1	T1-S2	...	Tn-Sn	C1	C2	...	Cn
Campos no binarios						Campos binarios				Campos binarios			

Figura 9: Formato del conjunto de datos que utilizaremos para aprender el sistema inteligente.

En este proyecto se contará con un total de 88 variables; las 27 primeras (6 de los datos personales de los asistentes y 21 de las subtemáticas con sus temáticas correspondientes) serán las variables predictoras y las 61 restantes (los contenidos del evento) las variables clase.

4.5 Selección de variables

Tras la recogida de datos y su posterior limpieza y preprocesado se decide realizar una selección de variables. La selección de variables es un proceso que consiste en la selección de un subconjunto de atributos relevantes (variables predictoras y variables clase) para su posterior uso en la construcción del modelo. Este proceso cuenta con numerosos objetivos [4], como pueden ser la reducción de "ruido" en un conjunto de datos o la aportación de modelos con un coste computacional reducido.

Se ha realizado una valoración de las diversas opciones que existen para llevar a cabo esta tarea y se ha decidido hacer la selección de variables aplicando un filtro en base al cálculo de informaciones mutuas. La información mutua mide la información que dos variables comparten, y su coste computacional es reducido. Es decir, mide en cuánto el conocimiento de una variable reduce la incertidumbre que puede existir sobre la otra. Si dos variables son independientes, entonces conocer la primera no da información sobre la segunda y viceversa, por lo que la información mutua entre ambas es 0. Así mismo, si las dos variables son iguales entonces toda información que proporciona la primera es compartida por la segunda. En resumen, la información mutua mide la cantidad de información transferida cuando la primera variable es transmitida y la segunda es recibida.

Para calcular la información mutua de una variable sobre otra se utiliza la siguiente fórmula, siendo x_i una variable e Y el vector de clases a predecir:

$$I(x_i, Y) = \log_2 \frac{P(x_i|Y)}{P(x_i)}$$

Sabiendo esto, se decide aplicar este proceso de cálculo en el proyecto para descubrir qué variables aportan más información al modelo que otras y, así, decidir qué variables va a conservar éste para trabajar más adelante.

Detalles de implementación

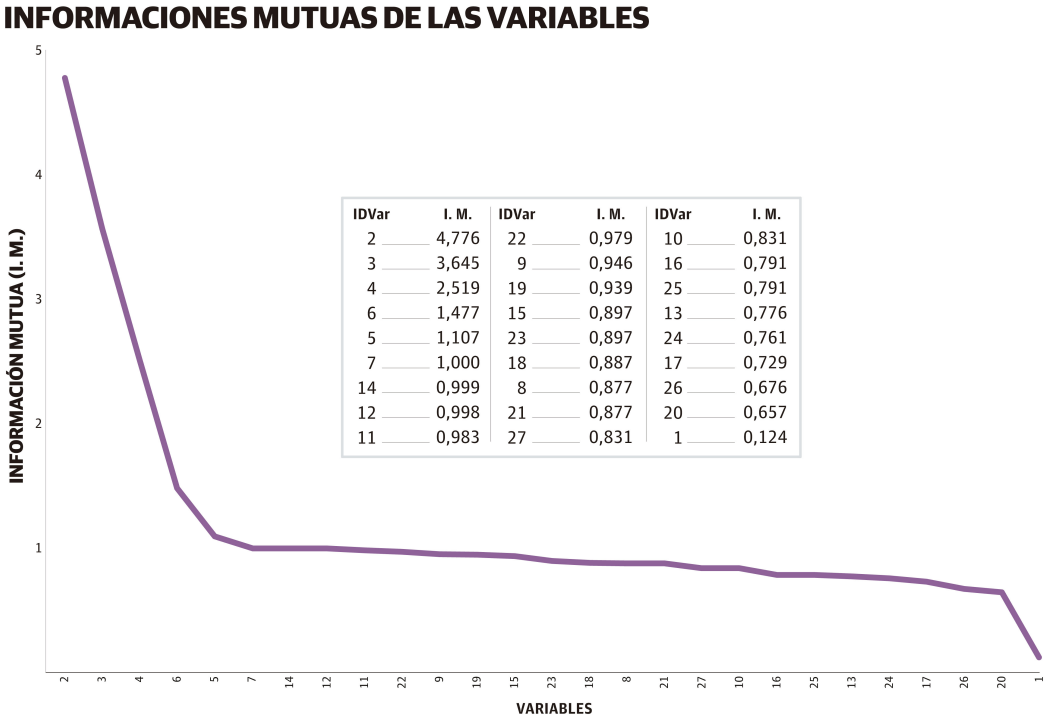
Tal y como se indica en el punto 2.2, el paradigma de clasificación del problema es *multilabel*. Para este problema los contenidos que tiene el evento en cuestión son atributos clase, y los datos de los asistentes y las subtemáticas de los distintos contenidos, junto con sus temáticas correspondientes, no. Para esta tarea se decide hacer uso de Meka, una extensión de Weka, que permite llevar a cabo la división que se describía en el punto mencionado entre atributos clase y el resto de atributos.

Por lo tanto, el primer paso es hacer la división con Meka, que se consigue haciendo uso del filtro del que dispone la herramienta para esta tarea: *MekaClassAttributes*. Indicando el rango de atributos que van a actuar como clase y, después, aplicando el filtro se obtiene el conjunto de datos con los atributos separados en las dos categorías deseadas y se procede a hacer la selección de variables.

Para este paso se utiliza un filtro de Weka con su correspondiente método de búsqueda, que genera un ranking con todos los atributos en orden descendente, desde los que aportan más información al modelo hasta los que aportan menos. En este caso se decide hacer uso de *InfoGain* como filtro y de *Ranker* como su método de búsqueda.

Resultados obtenidos

Para poder observar los resultados de una manera más visual, a continuación, en la figura 10, se muestran los resultados obtenidos junto a la gráfica que los representa:



IDVar	I. M.	IDVar	I. M.	IDVar	I. M.
2	4,776	22	0,979	10	0,831
3	3,645	9	0,946	16	0,791
4	2,519	19	0,939	25	0,791
6	1,477	15	0,897	13	0,776
5	1,107	23	0,897	24	0,761
7	1,000	18	0,887	17	0,729
14	0,999	8	0,877	26	0,676
12	0,998	21	0,877	20	0,657
11	0,983	27	0,831	1	0,124

Figura 10: Informaciones mutuas, obtenidas al aplicar el filtro *Ranker* en Meka a las variables predictoras del conjunto de datos, ordenadas de manera descendente.

Los resultados muestran unos primeros atributos con informaciones mutuas notablemente mayores que las demás. Después, se observa un descenso radical en estos valores, que eventualmente se reduce y se estabiliza. En este rango se encuentra la mayoría de atributos del conjunto de datos. Finalmente, los valores vuelven a descender de forma más significativa con los últimos atributos.

Por lo tanto, lo único que falta para continuar con la siguiente tarea es especificar cuál debe ser el valor de aportación mínima, al que se denominará α, que debe ofrecer un atributo para no ser descartado tras la selección.

5 Etapa 2: Predicción de preferencias

En esta sección se presenta la **etapa 2** de la figura 1. El objetivo es, conseguir un ranking de contenidos (de mayor a menor preferencia), partiendo del conjunto de datos reducido tras la realización de la selección de variables.

5.1 Descripción general

Antes de comenzar, para visualizar el proceso realizado hasta el momento y poder situarse en el punto actual del proyecto se presenta la figura 11:

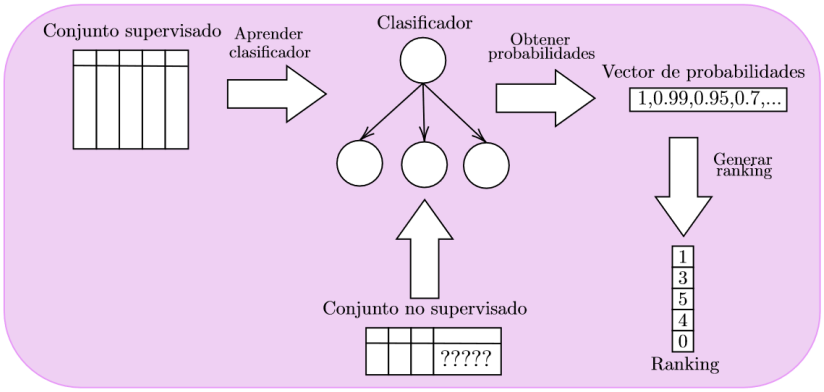


Figura 11: Proceso de generación de rankings de probabilidades. Se va a utilizar un conjunto de datos supervisado para el aprendizaje y construcción de un clasificador, al que después se le va a introducir un conjunto de datos no supervisado. Tras el cálculo de las probabilidades mediante las que se hacen las predicciones se obtendrá un vector con estas probabilidades que, finalmente, se insertarán en el ranking.

En este punto del proyecto, la siguiente tarea es aprender el clasificador que se va a utilizar para obtener las probabilidades de que a los casos de prueba del conjunto no supervisado les gusten los contenidos del salón. Por lo tanto, antes de proceder con la predicción de preferencias conviene revisar el conjunto de datos con el que se van a realizar las predicciones de preferencias. En la figura 12 se presenta el conjunto de datos que recibe el clasificador. No obstante, tras la selección de variables el conjunto de variables predictoras cuenta únicamente con las que no han sido descartadas en el proceso mencionado.

Variables predictoras										**Clases a predecir**			
Email	Fecha nacimiento	Lugar de origen	Código postal	Género	Nivel de estudios	T1-S1	T1-S2	...	Tn-Sn	C1	C2	...	Cn
Campos no binarios						Campos binarios				Campos binarios			

Figura 12: Descripción del conjunto de datos que recibe el clasificador.

Por un lado se cuenta con los atributos que actúan como variables predictoras, que son los datos del asistente y las subtemáticas que le gustan, con sus correspondientes temáticas, y, por otro, con las variables que actúan como clases a

predecir, que son todos los contenidos del evento. En ningún caso hay caracteres especiales en los campos de los datos, ya que podría haber inconsistencias en la base de datos, que no siempre los gestiona correctamente.

5.2 Predicción de preferencias mediante modelos probabilísticos

La tarea de predicción de preferencias de este proyecto es la siguiente: dados los datos de perfil de un asistente al salón (descritos en el punto 4.2) y referencias de temáticas y subtemáticas que le gustan, el objetivo es predecir la probabilidad de que le gusten cada uno de los contenidos del evento.

Tal y como se menciona en el punto 2.2, el problema con el que se trabaja en este proyecto es supervisado y, se decide hacer uso de un clasificador probabilístico. Esto es, que a partir de un conjunto de datos supervisado (denominado conjunto de entrenamiento) se intenta encontrar una función que permita clasificar ejemplos (o conjunto de prueba) que el sistema todavía no ha visto. Por lo tanto, para llevar a cabo la predicción de preferencias es necesario aprender los parámetros de un clasificador.

La estrategia que se decide adoptar es aprender un clasificador *Naïve Bayes*, o Bayesiano Ingenuo, un clasificador probabilístico fundamentado en el teorema de Bayes que se aplica en un entorno de aprendizaje supervisado. *Naïve Bayes* es una versión acotada de una Red Bayesiana, en la que todos los atributos son independientes dado el valor de la variable clase [5].

En la figura 13 se ve un ejemplo del grafo de probabilidades condicionadas que Naïve Bayes aprende (que induce una factorización particular de la probabilidad de cada variable clase). C corresponde a la variable clase, y las A_i denotan las variables predictoras. Es importante señalar que en el contexto de este proyecto, por cada variable clase (contenido del evento), se calculará un grafo como el de la figura 13.

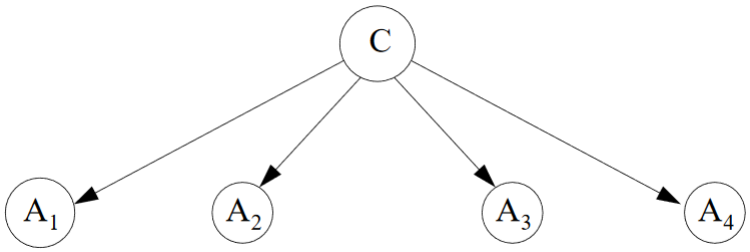


Figura 13: Grafo de probabilidades condicionadas representadas por el modelo probabilístico que aprende el algoritmo Naïve Bayes sobre un conjunto de cuatro variables predictoras y una variable clase.

Naïve Bayes realiza predicciones en base al cálculo de las probabilidades condicionadas. Sin embargo, en este proyecto no se buscan las predicciones, sino las probabilidades mediante las cuales se llega a dichas predicciones. Esto se debe

a que más adelante se pretende generar un ranking con los contenidos que van a resultar de mayor agrado al asistente. Por lo tanto, no basta con saber que un contenido le va a gustar. Conocer la probabilidad de que esto ocurra es lo que ayudará al modelo a recomendar los contenidos que mayor probabilidad tengan de gustarle.

Para calcular la probabilidad de obtener cierta clase c dado un caso x el clasificador utiliza la siguiente fórmula, siendo P(x|c) la probabilidad del atributo x dada la clase c, P(c) la probabilidad a priori de la clase c y P(x) la probabilidad a priori del atributo x:

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

El siguiente paso es dividir las instancias en conjunto de entrenamiento y conjunto de prueba y, por último, buscar las probabilidades de que a los casos de prueba le gusten cada uno de los contenidos del evento (este proceso se detalla en el punto 7, *Experimentación*).

5.3 Generación de rankings

Puesto que en la predicción de preferencias se obtienen las probabilidades necesarias para generar el ranking, la última tarea de esta etapa consta de insertar dichas probabilidades en un ranking. Por lo tanto, tras la obtención de las probabilidades se utiliza un algoritmo de ordenación para generar un ranking descendente de probabilidades y ver qué contenidos le deberían de gustar más a cada asistente de prueba y cuáles menos.

Para este problema se utiliza un algoritmo de ordenación por inserción simple, que consiste en ir insertando los elementos de una lista en su parte ya ordenada, asumiendo que su primer elemento es la parte ordenada. El algoritmo ordena los elementos de forma ascendente, por lo que el último paso es invertir el ranking para que las probabilidades más altas sean las que se muestren primero.

6 Etapa 3: Optimización del *scheduling*

En este apartado se llega a la **etapa 3** de la figura 1. El objetivo es, dado un ranking de contenidos del evento en base a las probabilidades computadas, y los horarios de los contenidos y actividades del evento, obtener la mejor solución posible para el problema planteado, esto es, un programa o *scheduling* para el asistente.

Antes de comenzar, recordar que, como se indica en el punto 2.2, los algoritmos que se van a diseñar son un heurístico constructivo y un algoritmo genético. Por lo tanto, dado un ranking de preferencias y los horarios de los contenidos del salón se aplican ambas estrategias para optimizar un *scheduling*.

6.1 Formalización del problema

La última fase para construir el *scheduling* es la de optimización. El objetivo es formalizar el problema de optimización definiendo su espacio de búsqueda, función objetivo y restricciones que se le van a aplicar.

El primer paso es, como se acaba de indicar, definir el espacio de búsqueda del problema. Es decir, el conjunto de todas las posibles soluciones del problema. En este problema de optimización en particular, el espacio de búsqueda de soluciones se define como el conjunto de todos los vectores binarios de tamaño n:

$$\Omega = \{x \in \{0,1\}^n\}$$

donde $x(i) = 1$ denota que el contenido i se ha incluido en el *scheduling*, (se denota con un 0 cuando no se incluye). El número de posible soluciones en el espacio de búsqueda es 2^n.

Por otro lado se define la función objetivo, o función de coste. Esta función mide la calidad de una solución x. El problema de optimización consiste en encontrar aquella $x \in \Omega$ que maximice la función:

$$\arg\max_{x \in \Omega} \sum_{i=1}^{n} x(i) \cdot p(x_i)$$

donde $p(x_i)$ es la probabilidad de que le guste la actividad x_i. Aunque el objetivo es optimizar la función de coste encontrando una solución x en el espacio de búsqueda, es necesario que las soluciones cumplan una restricción relacionada con el solapamiento de las actividades del salón. Cuando se utilice el algoritmo heurístico constructivo la restricción que se va a aplicar es que no puede haber solapamientos entre los contenidos que se le vayan a recomendar al asistente dentro de una solución, independientemente de cuáles se le hayan recomendado. En otras palabras, que el solapamiento (función g en la ecuación) que exista entre ellos sea 0:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} x(i) \cdot x(j) \cdot g_{x(i),x(j)} = 0$$

No obstante, hay un caso a tratar por separado. Sabiendo que la duración de las exposiciones que se suelen preparar en salones suele ser la misma del evento, a

la hora de sugerir contenidos se gestionan aparte. Si la probabilidad de que cierto contenido, etiquetado como *exposición*, es mayor o igual que cierto número n se le propone al asistente fuera del *scheduling*.

6.2 Algoritmo heurístico constructivo

En el punto previamente mencionado se presenta la idea de diseñar y utilizar un algoritmo heurístico constructivo para construir un *scheduling*. Para este problema se va a partir de una estructura de *scheduling* inicialmente vacía y se va a ir rellenando con contenidos que, según el sistema, le van a gustar al asistente.

Para la implementación de este algoritmo se decide seguir un esquema de *backtracking* [6]. *Backtracking* es un algoritmo que sirve para encontrar soluciones a ciertos problemas de cómputo que incrementan los candidatos a las soluciones o los abandonan tan pronto como determinan que no pueden ser utilizados para dar con una solución válida. Es decir, que, para este problema, cada vez que se encuentre un contenido que pueda añadirse al *scheduling*, se observa qué solución se obtiene tanto si se decide que se va a introducir como si no, intentando conseguir la mejor solución.

A continuación, en la figura 14, se presenta un ejemplo [7] de un problema que se puede solucionar por *backtracking*[6]:

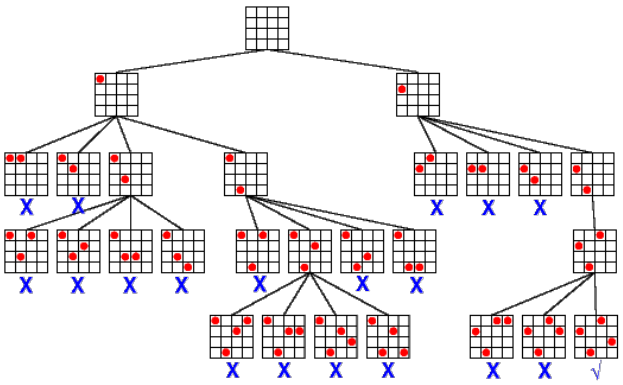


Figura 14: Ejemplo del problema de las cuatro reinas solucionado por *backtracking*.

Además, a lo largo de todo el proceso se tienen en cuenta los criterios de solapamiento definidos.

Por lo tanto, la primera estrategia para este problema es diseñar un algoritmo de este tipo, que se explica a continuación.

La función principal separa los contenidos que son actividades de los que son exposiciones, llama a la función de *backtracking* con las actividades del ranking, sus horas correspondientes en el evento y los horarios de las actividades del evento. Una vez obtiene el *scheduling* de esta segunda función y añade las exposiciones a los contenidos que se van a recomendar.

[6]https://ivoroshilin.wordpress.com/2015/02/05/toughest-backtracking-problems-in-algorithmic-competitions/

Algoritmo 1 Algoritmo heurístico constructivo

Rk: ranking de contenidos del asistente k con probabilidad >= 0.5
H: horarios de los contenidos, que sean actividades, del evento
for i in R_k **do**
    if El contenido es una actividad **then**
        $P \leftarrow$ Probabilidad del contenido
        $C \leftarrow$ Hora del contenido
    else
        $c \leftarrow$ Exposición
    end if
end for
$S \leftarrow$ sacarSchedulingBacktrack((P,C), (), H)
contenidos $\leftarrow$ contenidos de S
Añadir expos de c a *contenidos*
return (*contenidos*, horas de S)

Para la función de *backtracking* se sigue la siguiente estructura: primero se consulta si quedan elementos en el ranking de contenidos que le pueden gustar al asistente, ya que cuando no queden más significa que no se pueden añadir más sugerencias al *scheduling*, por lo que se puede devolver una posible solución. En caso contrario, se van revisando los elementos que tiene el ranking, y se van realizando llamadas recursivas a la función para ver qué contenidos se terminarían sugiriendo dependiendo de si se recomiendan los elementos revisados o no, o si estos elementos van a solapar con los ya recomendados o no.

Algoritmo 2 *Backtracking*

R: ranking de contenidos del asistente
S: sugerencias de contenidos hasta el momento
H: horarios de los contenidos del evento
if R es vacío **then**
    return S
else
    if hay sesiones del primer contenido de R que no solapen con las sugerencias de S **then**
        for *contenidos* **do**
            $STemp \leftarrow$ Llamada recursiva añadiendo el contenido a S y quitando el primer elemento de R
        end for
        $STemp \leftarrow$ Llamada recursiva sin añadir el contenido a S y quitando el primer elemento de R
        return Valor máximo de $STemp$
    else
        $SB \leftarrow$ Llamada recursiva sin añadir el contenido a S y quitando el primer elemento de R
        return SB
    end if
end if

6.3 Algoritmo genético

La segunda estrategia para este problema es diseñar un algoritmo genético, que se explica en el pseudocódigo presentado en el Anexo 4.

Tal y como se menciona en el punto previamente mencionado, los Algoritmos Genéticos trabajan con una población de individuos, siendo cada uno de ellos una posible solución al problema planteado. Su objetivo es encontrar el individuo más "apto" de la población, y lo buscan hasta que éste cumpla cierta condición, o hasta que se haya repetido el proceso de selección n veces.

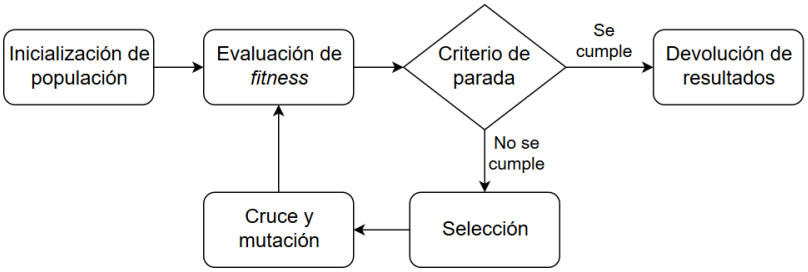


Figura 15: Esquema del funcionamiento básico de un algoritmo genético.

El algoritmo tiene la estructura [8] presentada en la figura 15: se empieza por la inicialización, donde se genera una población inicial aleatoria de n individuos. Después se computa el *fitness* (la aptitud) de cada individuo dentro de la población, y se consulta si se cumple el criterio de parada definido en el algoritmo. Si se cumple se devuelven los resultados pertinentes, pero si no se sigue al siguiente paso: selección. Se seleccionan los k mejores candidatos para, después, cruzarlos y generar mutaciones en la población. Tras estos pasos se obtiene un nuevo conjunto de datos, que seguramente tendrá algunos de los mejores candidatos de la población de la que se partía y algunos de la nueva población. Se vuelve a computar el *fitness* de cada individuo y se repite el proceso hasta que se cumple el criterio de parada.

Para este proyecto, se define una población inicial de 100 individuos, donde cada individuo es un vector binario de tantos elementos como atributos clase hayan quedado en el conjunto de datos tras la selección de variables. Además, se tienen en cuenta los contenidos que se repiten durante el evento a distintas horas. Es decir, que a la hora de seleccionar los contenidos que se van a recomendar al asistente, si una ocurrencia de uno de ellos ya ha sido recomendada, las siguientes no lo son.

Por otro lado, se toman una serie de decisiones con respecto al algoritmo. La primera es que el criterio de parada es cumplir 200 iteraciones. En cuanto al operador genético de cruce, se decide utilizar *One Point Crossover*, que intercambia los genes de los dos mejores individuos de cada iteración hasta una posición aleatoria. Por otra parte, para el operador de mutación se decide generar dos números aleatorios, que serán, respectivamente, las posiciones de los genes que se mutarán en los dos mejores individuos de cada iteración.

Por último, cuando se haga uso del algoritmo genético se va a buscar que

haya el mínimo solapamiento posible entre los contenidos. Para asegurar esto, a medida que se vayan añadiendo contenidos al *scheduling* se va a revisar que los que se vayan a introducir nuevos no produzcan solapamiento con los ya insertados. En caso de que sí lo produzcan, se aplica una penalización para que la puntuación que obtenga ese individuo sea decrementada. A continuación se presenta la función que se utiliza para este propósito, siendo μ el valor de penalización y la función g el solapamiento:

$$f(x) = \sum_{i=1}^{n} p_i(x_i) - \mu g(x)$$

7 Experimentación

Con el fin de validar el prototipo del sistema inteligente planteado en esta sección se llevan a cabo una serie de experimentos en los que compararemos los *schedulings* obtenidos con cada uno de los algoritmos. Además, en este apartado se pretende analizar algunas de las decisiones que se han tomado a lo largo del proceso de diseño en las diferentes etapas, y también guiar en la elección de parámetros y métodos.

7.1 Diseño experimental

En la **etapa 1**, tras ver los resultados de la figura 10, se decide generar dos conjuntos de datos. Se consideran dos α (el valor de aportación mínima que debe ofrecer un atributo para no ser descartado tras la selección): 0.7 y 0.85. Esto es, en el primer conjunto la última variable predictora que se guarda es el atributo número 17 y en el segundo conjunto el atributo número 21. De esta manera, se conseguirán ver las diferencias que puede haber entre ambos casos.

Tras realizar la selección de variables vemos que, de los 88 con los que se contaba inicialmente, ahora son 85 los atributos con los que se trabajará en el problema a afrontar con el primer conjunto de datos y 78 en el segundo.

Después, en la **etapa 2**, la tarea es generar los rankings de preferencias de los asistentes mediante *Naïve Bayes*. Se realizan dos pruebas, una por cada conjunto de datos que hemos generado durante la selección de variables. En todos los casos se decide utilizar 112 instancias, de las 118 totales como conjunto de entrenamiento y 6 como conjunto de prueba.

Por último, en la **etapa 3**, optimización del *scheduling*, se procede a generar los *schedulings* para los asistentes contenidos en el conjunto de prueba. Para un primer enfoque se hace uso del algoritmo heurístico constructivo diseñado. Se lleva a cabo 1 repetición para cada caso de prueba, puesto que los resultados no varían. Una vez obtenidos los resultados se procede al segundo enfoque, en el que se aplica el algoritmo genético. En este caso, se realizan 5 repeticiones por caso de prueba, puesto que la semilla de la que se parte es aleatoria y, por lo tanto, varía. Además, se define un tamaño de población de 100 individuos, donde cada individuo tiene tantos genes como contenidos se pueden recomendar. En ambos casos se tiene en cuenta la función de coste y los criterios de solapamiento previamente contemplados.

7.2 Resultados experimentales

El valor de *accuracy* que se obtiene con el primer conjunto de datos es 0.3573 y con el segundo 0.4179. Es decir, que la más alta corresponde a la obtenida con el segundo conjunto.

A continuación se presentan una serie de gráficos que representan los resultados obtenidos en la etapa 3, con ambos algoritmos, para compararlos.

En la figura 16 se muestra la comparativa de resultados del coste obtenido, para cada caso de prueba, con el algoritmo heurístico, con α=0.7 y α=0.85:

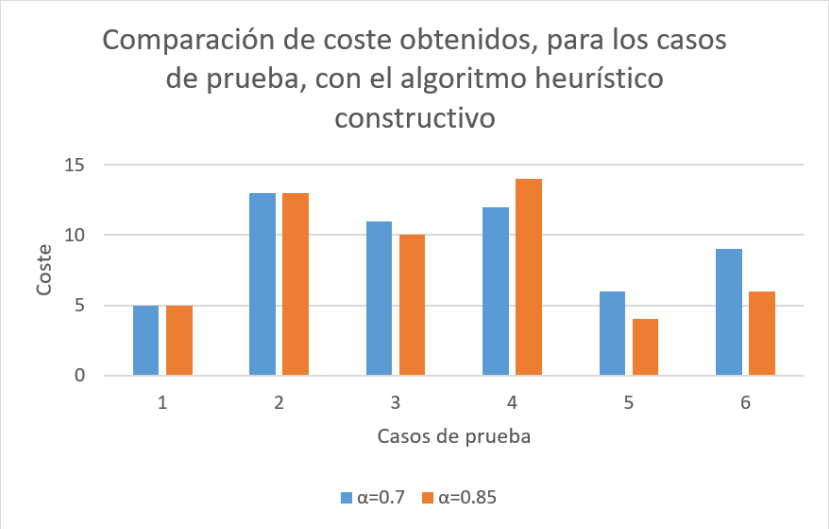


Figura 16: Resultados del coste obtenido con el algoritmo heurístico constructivo. Las pruebas se han realizado, tanto para α=0.7 como para α=0.85, con los seis casos de prueba.

A la vista de los resultados se puede observar que, pese a que en algunos casos hay alguna diferencia significativa, en general no hay grandes diferencias, con el algoritmo heurístico constructivo, entre los valores de coste obtenidos con α=0.7 y α=0.85.

Por otro lado, para observar el rendimiento del algoritmo genético se ha repetido 5 veces cada prueba para cada caso de prueba. Esto se debe a que, puesto que la población que se crea al comienzo del algoritmo es aleatoria, se requieren varias repeticiones del mismo para ver los cambios que sufre el coste partiendo de semillas distintas. Por lo tanto, la gráfica que se muestra a continuación, figura 17, muestra la media de los resultados de coste obtenidos para el primer caso de prueba, siendo $\alpha = 0.7$. Los resultados del resto de casos de prueba se muestran en el Anexo 5.

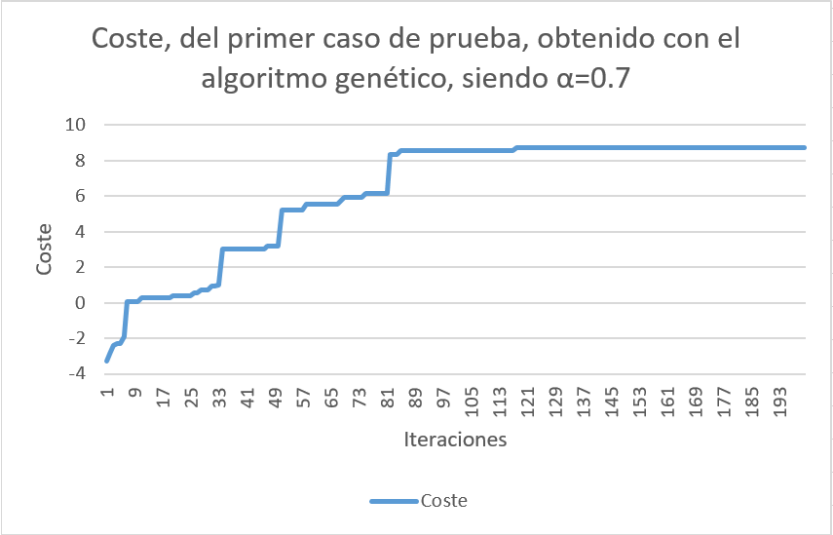


Figura 17: Media de los resultados del coste obtenido con el algoritmo genético para el primer caso de prueba para las 5 repeticiones, siendo α=0.7.

Ahora, se repite el proceso con α=0.85 (figura 18). Los resultados del resto de casos de prueba se muestran en el Anexo 6.

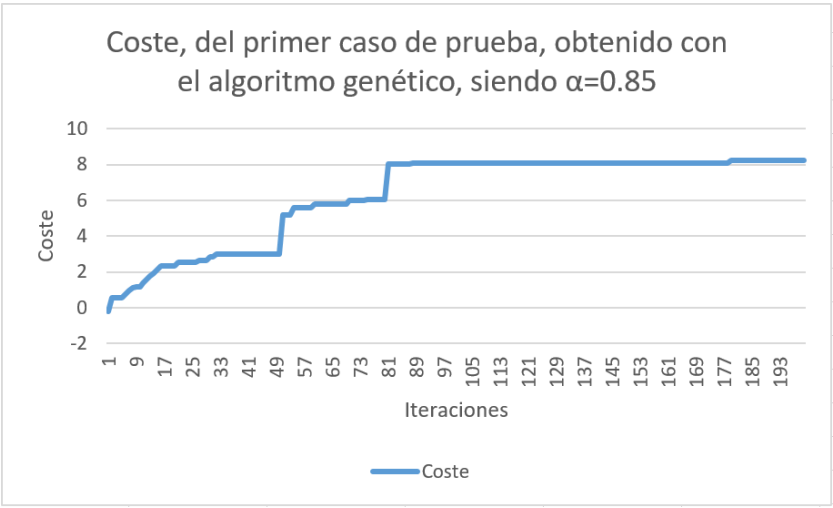


Figura 18: Media de los resultados del coste obtenido con el algoritmo genético para el primer caso de prueba para las 5 repeticiones, siendo α=0.85.

En ambos casos se observan resultados bajos en las primeras generaciones. Esto se debe a que, hasta que los individuos evolucionan y son más aptos, existe mucho solapamiento entre los contenidos que sugieren y, por lo tanto, la penalización que reciben es mayor. No obstante, una vez alcanzan el mejor coste que son capaces de obtener se mantienen estables hasta la última generación. Aun así, pese a que sí existen diferencias notables de coste entre los distintos casos de prueba, estas diferencias disminuyen entre los resultados obtenidos para cada uno de los casos de prueba con α=0.7 y α=0.85.

Una vez presentados los costes obtenidos por el algoritmo heurístico y el genético, con la media de sus 5 repeticiones, por separado, la siguiente comparación a realizar es del coste obtenido por ambos algoritmos, para todos los casos de prueba. A continuación, en la figura 19 se presenta la gráfica que representa estos resultados, siendo α=0.7:

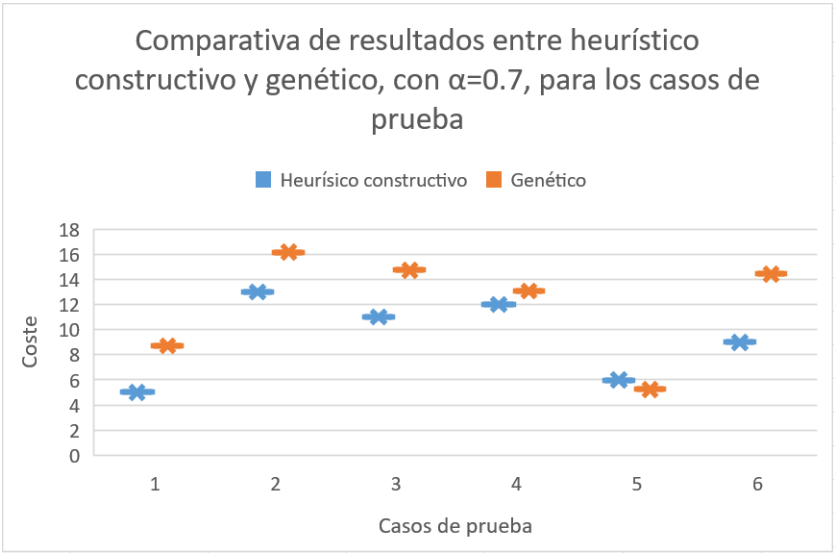


Figura 19: Comparación de resultados de coste obtenido con el algoritmo heurístico constructivo y genético, en la media de sus 5 repeticiones. Las pruebas se han realizado para α=0.7 con los seis casos de prueba.

A continuación, en la figura 20, se realiza la misma comparación para α=0.85:

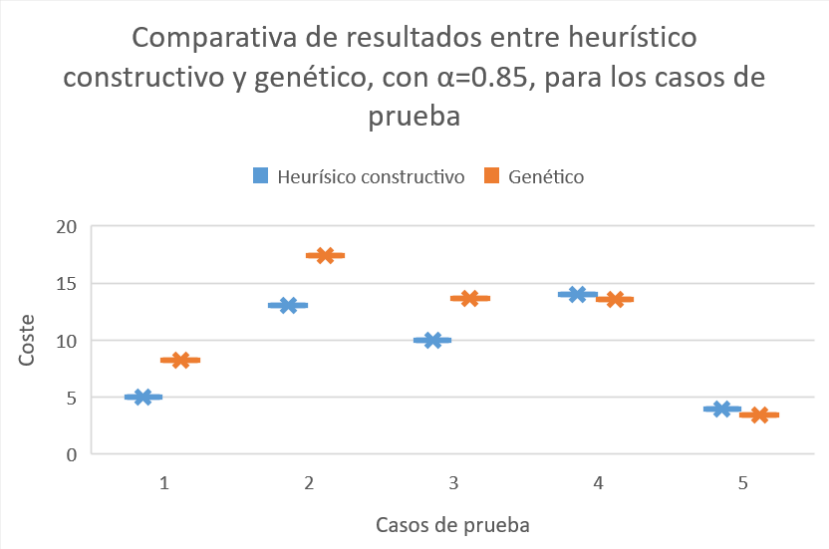


Figura 20: Comparación de resultados de coste obtenido con el algoritmo heurístico constructivo y genético, en la media de sus 5 repeticiones. Las pruebas se han realizado para α=0.85 con los seis casos de prueba.

A la vista de los resultados se puede afirmar que el coste obtenido con el algoritmo heurístico constructivo tiende a ser más bajo que el obtenido en los otros casos. Esto se debe a que, puesto que el heurístico no permite que exista solapamiento entre los contenidos sugeridos, se terminan recomendando menos contenidos, por lo que la suma de probabilidades final es más baja que la de los demás casos.

Por lo tanto, la última comparación a realizar es del número de contenidos sugeridos, por los algoritmos, a todos los casos de prueba. A continuación, en la figura 21 se presenta la gráfica que representa estos resultados, siendo α=0.7:

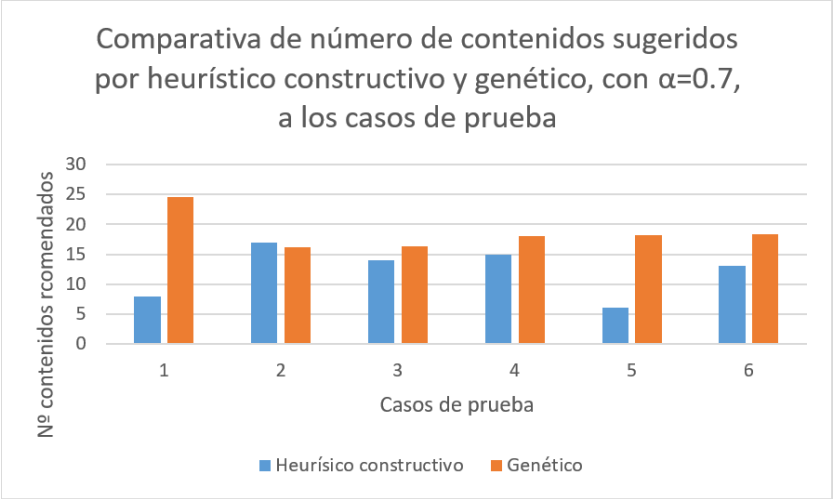


Figura 21: Comparación de número de contenidos sugeridos por el algoritmo heurístico constructivo y genético, en la media de sus 5 repeticiones. Las pruebas se han realizado para α=0.7 con los seis casos de prueba.

Después, en la figura 22, se realiza la misma comparación para α=0.85:

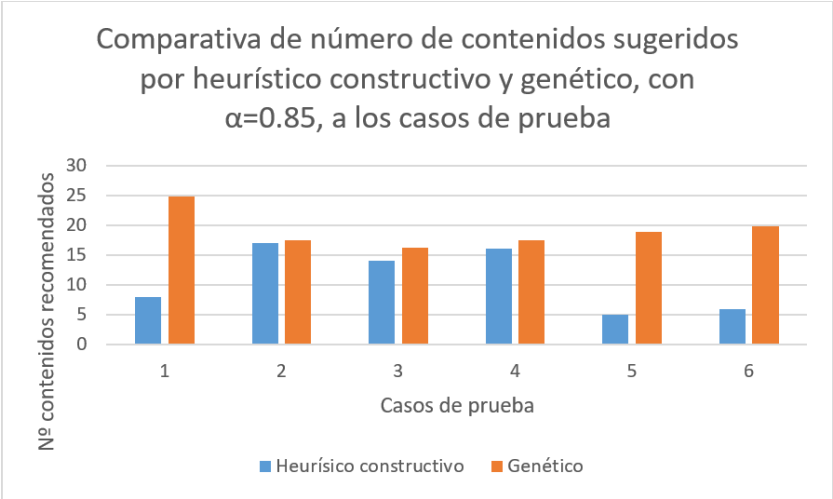


Figura 22: Comparación de número de contenidos sugeridos por el algoritmo heurístico constructivo y genético, en la media de sus 5 repeticiones. Las pruebas se han realizado para α=0.85 con los seis casos de prueba.

Tras visualizar los resultados se reafirma lo explicado previamente. Puesto que el algoritmo heurístico no permite que exista solapamiento entre los contenidos sugeridos, se recomiendan menos contenidos que en los demás casos. Para demostrarlo se presentan dos *schedulings* para el mismo asistente: uno de ellos propuesto por el algoritmo heurístico y el otro por el genético (ambos con α=0.7).

Figura 23: Comparación de *schedulings* propuestos por el algoritmo heurístico constructivo y el algoritmo genético para el mismo caso de prueba, ambos con α=0.7.

En conclusión, el algoritmo heurístico constructivo recomendará menos contenidos al asistente del evento, pero no habrá solapamiento entre ellos, mientras que el algoritmo genético recomendará más contenidos, pero habrá cierto solapamiento entre ellos.

8 Conclusiones

En este proyecto se planteaba un objetivo: construir un primer prototipo de un sistema inteligente que recomendara, a un asistente de un salón de cómic, un programa, para los días en los que se llevara a cabo el evento, ajustado a sus gustos. Para llevar a cabo esta misión se proponían una serie de etapas a recorrer y tareas a realizar, como la recogida de datos reales, la selección de variables del conjunto de datos, el aprendizaje de un clasificador, la generación de rankings o la optimización del *scheduling* que recibiría el asistente.

La recogida de datos del proyecto ha resultado un reto mayor del que se preveía en un principio. Los asistentes del evento no fueron muy proclives a completar la encuesta, por lo que se ha contado con un conjunto de datos menor del esperado para trabajar. Por lo tanto, si se tuviera que repetir el proceso habría que incentivar a los asistentes de otra manera a completarla.

Por otro lado, algunas de las herramientas utilizadas han precisado de una gran curva de aprendizaje. Meka, por ejemplo, ha sido una herramienta de la que no se tenía gran conocimiento previo al proyecto, por lo que en ocasiones la interpretación de los resultados ha resultado más complicada de lo esperado.

En cuanto a los algoritmos diseñados, el comportamiento que se ha observado ha sido que el algoritmo heurístico constructivo recomienda menos contenidos al asistente del evento, pero no existe solapamiento entre ellos. Por otro lado, el algoritmo genético recomienda más contenidos, pero existe cierto solapamiento entre ellos.

Por último, la organización del proyecto ha sido adecuada. Se ha seguido la planificación establecida y no ha habido grandes desviaciones sobre las dedicaciones temporales estimadas.

Por lo tanto, se puede afirmar que, en general y para tratarse de un primer prototipo, los resultados obtenidos han sido satisfactorios. Todo ello ha resultado en que se ha conseguido cumplir el propósito del proyecto, por lo que la sensación de haber terminado el trabajo es realmente gratificante.

9 Trabajo futuro

A continuación se presenta una serie de posibles tareas que se podrían realizar en un futuro para mejorar y ampliar el trabajo realizado:

Mejora del rendimiento del sistema inteligente

El rendimiento del sistema inteligente implementado en este proyecto ha sido satisfactorio. No obstante, si se siguiera trabajando en su desarrollo en el futuro se buscaría mejorar dicho rendimiento.

Para llevar a cabo esta tarea se buscaría, principalmente, conseguir una cantidad de datos considerablemente mayor que con la que se ha trabajado en el proyecto. De ser esto posible, se podría entrenar el modelo con un conjunto de datos mayor, por lo que llegado el momento de predecir las predicciones de los asistentes se esperaría obtener resultados más precisos.

Por otro lado, se consideraría aprender otro clasificador, como, por ejemplo, uno que hiciera uso de Redes Bayesianas [9] para obtener las predicciones. Así, se podrían comparar los modelos [10] y los resultados obtenidos con ambos.

El mismo principio podría aplicarse a los algoritmos diseñados para la optimización del *scheduling*. Además del algoritmo heurístico constructivo y el algoritmo genético, ya implementados, sería realmente interesante probar a realizar la misma tarea con otros, como, por ejemplo, con Algoritmos de Estimación de Distribuciones [11], y comparar los resultados obtenidos.

Creación de una aplicación

Desde el comienzo del proyecto se vislumbra la posibilidad de, en un futuro, crear una aplicación que ponga al alcance del asistente el sistema implementado. Sin embargo, en caso de que se decidiera seguir adelante con esta idea habría una serie de decisiones, que se explican a continuación, a tomar.

Para empezar, la primera decisión a tomar sería las plataformas a las que estaría dirigida. Concretamente, si la aplicación sería web y/o para móvil. En caso de que se decidiese hacerla para móvil habría otra decisión a tomar: si la aplicación sería nativa, híbrida o *web app*.

Por otro lado, se solicitarían los servicios de diseñadores gráficos que llevaran a cabo tareas como el diseño de la aplicación, la imagen de marca o el logo. Además, tendría que haber un equipo de desarrollo que trabajara en la aplicación y otro de asesoría legal que gestionara los asuntos de protección de datos.

En conclusión, sería una iniciativa muy interesante de llevar a cabo en el futuro pero inalcanzable en el momento de la realización de este proyecto.

10 Bibliografía

[1] Ríos Insua, Sixto, Concepción Bielza Lozoya, and Alfonso Mateos Caballero. Fundamentos de los Sistemas de Ayuda a la Decisión. 2002.

[2] Cheng, Weiwei, Eyke Hüllermeier, and Krzysztof J. Dembczynski. "Bayes optimal multilabel classification via probabilistic classifier chains." Proceedings of the 27th international conference on machine learning (ICML-10). 2010.

[3] Read, Jesse, et al. "Meka: a multi-label/multi-target extension to weka." The Journal of Machine Learning Research 17.1 (2016): 667-671.

[4] Saeys, Yvan, Iñaki Inza, and Pedro Larrañaga. "A review of feature selection techniques in bioinformatics." bioinformatics 23.19 (2007): 2507-2517.

[5] Zhang, Harry. "The optimality of Naïve Bayes." AA 1.2 (2004): 3.

[6] Pisinger, David. "Heuristics for the container loading problem." European journal of operational research 141.2 (2002): 382-392.

[7] Bitner, James R., and Edward M. Reingold. "Backtrack programming techniques." Communications of the ACM 18.11 (1975): 651-656.

[8] Abdeslam, Ahmadi & El Bouanani, Faissal & Ben-azza, Hussain. (2014). Four Parallel Decoding Schemas of Product Block Codes. Transactions on Networks and Communications. 2. 49-69. 10.14738/tnc.23.229.

[9] Sucar, L. Enrique, et al. "Multi-label classification with Bayesian network-based chain classifiers." Pattern Recognition Letters 41 (2014): 14-22.

[10] Calvo, Borja, Josu Ceberio, and Jose A. Lozano. "Bayesian inference for algorithm ranking analysis." Proceedings of the Genetic and Evolutionary Computation Conference Companion. ACM, 2018.

[11] Ayodele, Mayowa, John McCall, and Olivier Regnier-Coudert. "Estimation of distribution algorithms for the Multi-Mode Resource Constrained Project scheduling problem." 2017 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2017.

11 Anexos

Anexo 1

En este anexo se presenta el diseño de la base de datos (Figura 24) creada en este proyecto para el mismo y futuras aplicaciones.

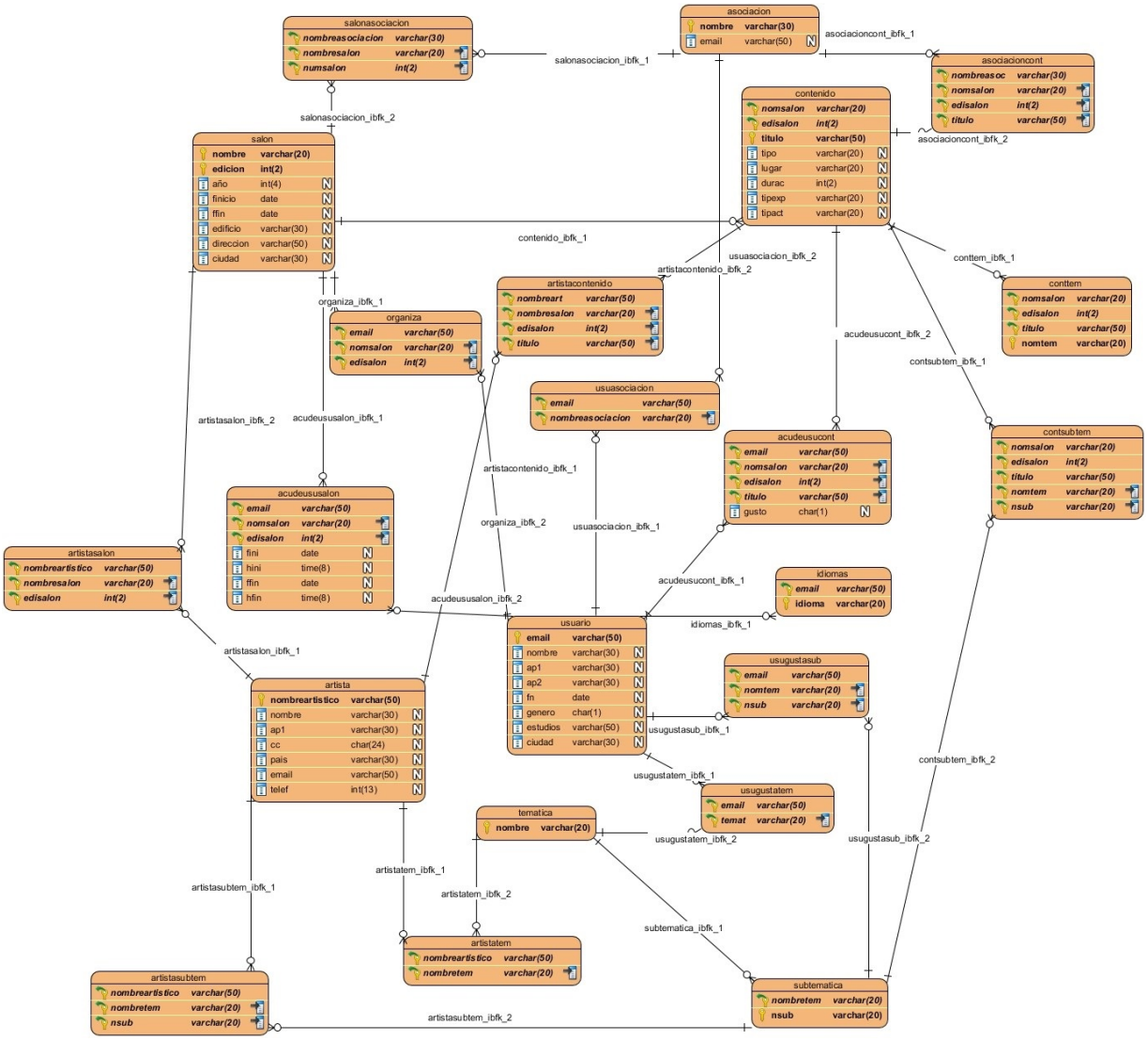


Figura 24: Diseño de la base de datos del proyecto

Anexo 2

A continuación se presenta la encuesta (figuras 25 a 33) que se realizó a asistentes de la tercera edición del Salón Internacional de Cómic y Manga de San Sebastián:

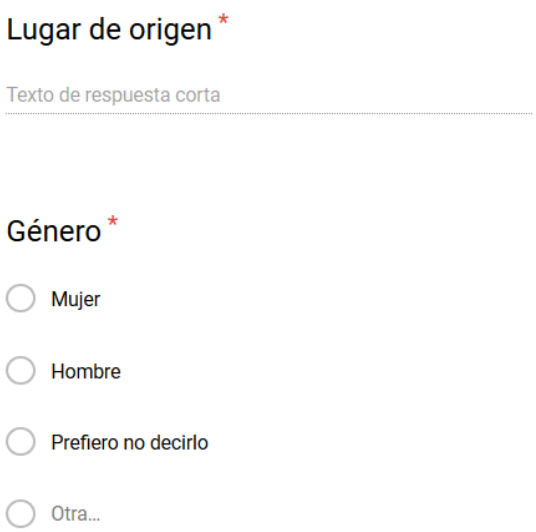


Figura 25: Primera parte de la encuesta realizada a asistentes del evento

Figura 26: Segunda parte de la encuesta realizada a asistentes del evento

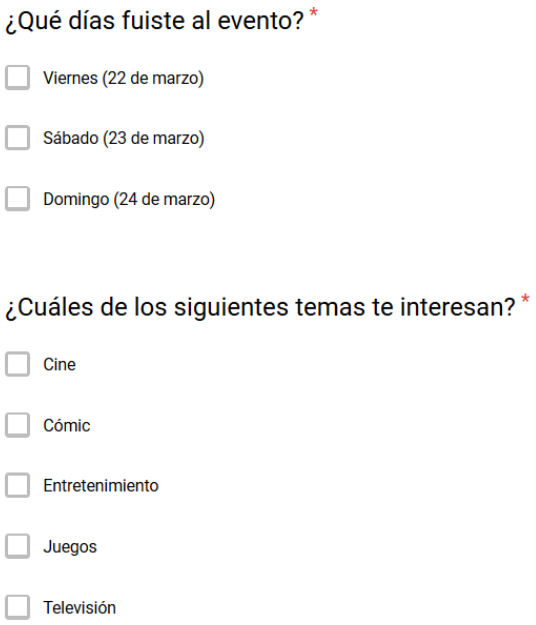


Figura 27: Tercera parte de la encuesta realizada a asistentes del evento

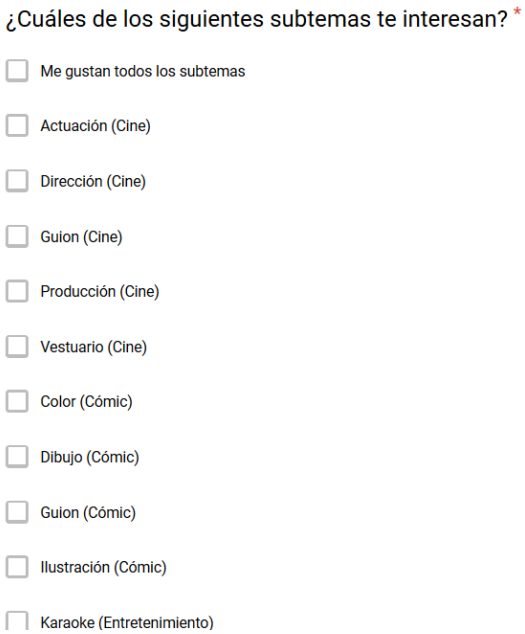


Figura 28: Cuarta parte de la encuesta realizada a asistentes del evento

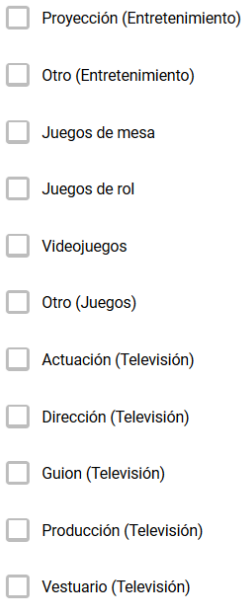


Figura 29: Quinta parte de la encuesta realizada a asistentes del evento

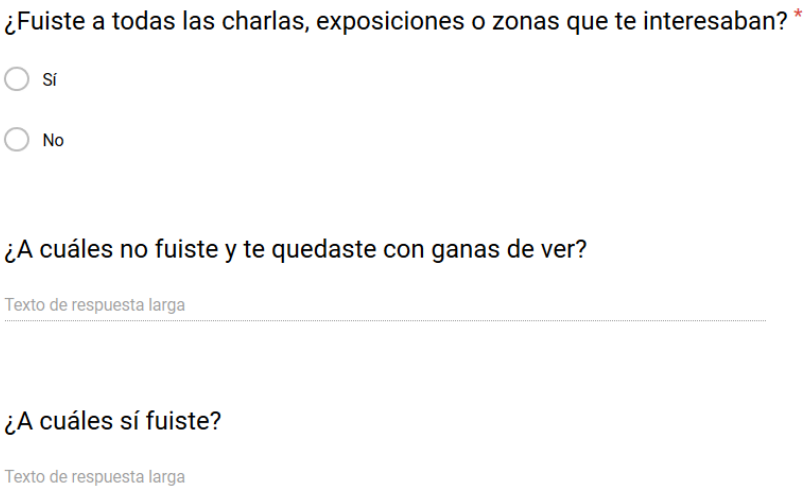


Figura 30: Sexta parte de la encuesta realizada a asistentes del evento

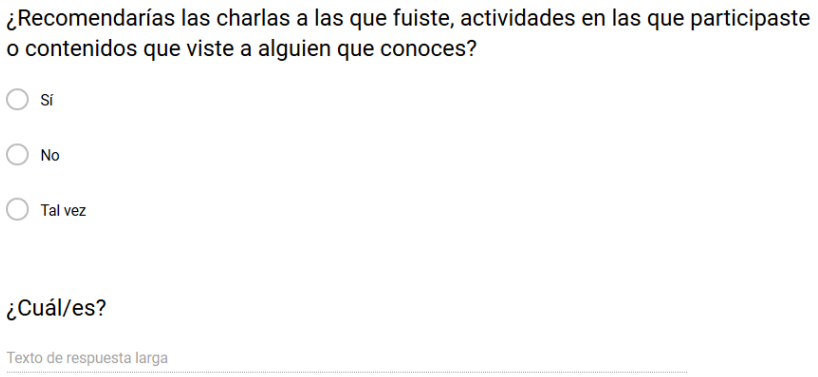


Figura 31: Séptima parte de la encuesta realizada a asistentes del evento

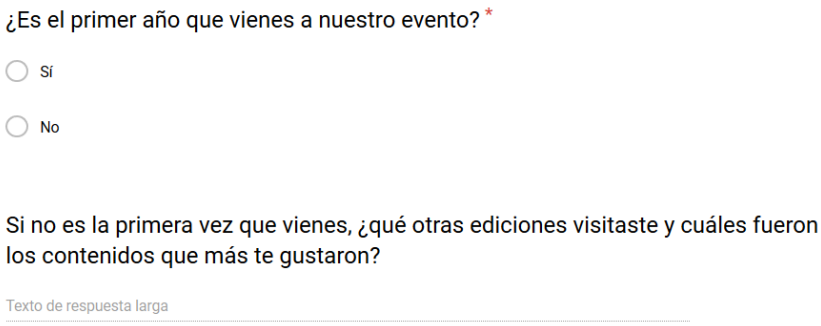


Figura 32: Octava parte de la encuesta realizada a asistentes del evento

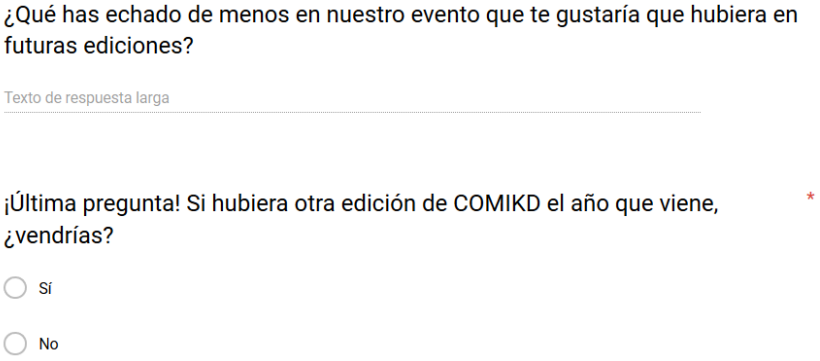


Figura 33: Novena parte de la encuesta realizada a asistentes del evento

Anexo 3

En este apéndice se muestra un fragmento (figura 34) del conjunto de emails aleatorios obtenidos con el generador de datos online:

	A	B	C	D
1	email			
2	Cras.interdum@sit.com			
3	ut.sem.Nulla@ultrices.co.uk			
4	sem.mollis@maurisMorbi.ca			
5	lectus@Duisa.ca			
6	est@insodaleselit.ca			
7	leo.Cras.vehicula@sem.net			
8	Donec@interdum.ca			
9	ligula@liberoettristique.net			
10	mauris.Suspendisse@temporest.com			
11	Cras@accumsannequeet.org			
12	risus@sollicitudincommodoipsum.ca			
13	est.mollis.non@dictumPhasellus.net			
14	mollis.dui.in@Mauris.org			
15	semper.erat@tortor.co.uk			
16	volutpat.Nulla@Crasconvallis.net			
17	lacus@veliteusem.ca			
18	lectus@eratVivamusnisi.edu			
19	vel.sapien.imperdiet@erat.ca			
20	eget.ipsum.Donec@metusAliquamerat.co.uk			
21	cursus@arcu.org			
22	lorem.eu@a.co.uk			

Figura 34: Emails aleatorios obtenidos con un generador de datos online.

Anexo 4

A continuación se presenta el pseudocódigo del algoritmo genético diseñado para la tarea de optimización del *scheduling*. El algoritmo sigue una estructura como la descrita en el punto 6.4, pero la función que se implementa para evaluar los individuos se adapta al problema a resolver en el proyecto.

Algoritmo 3 Algoritmo genético

```
Inicializar población P al azar
Evaluar individuos P_i
while N != 200 do
    Seleccionar padres de P
    Producir cruce a partir de los padres seleccionados
    Mutar los individuos hijos
    Extender P añadiendo hijos
end while
return Hijo con mayor puntuación
```

La siguiente función evalúa el coste de los individuos de la población. Para cada gen de un individuo comprueba si los contenidos que se quieren sugerir son actividades o exposiciones, ya que las exposiciones se gestionan por separado. En caso de que sean actividades, se comprueba si van a producir solapamiento o si ya se han propuesto otras sesiones de esos contenidos, en cuyo caso se aplica una penalización al coste. Sin embargo, tanto si son actividades que no producen solapamiento ni repeticiones como si son exposiciones se suma al coste la probabilidad de dicho contenido.

Algoritmo 4 Evaluación de individuos de la población

```
for numGenes do
    if Gen de numGenes == 1 then
        if numGenes es una actividad then
            contenido ← contenido y horas del contenido de numGenes
            if contenido no se solapa ni repite con contenidos ya propuestos then
                fitness ← fitness + probabilidad de Contenido
            else
                fitness ← fitness - 10 * probabilidad de Contenido
            end if
        else
            fitness ← fitness + probabilidad de numGenes
        end if
    end if
end for
```

Anexo 5

A continuación, en las figuras 35 a 39, se muestran la media de los resultados de coste obtenidos para los casos de prueba, exceptuando el primero (mostrado en el punto 7.2), siendo $\alpha = 0.7$.

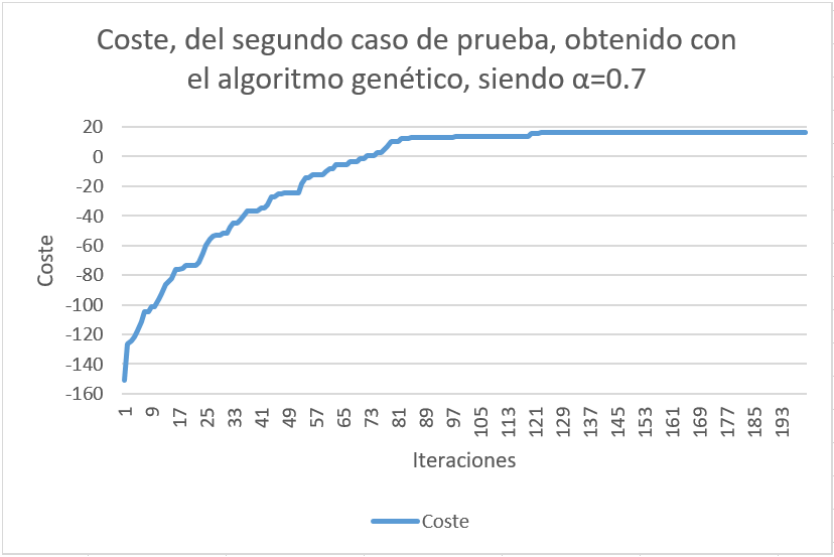


Figura 35: Media de los resultados del coste obtenido con el algoritmo genético para el segundo caso de prueba para las 5 repeticiones, siendo α=0.7.

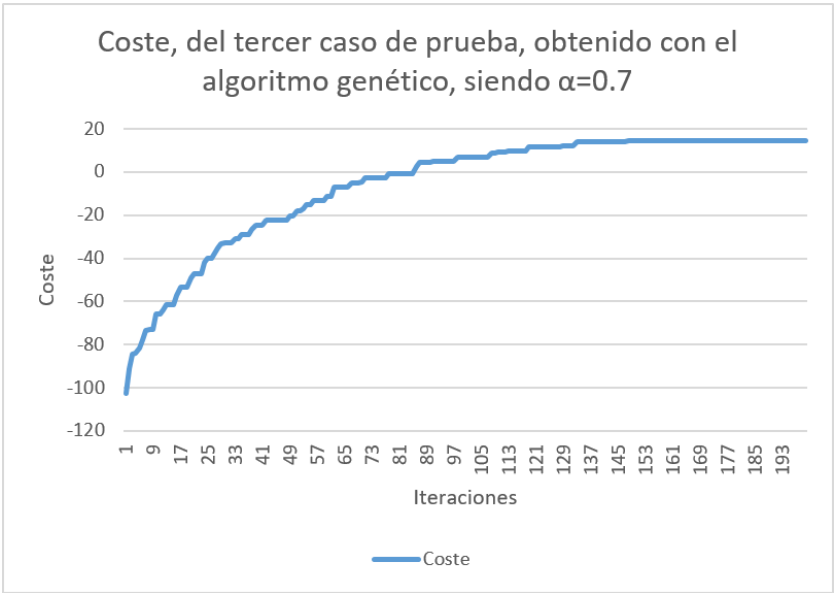


Figura 36: Media de los resultados del coste obtenido con el algoritmo genético para el tercer caso de prueba para las 5 repeticiones, siendo α=0.7.

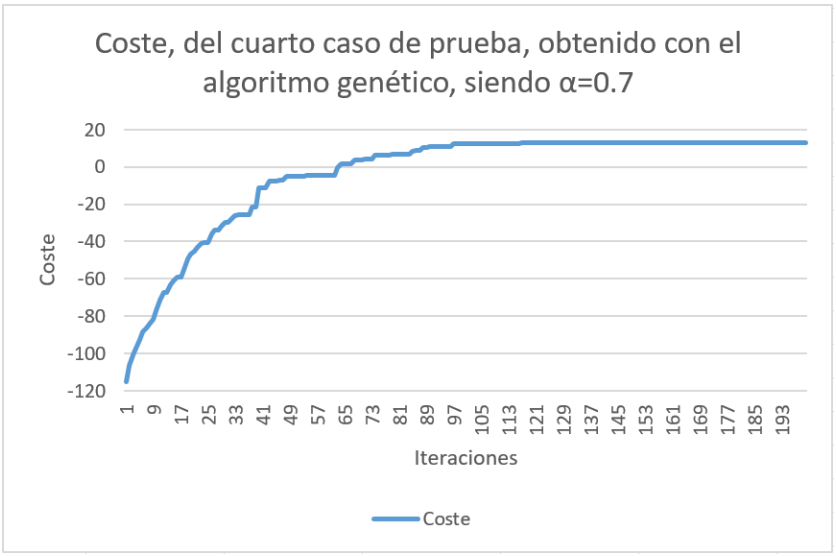


Figura 37: Media de los resultados del coste obtenido con el algoritmo genético para el cuarto caso de prueba para las 5 repeticiones, siendo α=0.7.

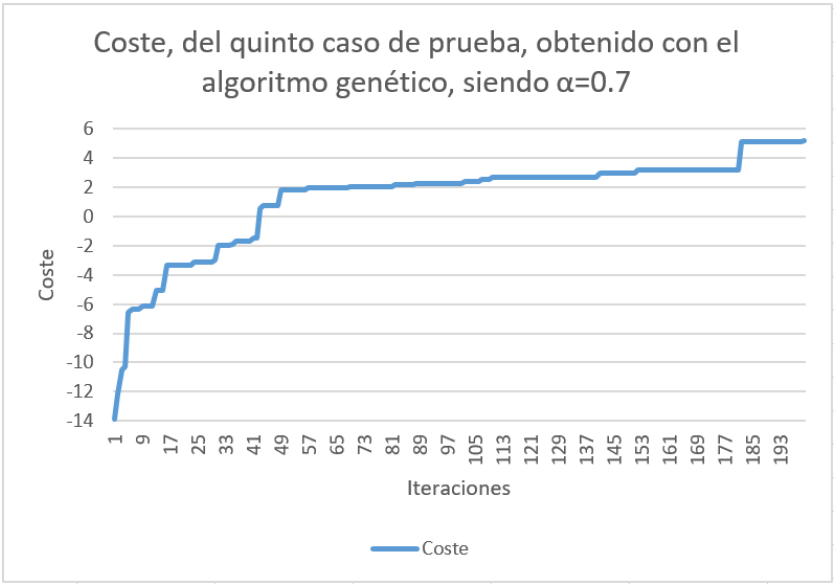


Figura 38: Media de los resultados del coste obtenido con el algoritmo genético para el quinto caso de prueba para las 5 repeticiones, siendo α=0.7.

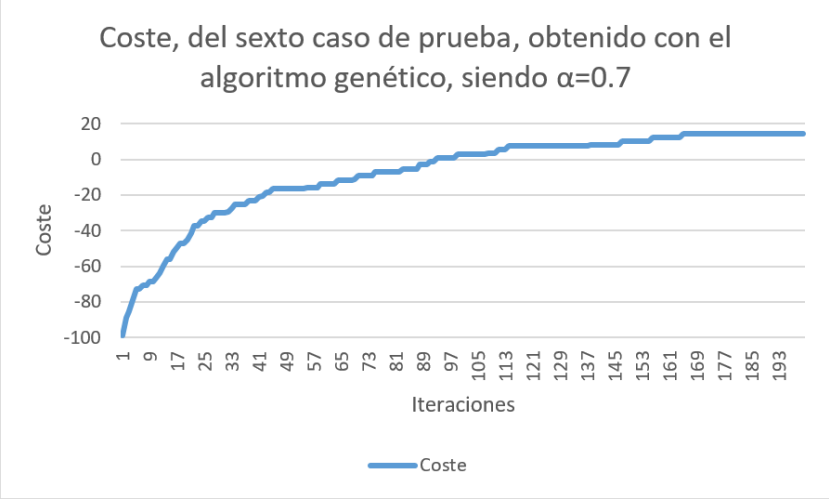


Figura 39: Media de los resultados del coste obtenido con el algoritmo genético para el sexto caso de prueba para las 5 repeticiones, siendo α=0.7.

Anexo 6

A continuación, en las figuras 40 a 44, se muestran la media de los resultados de coste obtenidos para los casos de prueba, exceptuando el primero (mostrado en el punto 7.2), siendo $\alpha = 0.85$.

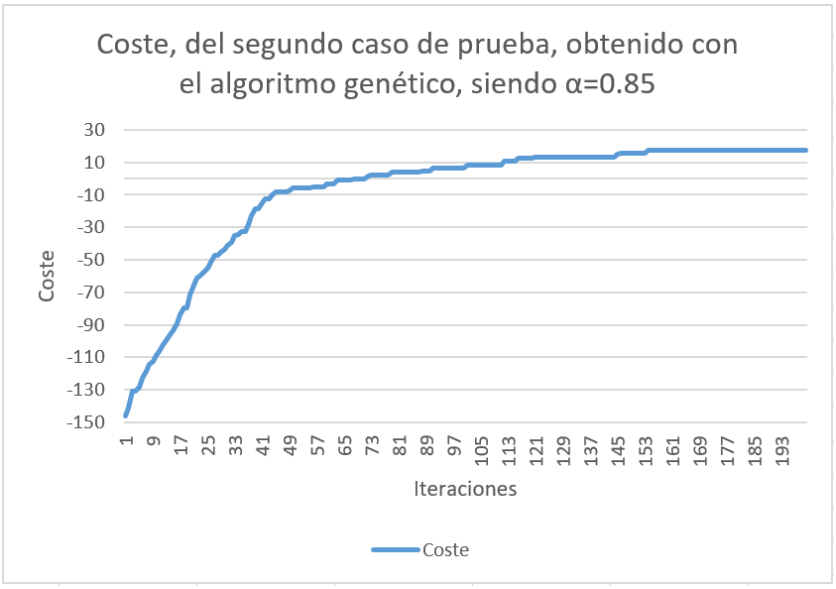


Figura 40: Media de los resultados del coste obtenido con el algoritmo genético para el segundo caso de prueba para las 5 repeticiones, siendo α=0.85.

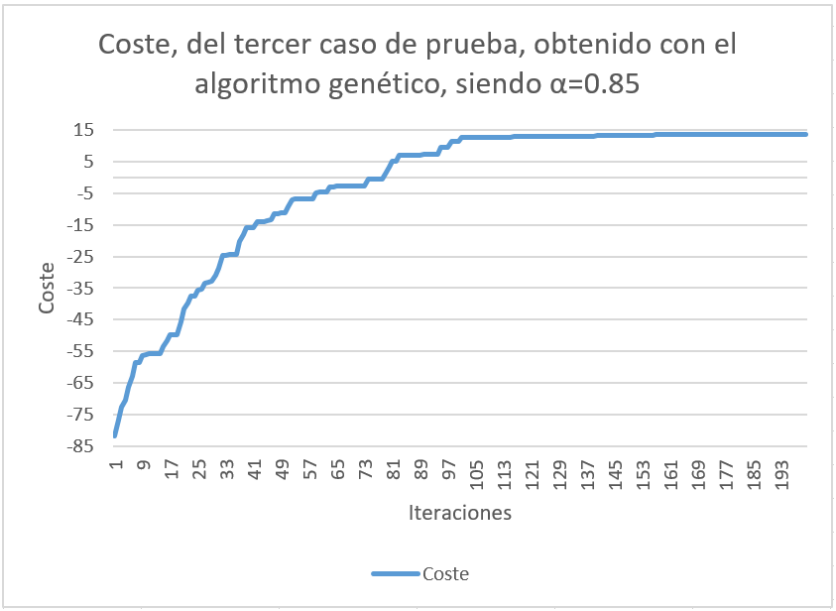


Figura 41: Media de los resultados del coste obtenido con el algoritmo genético para el tercer caso de prueba para las 5 repeticiones, siendo α=0.85.

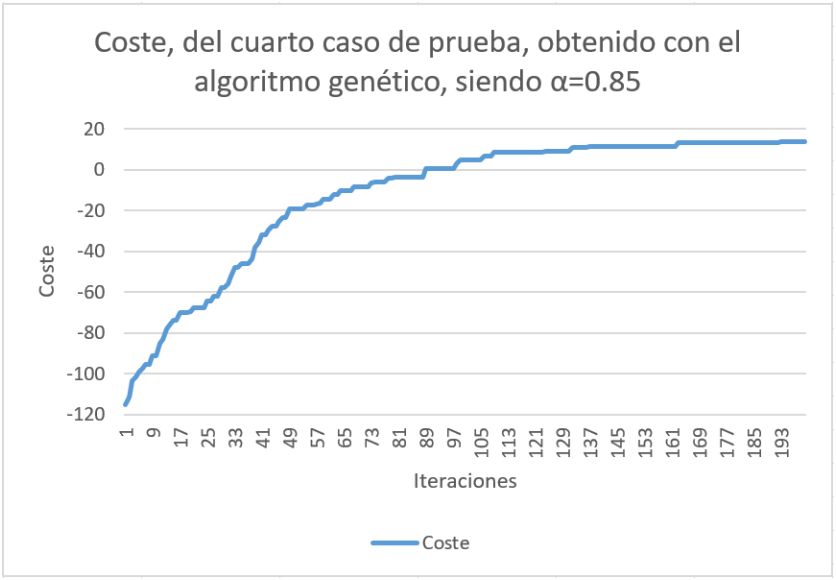


Figura 42: Media de los resultados del coste obtenido con el algoritmo genético para el cuarto caso de prueba para las 5 repeticiones, siendo α=0.85.

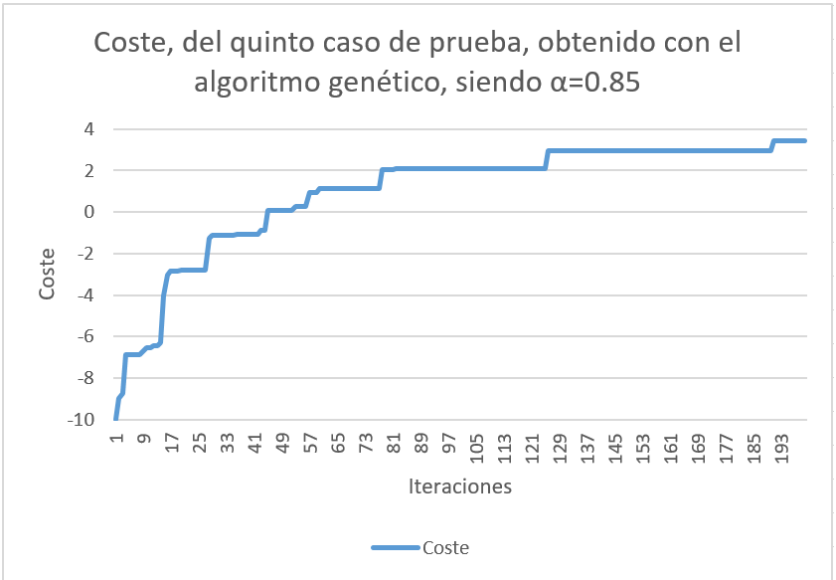


Figura 43: Media de los resultados del coste obtenido con el algoritmo genético para el quinto caso de prueba para las 5 repeticiones, siendo α=0.85.

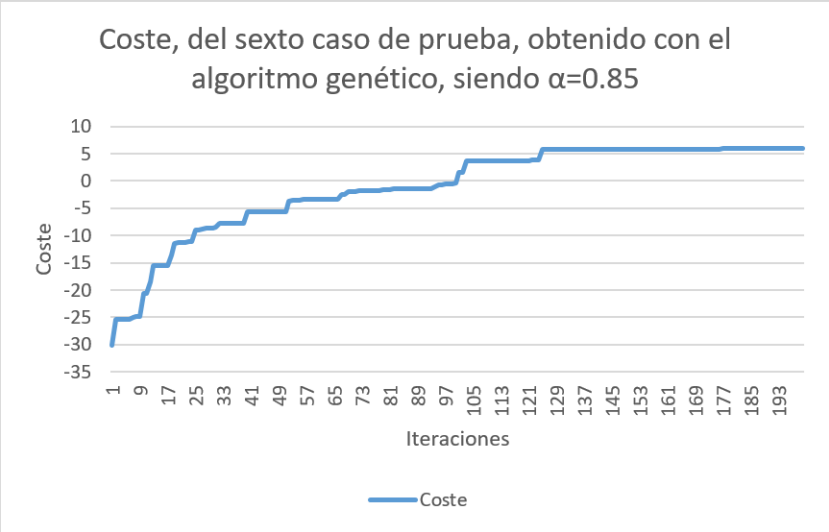


Figura 44: Media de los resultados del coste obtenido con el algoritmo genético para el sexto caso de prueba para las 5 repeticiones, siendo α=0.85.