UACM
Universidad Autónoma de la Ciudad de México
Nada humano me es ajeno

COLEGIO DE CIENCIA Y TECNOLOGÍA

LICENCIATURA EN MODELACIÓN MATEMÁTICA

Discriminación epistemológica en comunidades con discapacidad visual. Un caso en estudio

T E S I S

QUE PARA OPTAR POR EL TÍTULO DE

LICENCIADA EN MODELACIÓN MATEMÁTICA

P R E S E N T A :

ROCÍO ANTONIO ISLAS

D I R E C T O R

M. EN C. AGUSTÍN GONZÁLEZ VILLANUEVA

Ciudad de México, noviembre de 2023.

UACM
Universidad Autónoma de la Ciudad de México
Nada humano me es ajeno

CA Coordinación Académica

SISTEMA BIBLIOTECARIO DE INFORMACIÓN Y DOCUMENTACIÓN

UNIVERSIDAD AUTÓNOMA DE LA CIUDAD DE MÉXICO
COORDINACIÓN ACADÉMICA

RESTRICCIONES DE USO PARA LAS TESIS DIGITALES

DERECHOS RESERVADOS©

Índice general

Índice de cuadros

Índice de figuras

Agradecimientos

El apoyo de todas las personas que han estado a mi alrededor me ha ayudado para poder terminar mi tesis. Estoy agradecida con la vida. Deseo especialmente dar las gracias a mi mamá que no sólo me dio la vida, sino que me ha dado todo en mi vida: amor, cariño, educación, principios, etc.

Agradezco a mi asesor, el profesor Agustín Gónzalez, quien siempre me ha apoyado, en particular, por tomar la decisión de guiarme en la oscuridad de mis pensamientos para lograr aprender de la vida misma.

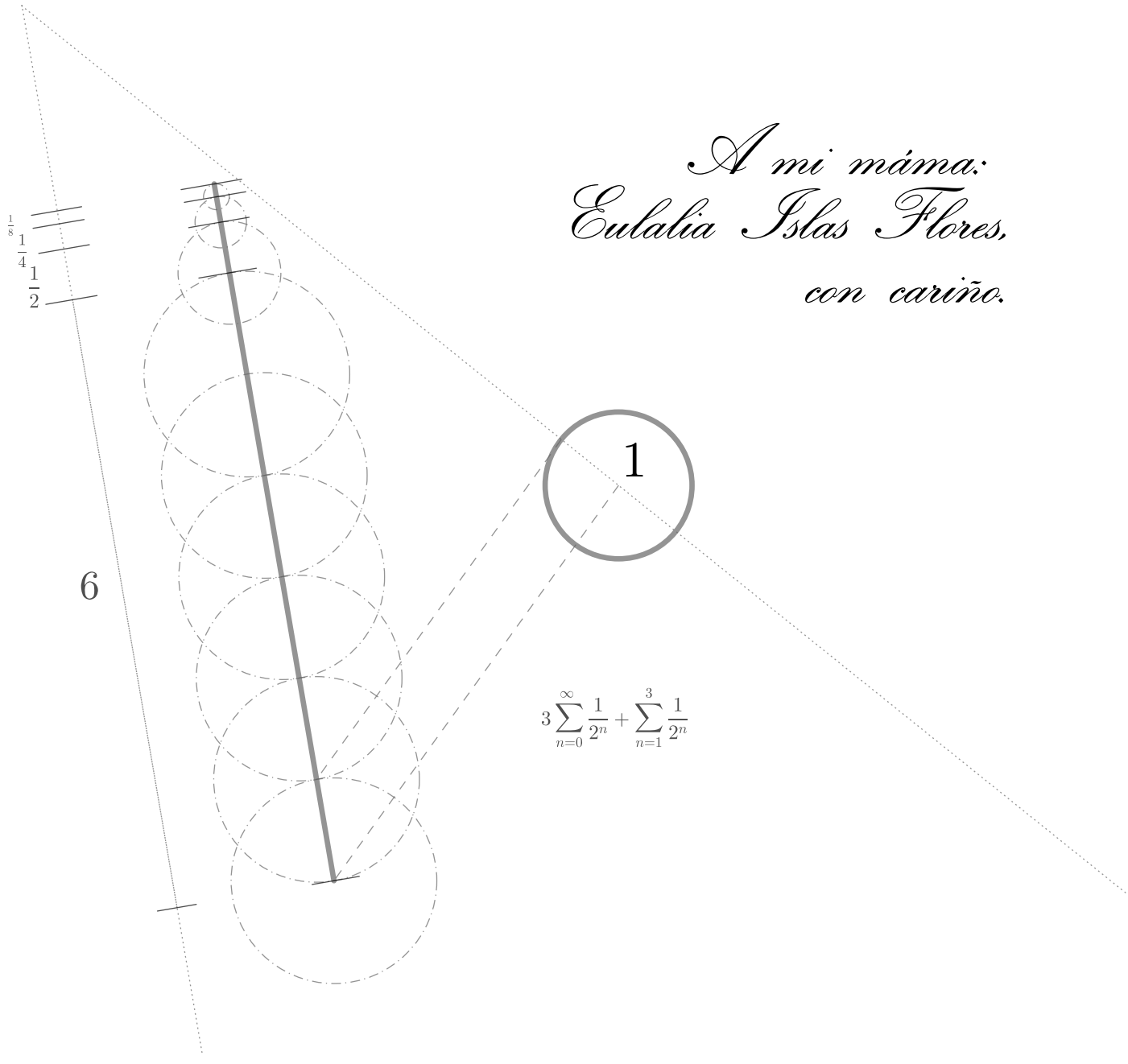
Al Laboratorio de Cómputo para la enseñanza y aprendizaje (LACECI) y al laboratorio de matemáticas (LAMAT) que fue como mi segunda casa. Encontré familia y amigos. Aprendí desde tipografía computacional hasta geometría.

A mi familia que siempre ha estado a mí lado en las buenas y en las malas, en especial a Balam que ha sabido motivarme a no abandonar mis metas.

Muchas gracias a todas las personas que me ayudaron a la correción de mi tesis: Yolanda Landeros, Enrique Cruz, Carlos Fuentes, Damiel Maisner.

Por último, a la Universidad Autónoma de la Ciudad de México (UACM) que me dió la oportunidad de conocer a un sin fin de personas maravillosas: a mis amigos.

Dedicatoria



A mi máma:
Eulalia Islas Flores,
con cariño.

Introducción

La modelación matemática es tan amplia como cada grano de arena, como cada fenómeno existente. Incluso, hay diversos métodos matemáticos describir del mismo fenómeno, es decir, distintos lentes para interpretar la realidad.

A lo largo del trabajo, se habla tanto del problema a fondo: la discriminación epistemológica que sufre el sector de las personas con discapacidad, como de una herramienta que sirve para modelar el aprendizaje de los usuarios para poder mejorarla. Se desarrolló una aplicación en Python, cuyo objetivo es ser una bisagra al horizonte de las matemáticas de conceptos abstractos, el cual está diseñado por medio de audio, enseña las operaciones básicas de teoría de conjuntos, es una breve introducción a sus propiedades.

Esta tesis es un ejercicio de recopilación y visualización de datos, donde se hace la aportación de un software que enseña teoria de conjuntos a las personas ciegas. Se trata de un tema poco estudiado, a saber: el aprendizaje de conceptos abstractos de matemáticas en personas ciegas y la barrera epistemológica. Al haber tan pocos estudios sobre el tema, está investigación es una exploración de los múltiples caminos que puede uno encontrar.

La discriminación es un problema que se encuentra en diversos sectores de la población y por distintas razones. Una de la más preocupante está en la educación, en la forma en cómo aprendemos, no todas las personas están contempladas en los esquemas ya existentes de la educación. Muchas veces ocurre por el desconocimiento de herramientas que permitan la accesibilidad a las diferentes aperturas del conocimiento, en otras, no existen tales. En este caso, es nuestra responsabilidad poder lograr abrir las puertas del aprendizaje: crear instrumentos técnicos. Esta tesis se ocupa en gran parte en esta apertura.

Aunado a lo anterior, la enseñanza de las matemáticas en los ciegos es un tema que se ha trabajado poco. Generalmente, se ha relegado a la aritmética y al álgebra en su espectro más sencillo, reduciéndolo a cuentas numéricas. Se ha dado el primer paso, mostrar la accesibilidad del conocimiento matemático, pero se ha trabajado poco el ir más allá. De tal manera que, el tema de teoría de conjuntos es un nuevo intento de construir las puertas de las propiedades abstractas del álgebra, en este caso la álgebra de Boole. El proyecto está dirigido al aprendizaje de los ciegos, no obstante todo el mundo que este interesado en aprender lo puede realizar como introducción a este aprendizaje.

Por otro lado, resulta de interés la visualización de los datos, el estudio sistemático de ciertas variables, para que el análisis y comprensión de las mismas, de tal manera que sean accesible al público en general.

En el primer capítulo, se hablará brevemente de los conceptos del aprendizaje, sobre los diversos recursos o aportaciones que ha habido respecto al aprendizaje de los ciegos. A continuación, se hace un planteamiento del problema y se verá la cuestión del arte, asimismo se enlaza con la discriminación que ha habido en el conocimiento. En el segundo capítulo, se toma la teoría de Bloom para crear de forma didáctica y justificada del por qué enseñarlo de ese modo, en cómo se debe diseñar el contenido y su importancia. En el tercer capítulo, se llevará acabo una descripción del proceso en el que se diseño la aplicación y cómo se recopilaron los datos, en otras palabras, está dedicado a hablar de la programación y recopilación de datos. En el cuarto, se hará un análisis estadístico exploratorio de los datos. Por último, se presentarán los resultados y las conclusiones.

Por otra parte, al incio de cada capítulo se encuentran fotos y una breve descripción del proceso que se llevó acabo en el uso de la aplicación de aprendizaje de teoría de conjuntos, desde el apoyo que se obtuvó del Sistema Nacional para el Desarrollo Integral de las Familias; hasta, el grupo de control que se obtuvo de la Universidad Autónoma de la Ciudad de México.

Capítulo 1

La discapacidad visual en la actualidad

23. La accesibilidad requiere que la educación a todos los niveles sea asequible a los estudiantes con discapacidad.[...]

Naciones Unidas, *Convención sobre los Derechos de las Personas con Discapacidad*

Grupo del Desarrollo Integral de las Familias que usó el modelo de aprendizaje de teoría de conjuntos. En la imagen de abajo, se encuentran sus familiares.

Introducción

En este capítulo se presenta la problemática y cuestión del arte de la educación de las matemáticas en las personas ciegas y cómo lo anterior deriva en una discriminación epistemológica y en qué sentido se puede entender. Se muestran las estadísticas del INEGI, las diversas soluciones que se han dado para hacer accesible el conocimiento a las personas con discapacidad visual y por qué es tan importante ver este sector de la población.

1.1. Un problema latente

Según el INEGI, se dividen en 4 categorías la medición de la discapacidad visual de una persona, respondiendo a la pregunta "¿Cuánta dificultad tiene para ver en su vida diaria aún usando lentes?" [3] La Organización Mundial de la Salud (OMS) toma la clasificación internacional de enfermedades del 2018, la divide la deficiencia visual de la siguiente manera:

Deficiencia de la visión		
Visión	Categoría	Agudeza Visual
Lejana	Leve	Inferior a 6/12
	Moderada	Inferior a 6/18
	Grave	Inferior a 6/60
	Ceguera	Inferior a 3/60
Cercana	Deterioro	menor a 40 cm

Para comprender la tabla 1.1, hay que aclarar que la visión normal es 6/6 según la escala de Snellen [1]. La agudeza visual se entiende como: "la capacidad de percibir y diferenciar dos estímulos separados por un ángulo determinado (α), o dicho de otra manera es la capacidad de resolución espacial del sistema visual"[11], básicamente se mide la capacidad de percibir detalles, entre menos detalles se vean mayor será la perdida de visión. Se considera débil visual a aquella persona que no es capaz de ver aunque ocupe algún tipo de utensilio, ya sea: gafas, lupas, etc. En el cuadro 1.1, se considera débil visual desde la categoría grave.

De acuerdo con la OMS , se estima que $2,200$ millones de personas a nivel mundial tienen un deterioro de la visión cercana a nivel mundial, número que aumentará significativamente, debido a que es un proceso propio del envejecimiento. Con respecto a la visión de lejos, 188,5 millones de personas tienen una deficiencia visual de moderada, es decir, con usar algún tratamiento, ya sean lentes o lentes de contacto su agudeza visual mejora considerablemente. Por otro lado, 217 millones tienen una deficiencia visual moderada a grave; y 36 millones son completamente ciegas. Estas representan aproximadamente el total de la población de Canadá, número que no es despreciable. Entre las diferencias regionales

[1] Generalmente, en los exámenes de vista, la escala decimal que se ocupa es: 20/200 , 20/100, 20/70, hasta 20/20, dónde ésta representaría una visión normal o si se quiere ver en porcentajes representaría el 100 % de agudeza visual.

que existen sobre este tema, se estima que la prevalencia en el deterioro de la visión es cuatro veces mayor en las regiones de bajo y mediano ingreso, respecto a las regiones de ingresos altos [4], simplemente, por hablar de un caso, el desprendimiento de retina puede ser curado cuando se diagnostica a tiempo y se opera, en otro caso termina siendo ceguera. De este modo, se observa que uno de los factores de la discapacidad visual es económico. Así es menester crear herramientas accesibles a estos sectores de la población.

En el último censo de México, que realizó el INEGI en el 2020, como responsable de la Subcomisión del Sistema Nacional de Información sobre Población con Discapacidad, se reportó que el 5,7 % (7, 168, 178 personas) de la población tiene algún tipo de discapacidad, cifra que aumentó considerablemente respecto al censo anterior del 2010 (5,1 %). De la población total de discapacidad, el 37,54 % (2, 691, 338 personas) [6] declararon tener dificultad para ver, incluso usando lentes, es decir, o son completamente ciegos o tienen muy baja visión.

A pesar de que la mayoría de las personas con discapacidad visual y ceguera tienen más de 50 años, la pérdida de visión afecta a personas de todas las edades. Cuando es a temprana edad, el niño puede sufrir retrasos en el desarrollo motor, lingüístico, emocional, social y cognitivo, con consecuencias irreparables. Después, la discapacidad visual en la adultez presenta tasas más bajas de participación en el mercado laboral y de productividad, además que también registran más problemas psicológicos como ansiedad o depresión. El aumento de personas ciegas o con visión baja debe involucrar mayor atención, sobre todo en la educación.

1.2. ¿Qué es la discriminación en el sector de la educación?

El Diccionario de la Real Academia define la palabra como: "*Dar trato desigual a una persona o colectividad por motivos raciales, religiosos, políticos, de sexo, de edad, de condición física o mental, etc.*", esto significa que, se excluye a un grupo de personas (o a una sola) de un conjunto de actividades o prácticas ya sean institucionalizadas o costumbres, por lo tanto, la discriminación muchas veces pasa desapercibida, simplemente por la normalización de nuestras prácticas sociales.

En el caso que nos ocupa, la dorma en que está diseñada la educación provoca una discriminación implícita, hay una forma en la que están dados los planes de estudio, los modos de cómo se debe enseñar. Uno de los principales problemas respecto a la discriminación de la educación, menciona Abraham Magendzo K., experto en la investigación educativa, es la falta de reconocimiento de la gran diversidad de personas que hay, pues la Convención relativa a la lucha contra la discriminación en la esfera de la enseñanza, estipula que: "se considera un acto discriminatorio en el plano educacional cuando se excluye a una persona o a un grupo del acceso a los diversos grados y tipos de enseñanza; cuando se limita a una persona o a un grupo a un nivel inferior de educación por razones ajenas a su capacidad; cuando se instituyen o mantienen sistemas de establecimientos de enseñanza separados para personas o grupos (artículo 1°)"[8]. Se deben de crear las condiciones y herramientas para que cada persona pueda acceder al sistema de educación. Se presenta como un reto,

pues existe una gran diversidad de formas de aprender.

No sólo la diversidad se presenta como un desafío, sino la institucionalización de cómo se enseña. En este caso particular, la educación está diseñada para personas normovisuales y las pocas herramientas para personas no visuales son escasas y de difícil acceso económico; en consecuencia, muchas veces el conocimiento no es accesible para un gran número de personas con algún tipo de discapacidad. En ese sentido, se puede hablar de una discriminación epistemológica.

1.3. Discriminación epistemológica en comunidades con discapacidad visual

La epistemología es una rama de la filosofía que se encarga de la teoría del conocimiento, se pregunta por el *cómo* y *qué* es lo que se puede aprender, ¿Cuáles son los factores que se relacionan con el mismo? Se pueden dar diversas respuestas a estas preguntas desde distintas perspectivas, si el conocimiento es una representación mental, si es un producto colectivo o si hay una intersección en ambas perspectivas. Hay ciertas ramas de la ciencia que se encargan de explicar las causas del conocimiento, desde la fisiología hasta las ciencias sociales dado las consecuencias. Mientras la epistemología trata de establecer las condiciones de validez y justificación del conocimiento, su objetividad. En ese sentido, se enfoca en la creencia, verdad y justificación [14]. Existen muchos conceptos fronterizos entre la epistemología y otras áreas del conocimiento como se podrá ver, de la misma manera, la teoría del conocimiento se ve indirectamente entrelazada con el aprendizaje, como una consecuencia de la misma, una aplicación de la teoría.

Pero entonces ¿Por qué hablar de una discriminación epistemológica en las comunidades con discapacidad visual? ¿Por qué traslapar conceptos que parecerían a primera vista distantes entre sí? Si la epistemología se encarga de la validez y justificación del conocimiento ¿No debería inmiscuirse en esta discusión? No se trata de sugerir que no puedan aprender o que no tengan las capacidades, etc., o incluso de dar proposiciones objetivas de la epistemología, ni de hacer una teoría que indique cuál es el camino de la objetividad: No es una teoría del conocimiento. Se trata de señalar la poca accesibilidad que existe en diversos temas de aprendizaje, no sólo en matemáticas avanzadas; también en otras áreas del conocimiento. La poca apertura de las herramientas, medios y diseños de planes hacia personas con discapacidad visual han, de alguna manera, bloqueado las puertas del conocimiento. Esto crea una barrera al conocimiento, un bloqueo a las condiciones de éste, pues hay una carencia del *cómo* acceder a él.

En la figura 1.3, se representa el problema de discriminación de personas con discapacidad visual a partir de pensarlo desde sus contrarios. Existe un antagonismo o una escala de contrarios: lo visual y lo no visual, el conocimiento básico y no básico, entre los que desean aprender y los que no saben que existe ese conocimiento. La cuestión a resolver trata de borrar esa escala de accesibilidad, esa frontera que naturalmente parecería tener sentido. Crear formas, maneras para poder lograr mitigar obstáculos de conocimiento. Este es el objetivo en general de esta investigación: crear una herramienta que suprima en lo más posible los contrarios.

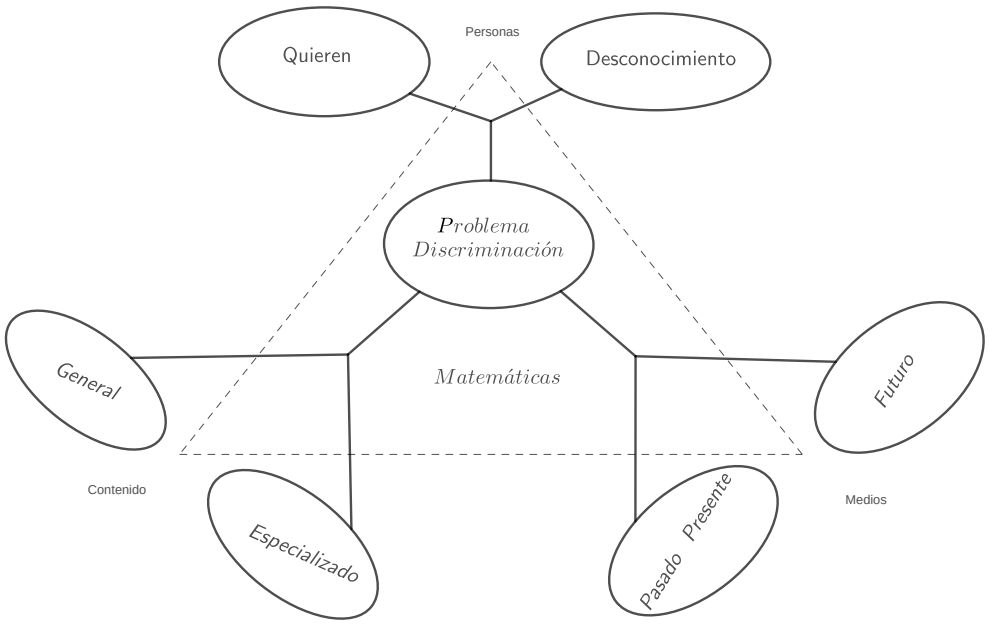


Figura 1.1: Representación gráfica del problema

1.4. La discapacidad visual en la educación

Una problemática que se presenta en todas las edades es la accesibilidad al conocimiento y aprendizaje para personas con discapacidad visual, como se vio en la sección anterior, ya que puede dificultar el aprendizaje de la lectura, procesos de información que se enseñan visualmente, etc., causando niveles más bajos de rendimiento académico, o incluso el completo desinterés o deserción del estudio por falta de atención o falta de accesibilidad. En el 2004, como se muestra en la tabla 1.4, solo el 31,5 % de personas con discapacidad visual grave o ceguera consiguieron sólo una parte de la educación primaria pues abandonaron el estudio el restante y sólo el 3,8 % concluyeron sus estudios universitarios[2]. Esto nos habla de un alto índice de deserción desde inicios de la educación escolar e implica la poca accesibilidad o atención que tienen los estudiantes dada una enseñanza diseñada para normovisuales, pues en caso contrario veríamos que el abandono de la escuela sería más uniforme y no concentrado al inicio. Dadas las formas de enseñanza, se ha dificultado la accesibilidad a las personas con discapacidad visual.

Nivel de instrucción de personas con discapacidad visual grave o ceguera						
Sin instrucción	Primaria incompleta	Primaria	Secundaria incompleta	Secundaria	Media superior	Superior y Posgrado
34.6 %	31.5 %	14.2 %	2.5 %	6.8 %	5.3 %	3.8 %

En la enseñanza de las matemáticas la situación se complica más, ya que es una diciplina altamente visual que normalmente se enseña a través de gráficos, diagramas, representaciones visuales, dibujos, ecuaciones, remarcados, etc. En pocas palabras, se necesita tener una comprensión total del espacio geográfico, para tener la capaciddad de dimensionar áreas, cantidades en volumen, etc. No se trata de enfatizar texturas, olores, sino formas. Actualmente, ya existen algunos métodos para enseñar aritmética, e incluso álgebra a nivel básico, pero algunos son costosos o poco accesibles al aula. También, muchas de las cosas que hay para la enseñanza de matemáticas en personas ciegas suelen ser poco útiles, como veremos posteriormente.

Por ejemplo, en la educación básica comúnmente se usa el ábaco (chino, japonés, azteca) con algunas modificaciones para el aprendizaje de la aritmética. En otras ocasiones, tablas con algún tipo de relieve o con ligas para aprender las figuras geométricas. Pero esto se ve superado cuando se trata de abstraer las propiedades de las operaciones; por ello, como resultado de esta investigación, se desarrolló una aplicación auditiva que puede mostrar el camino para el entendimiento de dicho campo, cuestión que se explicará en el siguiente capítulo.

1.5. Herramientas y estrategias para la enseñanza de personas ciegas a lo largo de la historia

Con las primeras civilizaciones, la ceguera estaba llena de estigmas, se interpretaba como mal presagio, castigo a la familia, y hasta la religión lo veía mal, como un abandono por parte de Dios, como parte de la oscuridad, a pesar de haber sido vistos como aquellos que habían perdido la vista por ver a Dios [10]. Esto provocó que, al igual que a personas con otras discapacidades, se les abandonara a su suerte: usando un bastón como guía y método de defensa. En la edad media, algunos conseguían adaptarse y eran serviciales en sus comunidades con trabajos manuales (trabajos táctiles), mientras que una persona vidente les indicaba lo que tenían que hacer, fue una época de humanización, ya no era una razón divina sino de castigo o desconfianza. Dadas estas épocas, la educación no estaba contemplada. En la primera guerra mundial, se empezaron a entrenar perros para personas ciegas, que les guiaban y avisaban de situaciones de peligro.

Actualmente, se ha eliminado la creencia de que la ceguera tiene que ver con cuestiones divinas o por castigos y se sabe que es una consecuencia de diversas enfermedades: glaucomas, cataratas, refracciones no corregidas, infecciones oculares, desprendimiento de retina o incluso congénitas. Entre otras, existe una serie de tabúes acerca de las personas ciegas. No obstante, existe también una serie de herramientas que han favorecido la accesibilidad a diversos entornos a estas personas. A continuación, se describen brevemente algunas de estas herramientas que han servido para que tengan acceso a la educación y más específicamente en el área de las matemáticas. Para entender la cuestión del arte primero se hablará de sistema Braille.

1.5.1. Sistema Braille: Primera herramienta sofisticada para la enseñanza de personas ciegas

A pesar de que desde hace varios siglos ya existían avances tecnológicos notables, fue hasta el siglo XIX, en Francia, en el año 1829, cuando Louis Braille (1812-1852,) inventó un sistema que revolucionará la manera en la que las personas ciegas pueden leer y escribir.

Todo comenzó cuando el pequeño Louis de tres años, quien nació sin problemas de visión, jugando en el taller de arneses de su padre, se perforó un ojo jugando con una herramienta puntiaguda (lezna), que le provocó una grave infección que no sólo no sanó, sino que se le pasó al otro ojo y quedó completamente ciego a la edad de cinco años. A pesar de haber conocido el mundo con sus ojos, era tan pequeño que todavía no había aprendido a leer ni a escribir. Sus padres, determinados a que no debía perder la oportunidad de acudir a la escuela, lo inscribieron a la primaria local y para cuando cumplió los diez años, ganó una beca para estudiar en la Real Escuela de Ciegos de París. A medida que Braille avanzaba en sus estudios se dio cuenta de que las herramientas disponibles para la lectura y escritura para ciegos eran insuficientes y poco prácticas. Antes de Braille, ya se habían hecho varios intentos para que las personas ciegas pudieran leer: uno de ellos era resaltar el relieve de las letras como se conocían, pero esto resultaba poco práctico y muy difícil de leer.

Braille se inspiró en un sistema de lectura táctil, llamado sonografía, fue utilizado por los soldados franceses, inventado por Nicolás Charles Marie Barbier de la Serre, para transmitir mensajes en la oscuridad, de tal manera que los enemigos no pudieran escuchar sus mensajes. Después fue llamado escritura nocturna. Fue un antecedente de la escritura Braille, pues con éste se estaba tratando de enseñar a los ciegos, empero aún era muy poco práctico por la longitud en la distribución de los puntos en relieve que ocupaba. Consistía en dos columnas de seis puntos, un total de 12 puntos por sonido: característica que resultó obsoleta, pues eran difícil de recordar. El problema fue que los símbolos obedecían a los sonidos franceses y no a la ortografía francesa, lo cual hacía que la escritura no tuviera signos de puntuación y que la escritura no correspondieran. Braille, modificó y mejoró la escritura nocturna, creando así un nuevo sistema que se convertiría en el sistema que llevaría su nombre .

El sistema consiste en una serie de puntos en relieve dispuestos en celdas de seis puntos, que pueden ser leídas con las yemas de los dedos, como se ve en la figura 1.2. Los símbolos corresponden a las letras, no a los sonidos; logró resolver los signos de puntuación agregando algunas variantes. Por otra parte, este sistema también se utiliza para aprender a escribir matemáticas, los números son las letras del a-j, iniciando con un marcador que es un número.

Louis Braille falleció a los 43 años a causa de una tuberculosis, pero su sistema continuó evolucionando y mejorando. En la actualidad, el sistema Braille se utiliza en todo el mundo para permitir a las personas ciegas acceder a la literatura y la educación.

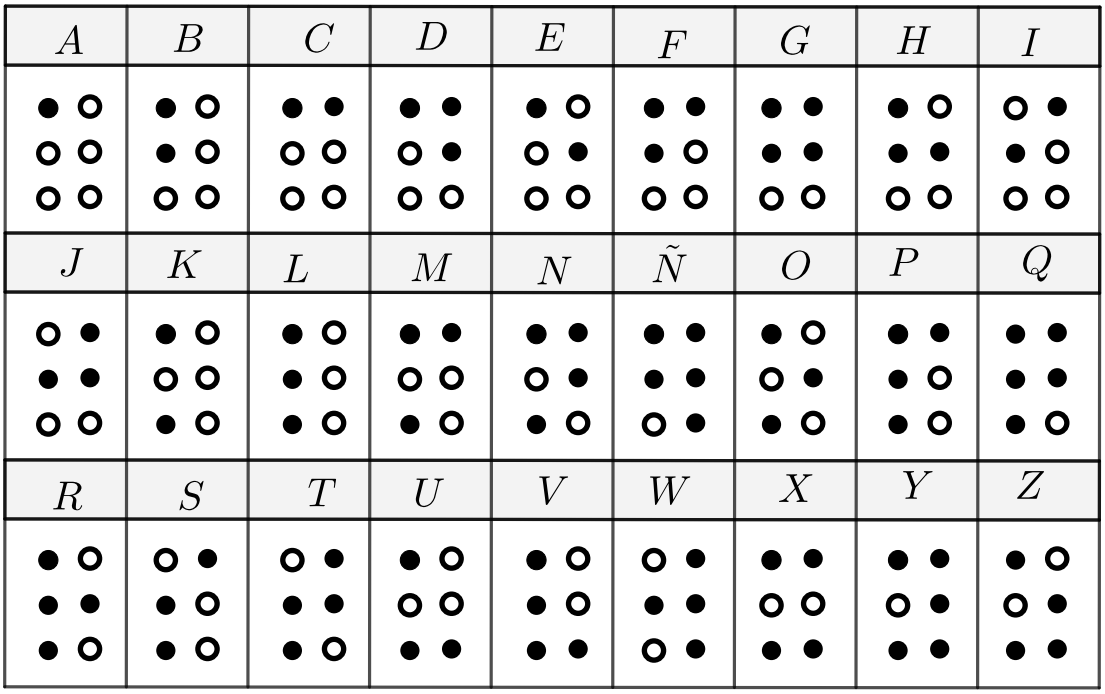


Figura 1.2: Alfabeto Braille

Sin embargo, el sistema Braille queda sujeto al tacto; hoy en día, es de importancia fundamental poner énfasis en las herramientas auditivas pues son la puerta a la tecnología de las personas ciegas. Más adelante se hablará al respecto.

1.5.2. Algunas otras herramientas

A continuación se mencionan algunas herramientas con las que actualmente cuentan los ciegos para tener accesibilidad al estudio.

- **Ábaco:** Los ábacos son dispositivos que permiten realizar cálculos matemáticos mediante el uso de cuentas que se desplazan por varillas. Esta herramienta es usada por los ciegos para realizar operaciones aritméticas y otros cálculos. Generalmente, la adaptación que se les hace es poner una tabla paralela a las cuentas con tela rugosa, para que no se muevan tan fácilmente, sino con el uso de las manos. Sirve mucho en el aprendizaje de la aritmética pues se pueden hacer todas las operaciones básicas.

- **Audiolibros y sistemas de sonido:** Los audiolibros son grabaciones de lecturas en voz alta. En otros casos, se usan lectores en voz alta que pueden leer casi cualquier tipo de texto. Un ejemplo, es el sistema de sonido Arithmomètre, que se adaptó en el siglo XIX, cuya versión se realizó mucho antes con Leibniz en 1694, servía para

enseñar aritmética emitiendo un sonido diferente para cada número, realizando las operaciones básicas.

- **Relieves y maquetas:** Las maquetas y relieves en tres dimensiones permiten explorar y comprender el mundo con el sistema háptico. Se ha tratado de enseñar de esta manera las figuras geométricas, formas y geometría a los ciegos.

- **Tecnologías de asistencia:** Las tecnologías de asistencia para ciegos incluyen herramientas como lectores de pantalla [2], programas de reconocimiento de voz y aplicaciones móviles diseñadas específicamente para personas con discapacidad visual. Además de contar con lectores de imágenes que describen el contenido de las mismas en voz alta.

A pesar de existir innumerables avances en las herramientas y la accesibilidad para los ciegos, las matemáticas todavía se encuentran relegadas, no se ha avanzado más en tratar de hacer accesible los conceptos fundamentales de las matemáticas.

Actualmente, existe software auditivo que trata de enseñar ciencia a los ciegos, por ejemplo el *AudioLink*, el cual consiste en un juego de rol (RPG), donde el jugador tiene que pasar una serie de escenarios para cumplir misiones: en ellas aprende conceptos de ciencia [7]. Cuenta con una interfaz gráfica de juego de 8 bits, y trata de mostrar el escenario a partir de los ruidos que se podrían encontrar en ese lugar, por ejemplo, el agua, el aire, sonidos de animal, etc. Lo interesante, es que al ser un juego, la forma de aprender es diferente a la tradicional, pues entra en otra dinámica más didáctica a la hora de aprender. Esta es una de las cualidades que se resaltarán más adelante en el desarrollo del software.

1.6. ¿Por qué Teoría de Conjuntos?

Como hemos visto falta un gran camino en el acercamiento de las matématicas profundas o más abstractas a la enseñanza de las matemáticas, en particular, a las personas con discapacidad. Teoría de conjuntos aparece como una oportunidad de poder acercar temas como grupos, anillos o campos, es decir introducir al usuario al algebra moderna y a otros temas de interés. Es por ello que aparece como un tema introductorio que abre las puertas a un mundo más complejo.

La teoría de conjuntos es una rama de las matemáticas que se ocupa del estudio de los conjuntos, sus propiedades y sus relaciones. Es un pilar fundamental en las matemáticas modernas y el pensamiento lógico-matemático. A continuación se enlistan algunos factores que hacen importante el dominio de este tópico en distintos ámbitos:

[2]Es importante resaltar que de esta manera los ciegos aprenden a manejar la computadora, tanto en sistemas operativos comerciales como libres; es totalmente auditiva. Los lectores de pantalla describen el sistema como si fuera una hoja de cálculo

- **Fundamentos para entender el razonamiento matemático:** Muchas ramas de las matemáticas tienen sus bases en la teoría de conjuntos, sino es que todas utilizan su lenguaje como una herramienta fundamental. No se puede entender matemáticas sin un andamiaje en el aŕea.

  En específico, el entendimiento de las operaciones básicas con los distintos campos de las matemáticas, por mencionar algunos ejemplos:

  - **Operaciones de conjuntos**: En álgebra se ocupan las operaciones básicas de teoría de conjuntos para manipular expresiones algebraicas como sumas y diferencias de términos.

  - **Conjuntos y funciones**: Una función es una regla de correspondencia que asigna a cada elemento del dominio un elemento de conjunto llamado rango. En este contexto, la teoría de conjuntos proporciona un entendimiento claro, participa en la definición, clasificación y manipulación de funciones.

  - **Sistemas de ecuaciones**: La teoría de conjuntos se utiliza para definir las familias de soluciones del sistema; un conjunto de funciones que satisfacen la ecuación diferencial.

  - **Estructuras algebraicas**: Ya sea teoría de grupos, anillos, campos, espacios vectoriales muestran las relaciones abstractas y propiedades de un conjunto de elementos. Por ejemplo, un grupo es un conjunto de elementos junto con una operación binaria que combina dos elementos del grupo. Los grupos son una herramienta fundamental en la teoría de la simetría, temas de conjuntos disjuntos o conjuntos. Se utilizan en muchos campos de las matemáticas, física e ingenierías, electrónica digital, teoría de la información, entre otros.

- **Álgebra:** La teoría de conjuntos tiene todas las propiedades de un álgebra de Boole. Es decir, el álgebra moderna se encarga del estudio de las propiedades abstractas de un conjunto matemático: en primera instancia está el grupo, luego el anillo, grupo, campos, espacios vectoriales y por último un álgebra.

- **Geometría y topología:** La teoría de conjuntos se ocupa del estudio de formas, las propiedades y las relaciones espaciales. Se puede utilizar para definir las propiedades de los conjuntos geométricos, como los puntos, las líneas, y las figuras tridimensionales.

- **Desarrollo del pensamiento lógico-matemático:** Al estudiar la teoría de conjuntos se desarrollan habilidades para analizar y resolver problemas de manera sistemática.

- **Aplicaciones informáticas y programación:** Esta área se ocupa de la teoría de conjuntos para el diseño de algoritmos y la programación. Es la base para el desarrollo de lenguajes de programación y la creación de sistemas informáticos complejos. También es importante en la lógica matemática, que se utiliza para la verificación de programas y la seguridad de datos.

- **Análisis de datos y estadística:** Se utiliza para clasificar y agrupar datos, identificar patrones y relaciones entre conjuntos de datos y realizar operaciones de probabilidad.

- **Física y matemáticas aplicadas:** La teoría de conjuntos se usa en muchas áreas de la física, como la mecánica cuántica, la teoría de la relatividad, la teoría del caos y la teoría de los sistemas dinámicos. También es usada para modelar sistemas complejos en biología, economía y otras áreas de las ciencias sociales.

- **Vida cotidiana:** En la vida cotidiana podemos utilizarla para todos los eventos que estén relacionados con el razonamiento lógico.

En resumen, la teoría de conjuntos es una herramienta poderosa y versátil que se utiliza en muchas áreas de la ciencia, la tecnología y la vida cotidiana. Su comprensión es esencial para quien necesita trabajar en campos que requieren habilidades matemáticas avanzadas y para aquellos que desean desarrollar habilidades de pensamiento lógico y análisis. En concreto, es una apertura para el aprendizaje abstracto que se debe trabajar con las personas ciegas.

1.7. Conclusión

Como modelador matemático uno debe de ser capaz de analizar un problema y proponer modelos, entender el comportamiento de los fenómenos desde una perspectiva científica que permita cambiar la realidad. De esta manera el problema de la accesibilidad del conocimiento y su análisis gana terreno en el campo académico.

La importancia de la accesibilidad de todo el conocimiento es fundamental para evitar la discriminación epistémica, dada por la falta de instrumentos diseñados para el aprendizaje de personas con alguna discapacidad. Debemos entender cómo es la aprehensión de la realidad de las personas que no han visto el mundo. Por ejemplo, el paso de los cuantificadores universales no es tan directo cómo podría llegar a ser el de una persona normovisual, pues existen diversas maneras de interactuar con el mundo, los ciegos conocen elemento por elemento y posteriormente generalizan. Aquí es, se considera importante enseñar conceptos abstractos de las matemáticas tanto en la vida cotidiana como en la escolar y profesional, pues el razonamiento lógico siempre será útil.

De esta manera, a lo largo de la investigación se trabaja una propuesta de aprendizaje para teoría de conjuntos que se analiza desde varios modelos matemáticos.

Un factor de discriminación en la educación se debe a la poca accecibilidad y falta de herramientas del aprendizaje que tienen las personas ciegas o con baja visión; si hubieran más herramientas, bajos costos y más divulgación de lo que ya hay, se podría ir borrando la barrera del aprendizaje, la tan marcada forma institucionalizada de aprendizaje normovisual. Es de trascendental importancia abrir caminos del conocimiento y aprendizaje. En el siguiente capítulo se plantea una propuesta incluyente de aprendizaje en teoría de conjuntos.

Capítulo 2

Diseño de un modelo de aprendizaje asistido computacional

El secreto del maestro es saber reconocer la distancia entre la materia enseñada y el sujeto que instruir, como así también la distancia entre aprender y comprender.

Jacques Ranciere, *El maestro ignorante*

La prueba se realizo en 2 dias diferentes, en el primero se hizo la unión-diferencia y en el segundo diferencia-complemento.

Introducción

Como se vio en el capítulo anterior, hay un largo camino por recorrer en la accesibilidad del conocimiento de las matemáticas en los ciegos (y otros sectores), y sobre todo, hacer un modelado del aprendizaje para ver cuáles son las actividades que hay que enfatizar o simplemente comprobar si realmente se está cumpliendo el objetivo de enseñar. En ese sentido, en este capítulo se plantea el plan de trabajo que se llevó a cabo en el diseño de la aplicación de la enseñanza de teoría de conjuntos. En específico, se centra en el contenido de la aplicación, es decir, responde a la preguntas: ¿Cómo se diseñó el contenido de la aplicación? ¿Qué factores se tomaron en cuenta y por qué se planteó de esa forma?

Para responder a las preguntas anteriores, se hace una breve revisión de la taxonomía de Bloom para hacer un enlace con el contenido de teoría de conjuntos, de tal manera que sea una coyuntura al desarrollo y estructura de la aplicación.

2.1. Metodología de proyecto

Un tópico a resolver fue la forma de aprehender el mundo de las personas ciegas. Por ello, a lo largo del proyecto, se tomó como base un artículo para el diseño de aplicaciones en personas ciegas, llamado *Una metodología para desarrollar y evaluar la usabilidad de entornos virtuales basados en audio para el aprendizaje y la cognición de usuarios ciegos* [12], en donde se plantea una estrategia para abordar el desarrollo del software dirigido a esa población. Menciona las cosas que se tienen que tener en cuenta a la hora de pensar en una herramienta para ellos.

El proceso de la investigación consiste en dos etapas que se subdividen, como se muestra en el cuadro 2.2. El capítulo se enfoca en explicar la primera etapa del contenido: por qué se dividió de esta manera las operaciones de teoría de conjuntos y cuáles fueron las pautas de medida de las variables. Se aborda brevemente sobre la implementación y la validación, aunque esta sección se ve más detalladamente en el siguiente capítulo. De esta manera la primera etapa se revisa en los capítulos dos y tres. Por otra parte, la segunda etapa que tiene que ver con la usabilidad y el impacto cognitivo, se revisa en el capítulo del análisis estadístico.

Cabe resaltar que la tabla 2.2, es el esquema básico de la investigación. La primera etapa se refiere al desarrollo global del software, desde su planteamiento conceptual, hasta las pruebas y la recopilación de datos, mientras que la segunda etapa tiene que ver con el análisis estadístico de los datos recopilados. Las secciones de validación y a la segunda etapa están bastante ligadas; sin embargo una es la aplicación y la otra el análisis de la aplicación. En la primera se evalúan habilidades, como son: estructura espacial, memoria abstracta, memoria espacial, memoria de corto plazo, percepción háptica, colaboración, resolución de problemas, habilidades matemáticas, orientación y movilidad, estructuras

Etapas	**Definición de los pasos metodológicos**	**Contenido** **Método**	**Aplicación** **Herramienta**
La primera etapa va desde el prediseño, hasta la implementación.	Análisis: Es el pre-uso, donde se definen las habilidades cognitivas a definir y las herramientas con las que se va a trabajar.	Se analizó la Clasificación de Bloom, para diseñar los niveles de enseñanza de teoría de conjuntos.	Se tomaron las operaciones básicas de teoría de conjuntos y se llegó a un consenso de usar Python.
	Diseño: Es parte de la construcción de la aplicación, se define el entorno virtual y las tareas cognitivas a realizar.	Se ordenaron las operaciones respecto a la dificultad que representan.	Se dividieron en cuatro niveles y a su vez en otros cuatro subniveles.
	Implementación: Prueba de usuarios con retroalimentación y cambios en el prototipo.	Se tuvieron varios prototipos que fueron puestos a prueba.	Se crearon más adaptaciones al software.
	Validación: Realizar actividades para evaluar la cognición y la usabilidad, se pasa al uso.	Retroalimentación de los conocimientos adquiridos en la sección.	Revisión de los datos.
La segunda etapa revisa la efectividad del software	Usabilidad: Responde a la pregunta ¿a qué usuarios va dirigido?	El programa puede ser usado a partir de primaria en adelante.	Análisis estadístico.
	Impacto cognitivo: En este apartado se evalúan las habilidades cognitivas que se adquieren en el uso.	En esta sección se hizo un análisis para poder medir el impacto cognitivo.	

Cuadro 2.2: Metodología para desarrollar la aplicación

cognitivas, tiempo-espaciales, habilidades de lenguaje, navegación compleja. En la segunda se analiza qué tanto se aprendió de las habilidades, cuales fueron más fáciles o difíciles y en general cuál fue el camino de aprendizaje.

Por otra parte, el plan de trabajo de toda la investigación se puede visualizar en la imagen 2.1, en la cual se detalla más sobre el análisis estadístico. Dicho lo anterior, en la siguiente sección se inicia el análisis de pre-uso.

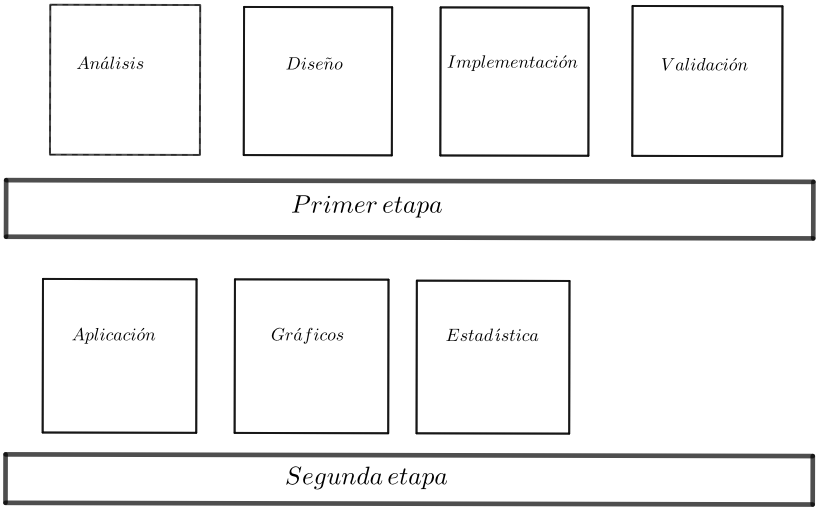


Figura 2.1: Plan de trabajo

2.2. La taxonomía de Bloom una propuesta a la era digital

Tratar la aprehensión del conocimiento con una taxonomía permite comprender cómo hacer y diseñar las actividades e inclusive saber si cumplen sus objetivos. Por ello, en el lapso de la elección del tema, fue de gran importancia poder diseñar de forma didáctica la enseñanza de operaciones de conjuntos. Para ello se tomó como referencia la taxonomía de Bloom. ¿Por qué esta clasificación? A pesar de no ser una teoría dirigida a las tecnologías de información y comunicaciones (TIC's); sino al entendimiento de las operaciones mentales o cognitivas; plantea un escenario favorable para la enseñanza digital; adquisición, profundización y creación de conocimiento. Conviene subrayar, que Bloom toma tres aspectos para los objetivos educativos, a saber: el cognitivo, el afectivo y el psicomotor. No obstante, sólo se hablará del aspecto cognitivo, que conlleva los procesos de información y habilidades mentales.

La taxonomía de Bloom es una clasificación de los procesos de adquisición del aprendizaje que se utiliza para clasificar diferentes objetivos de la enseñanza y establecer niveles de complejidad dentro de las actividades pedagógicas. Fue desarrollado por Benjamín Bloom y un equipo de educadores en 1956. No obstante, en años recientes ha tomado mayor interés debido a la inserción de TIC's como medio de diseño de software. En 2001 Anderson y Krathwohl iniciaron sus investigaciones respecto a esta taxonomía, empezaron por cambiar los sustantivos por verbos, pero no fue sino hasta 2009 que le dieron otro nombre: Taxonomía de Bloom para la era digital.

Originalmente, consta de seis niveles de complejidad cognitiva, que se presentan en orden jerárquico, desde los niveles más básicos hasta los más complejos, es decir, en forma pi-

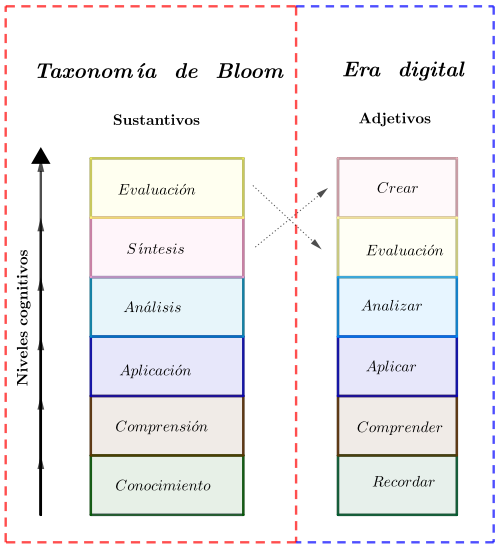


Figura 2.2: Taxonomía de Bloom

ramidal. Lo anterior, no supone que el aprendizaje se dé de manera tajante y jerárquica, sino que marca la complejidad en los procesos mentales. Los seis niveles se muestran en la figura 2.2, del lado izquierdo son sustantivos que fueron descritas por Bloom, mientras en el derecho, se observan los verbos que generalizan en la era digital [1]. A continuación se explica con más detalle cada nivel y después se hace énfasis en la era digital y por qué tiene sentido usarla para teoría de conjuntos.

1. **Conocimiento**: Se adquiere información y conocimiento básico sobre un tema. Esto incluye recordar información, como definiciones, términos y hechos. Básicamente, se puede traducir a conocer que existe algo y obtener un lenguaje mínimo.

2. **Comprensión**: Se demuestra una comprensión más profunda del material, cómo explicar ideas o conceptos en sus propias palabras y hacer conexiones entre diferentes ideas.

3. **Aplicación**: Se aplica el conocimiento adquirido a situaciones prácticas o teóricas. Esto puede incluir la solución de problemas, la aplicación de fórmulas o la realización de tareas específicas. Este proceso supone más dificultad porque se necesitan tener claros los niveles anteriores.

4. **Análisis**: Se descompone el material en partes para entenderlo mejor. Esto incluye la identificación de patrones, la organización de información y la identificación de relaciones entre diferentes partes.

5. **Síntesis**: Se combinan diferentes ideas o partes del material para crear algo nuevo.

Esto puede incluir la creación de un nuevo producto, la formulación de una hipótesis o la escritura de un ensayo.

6. **Evaluación**: Se evalúa el material para hacer juicios o conclusiones. Esto puede incluir la evaluación de la calidad de un argumento o la comparación de diferentes perspectivas sobre un tema.

En términos generales, la taxonomía de Bloom se describe arriba (los sustantivos) ¿A qué nos referimos con Bloom en la era digital? A partir del 2009, Churches propuso una nueva vinculación con la TIC's, como se observa en la figura 2.2, de lado derecho hay verbos; ésta última es la nueva variación; notable el cambio de orden entre la evaluación y la síntesis, ya que en la era digital crear implica una mayor dificultad. Por ejemplo, crear un programa. Estos verbos representan en general, la actividad que se debe dominar respecto a un conocimiento de uso tecnológico. Otra de las grandes aportaciones que hizo Churches fue hacer una vinculación de los verbos con las habilidades del pensamiento. A continuación se explican: la adquisición de conocimiento, la profundización del conocimiento y la creación del conocimiento, cuya dificultad va aumentando gradualmente a como fueron mencionados. En la tabla 2.3, se observa con qué adjetivo está sujeto cada habilidad.

Asimismo, en la tabla 2.3 se acomodar los contenidos de teoría de conjuntos en el software, usando la teoría anterior:

Habilidades adquiridas	**Verbo**	**Actividad de teoría de conjuntos**
Adquisición del conocimiento	Recordar	Aprender las definiciones.
	Comprender	Ejemplos de la vida cotidiana.
Profundizar el conocimiento	Aplicar	Realizar ejercicios para fortalecer los ejemplos.
	Analizar	Realizar ejercicios que enseñan diferentes acepciones de concepto.
Creación de conocimiento	Evaluar	Retroalimentación del programa.
	Crear	Poder abstraer el concepto a otras actividades.

Cuadro 2.3: Análisis de Bloom para aprender teoría de conjuntos desde una perspectiva digital

De esta manera, la tabla pretende ser una guía en la estructura del aprendizaje de las operaciones básicas.

2.3. Análisis del contenido

Como se vio en la sección anterior, se deben tomar varios factores en el modelo de enseñanza digital. En la figura 2.4, se muestra la organización del contenido, como se observa se usó la clasificación de Bloom para garantizar el dominio del contenido. Es decir, la tabla es el análisis de la estructura general que se debe plantear en las cuatro operaciones: unión, intersección, diferencia y complemento [1].

Pero antes de continuar, es importante responder a la pregunta ¿Por qué enseñar estas cuatro operaciones por separado y no, por ejemplo, tomar la diferencia como una operación adjunta al complemento? ¿Por qué seguir un orden a la hora de enseñarlas? De alguna manera, también se trató de simplificar las operaciones, verlas como acciones, es decir, la unión colectar, la intersección quitar; por otro lado, las restantes operaciones necesitan de estás primeras, de ahí que se consideren más difíciles. Todas estas proposiciones son supuestos, que se concretarán más adelante. Una de las hipótesis, que se analizarán en el capítulo cuatro, es que la unión es la operación más sencilla cognitivamente hablando; sin embargo, en las otras operaciones no ocurre de esta manera.

La diferencia, por su parte, se consideró como la segunda operación más difícil, pues no sólo se trata de colectar sino de separar elementos que no están dentro de la intersección. En ese sentido, es una operación que requiere no sólo juntar una propiedad, sino discriminar elementos. Por otra parte, la diferencia es el complemento de la intersección y, a tal efecto, podría entenderse como un anexo del complemento; sin embargo, fue separada para introducir la operación que se consideró como la más difícil cognitivamente, es decir, se le le hicieron sus propios niveles. Por último, el complemento es una operación donde se tiene que tomar todo lo anterior en cuenta. Es por ello que se enseña hasta al final del programa. Cabe destacar que esta división se retomará en el análisis estadístico, pues será una de las variables a analizar. Por último, destacar que estas conjeturas se tomarán en cuenta en la interpretación de los gráficos.

Figura 2.3: Recordar y comprender en la didáctica

[1] Estas operaciones se enseñan por separado aunque algunas se pueden contruir a partir de las otras

2.4. Diseño del contenido

A continuación se expone cómo se entrelaza el cuadro 2.3, con las actividades de cada operación :

- **Recordar y comprender:** Esta sección es llamada *didáctica*, en la figura 3.2 sección de aprendizaje, es donde se encuentran las definiciones de cada operación. En la figura 2.3 se muestra la división de este apartado. La definición sería el apartado de recordar; mientras, en los ejemplos cotidianos, se buscaron los que fueran asequibles a la vida de los ciegos, se vería la forma de comprender. La sección de didáctica se encuentra en todas las operaciones, tienen la misma estructura respecto al contenido.

- **Aplicar:** Después de los ejemplos de la didáctica, empieza el primer acercamiento a la aplicación de la operación, también se puede ver en la figura 2.3. No obstante, el primer nivel de cada operador refuerza esta parte, donde sólo se debe realizar sin ver algún matiz en especial de la definición.

- **Analizar:** En la imagen 2.4, se puede ver la división de cada nivel en cada operación. Los últimos tres niveles, sirven para analizar las diversas acepciones de cada concepto.

- **Evaluar:** Después de realizar los ejercicios, se da una retroalimentación. Empero, se puede entender que es pasiva, pues el usuario no realiza la acción de evaluar. El propósito de la evaluación es dirigir al usuario, no se le da una calificación en concreto, sino que se le ofrece un comentario que le sugiere qué camino seguir; es decir, volver a estudiar la didáctica, repetir el nivel o continuar al siguiente. Aunque esta evaluación es un tanto pasiva, el usuario tiene la posibilidad de ser autocrítico y evaluarse a sí mismo. En ese sentido es que el verbo evaluar toma sentido.

- **Crear:** El programa no llega a este apartado en sentido estricto. Sin embargo, el objetivo de todo el programa es que el usuario sea capaz de poner en práctica lo aprendido, desde la forma de hablar, hasta la de razonar. Dicho lo anterior, en la sección de estadística también se hace un análisis para corroborar este punto.

Después de entender lo que deben contener las operaciones, ahora se explica brevemente cómo se divide el contenido de las operaciones. En la figura 2.4 se muestra la división de las operaciones y enfrente de cada una se resume el contenido de cada operación, el cual acaba siendo un nivel. En total son 16 niveles, cuatro por cada operador.

Una problemática a la hora de enseñar conjuntos es que se piensa en diagramas de Venn, que al ser tan visuales, resulta más difícil comprender el concepto. De esta manera, el programa usa conjuntos de números, en figura 2.4 se puede ver un ejemplo, donde obtenemos diferentes resultados al operar con los conjuntos. Algo interesante a resaltar es que, trabajar de esta manera es ir conociendo el conjunto desde sus elementos hasta el conjunto

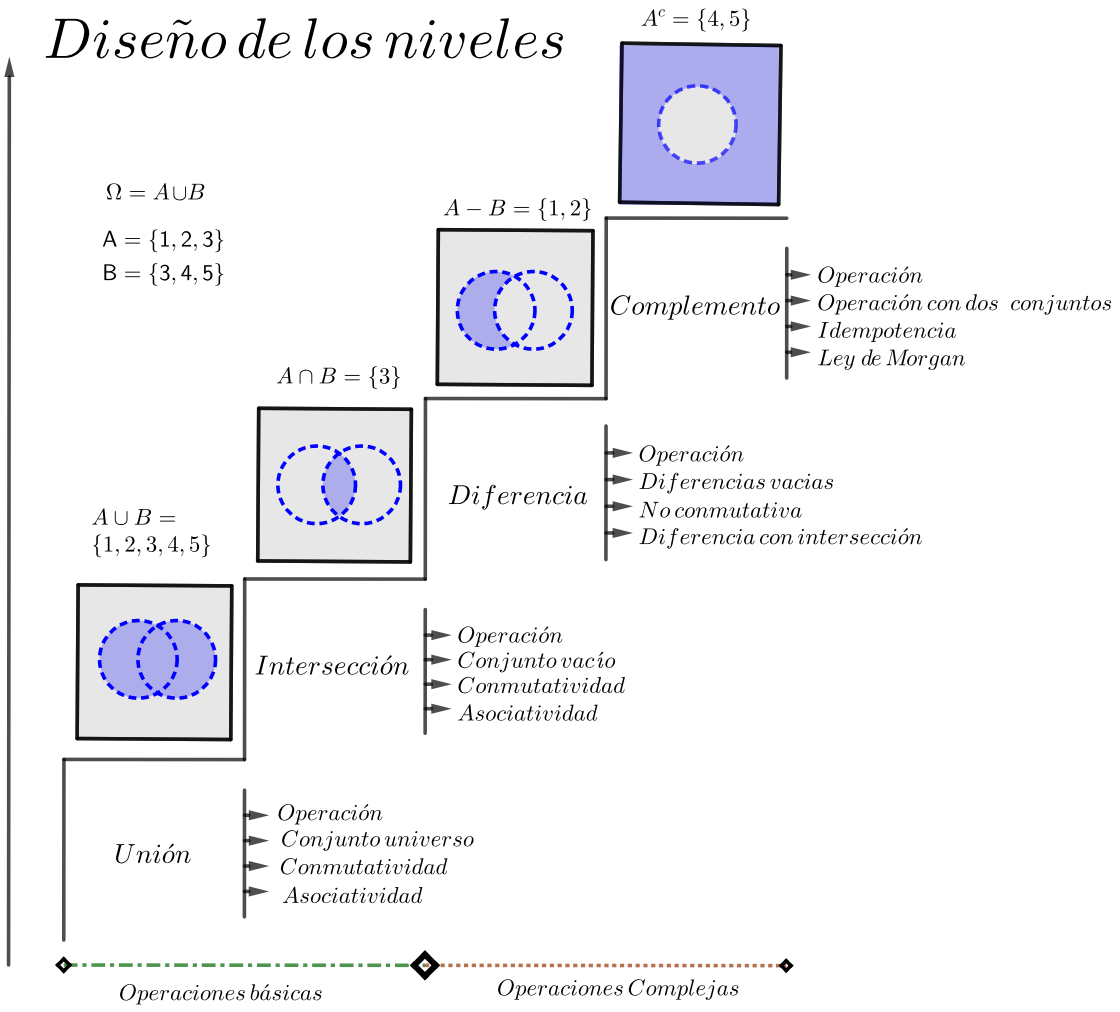


Figura 2.4: Diseño del contenido de los niveles de la unión, intersección, diferencia y complemento

universo, *i.e.*, se aprende desde lo particular hasta lo universal. Imaginemos una hoja de cálculo, donde para cada casilla corresponde una coordenada; cuando un ciego aprende el patrón de las mismas después de ir tecleando una por una, generaliza el patrón: es un razonamiento inductivo que después se generaliza. De la misma manera que una hoja de cálculo, se puede imaginar los conjuntos; en cada fila hay un conjunto y en cada columna un elemento. Puede que haya casillas finitas o infinitas, pero entenderlo de esta manera, permite dar un orden a los conjuntos, a pesar de no existir como tal en los elementos.

Lo anterior, llevó a diseñar dos formas en las que pudiera interactuar el usuario con la pregunta que se le realiza. Una fue llamada *exploración* y la otra *repetición*, en el siguiente capítulo se hablará de su diseño de software. Es importante destacar que la exploración fue pensada para el razonamiento inductivo, el reconocimiento de elemento por elemento; mientras que la repetición vuelve a mencionar toda la pregunta.

Por otro lado, también se consideró dividir en dos bloques las operaciones, a saber: básicas y complejas. Dado que las segundas requieren dominar las primeras, puesto que, es más sencillo para la mente juntar y separar que hacer estas acciones con ciertos matices, como ya se mencionó con anterioridad.

2.5. Conclusión

Se implementó el programa por primera vez con usuarios de nacimiento ciegos, quienes retroalimentaron la aplicación. El grupo que usó el software fue de ciegos adquiridos, es decir, personas que en alguna etapa de su vida perdieron la vista; algunos desde el año de vida y otros desde su adolescencia o en la adultez. Además, su nivel de estudio fue diferente, desde la secundaria hasta estudios universitarios. Una cualidad en común fue que todos tenían algún antecedente con la computadora, a saber, clases de computación por medio de lector de pantalla: conocían el teclado. Esto se decribirá detalladamente en el siguiente capítulo, al igual que la validación.

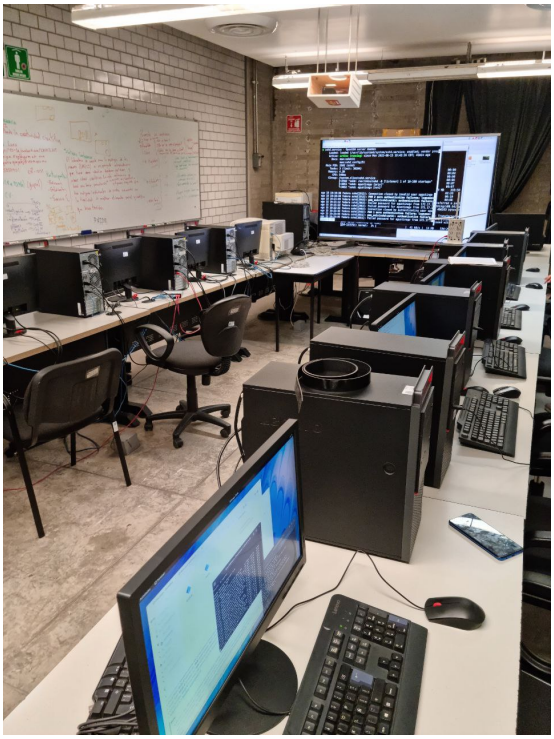
Se vieron varias cosas en este capítulo. Resaltando las más importantes: se planteó el plan de trabajo tanto del desarrollo de la aplicación como su investigación y análisis estadístico; se hicieron algunas conjeturas importantes acerca del aprendizaje de las operaciones básicas de teoría de conjuntos y se dieron las bases teóricas para el diseño conceptual de la aplicación. Por otra parte, también se dio una propuesta para introducir conceptos matemáticos que van más allá de teoría de conjuntos, a saber: algunas propiedades albebraicas

Capítulo 3

Desarrollo de un modelo de aprendizaje asistido de teoría de conjuntos

El Laboratorio de Cómputo para la Enseñanza de las Ciencias tiene como finalidad el aprendizaje y el desarrollo profesional de todo aquel interesado en el conocimiento.

LACECI, *Segundo taller de Tipografía Computacional*



La programación y el uso de la aplicación se realizo en LACE-CI y LAMAT.

Introducción

El presente capítulo tiene como objetivo explicar las diferentes etapas que se llevaron a cabo en el desarrollo del software del aprendizaje para la enseñanza de teoría de conjuntos y su modelado matemático. El software fue creado con el fin de ofrecer una solución eficiente y efectiva a la enseñanza de teoría de conjuntos en personas ciegas, además de crear un modelo matemático del aprendizaje, con base en un análisis estadístico. Se realizó un proceso de investigación, como se vio en el capítulo anterior y un análisis de los requisitos necesarios con la finalidad de diseñar y construir el software, *i.e.*, esta sección trata sobre el desarrollo del software (de la programación), tomando su análisis, diseño, implementación y validación.

En este capítulo se describen las etapas del proceso de desarrollo de software. Desde la concepción de la idea, ya incluyendo la taxonomía de Bloom, hasta la validación. Asimismo, en el siguiente capítulo se presentan los resultados y análisis de los datos obtenidos, mediante herramientas estadísticas, para ver las limitaciones y posibilidades de mejora para trabajos futuros.

Como se vio en el capítulo anterior, se construyó un arsenal teórico conceptual para poder hacer la división de los bloques de enseñanza; sin embargo, también hubo un momento de aprendizaje respecto a la programación. Es decir, hubo varios cambios en el proyecto desde su primera versión hasta la última. Aunque no es el objetivo general de la investigación, fue necesario poder hacer una aplicación que pudiera recompilar los datos y al mismo tiempo fuera el medio con el que estas aprendieran teoría de conjuntos.

3.1. Análisis de requisitos

En esta sección se mostrará el análisis de requisitos [1] necesarios para el desarrollo del software de manera resumida, pues no entra en el objetivo general de investigación. Sin embargo, aclara el panorama de la investigación y los problemas que se fueron presentando a lo largo del desarrollo, a continuación se enlista:

1. **Identificación de las partes interesadas**: El software para la enseñanza de teoría de conjuntos va dirigido a personas ciegas a partir de nivel secundaria que deseen aprender. Pero no se limita a eso, sino también a profesores que deseen implementar el software en sus clases de matemáticas o algún interesado fuera de la escuela, ya que esta área de las matemáticas es muy socorrida; inclusive se ve en todos los niveles escolares ya sea directa o indirectamente. Por otro lado, también puede ser utilizado por personas visuales.

2. **Recopilación de información**: A lo largo de la programación se fue con estudiantes y profesores de 'Letras Habladas' de la Universidad Autónoma de la Ciudad de México. Asimismo, se tuvo contacto con los estudiantes de la clase de Informática o Computación para ciegos que se imparte en el Sistema Nacional para el Desarrollo

[1] Se habla brevemente de los requisitos para seguir el plan de trabajo que se propuso en el capítulo anterior

Integral de la Familia (DIF) de Coyoacán (especializado en personas con baja visión y ciegas) y con algunos especialistas en la enseñanza de personas ciegas. El propósito fue reunir información sobre las características y funcionalidades que debe tener el software, no el contenido sino las herramientas. Hubo diversas modificaciones en el proyecto, como se verá más adelante. Respecto al contenido, como se menciona en el capítulo anterior, se hizo un investigación sobre el contenido.

3. **Requisitos funcionales**: Necesidades que se tienen para enseñar y entender el programa, tales como:

   ❖ Instrucciones: El programa cuenta con una explicación detallada de su manejo y funcionamiento, para que el usuario pueda interactuar de manera autónoma.

   ❖ Explicación de conceptos: Incluye una sección de aprendizaje, el cual contiene las definiciones formales, además de explicaciones detalladas para facilitar su comprensión y asegurar que quien interactúe con el programa entienda completamente el contenido.

   ❖ Actividades didácticas guiadas: Hay una sección en donde se enseña cómo se debe usar el software, instrucciones desde cómo ingresar sus respuestas hasta dar siguiente. Vienen ejemplos de la vida cotidiana y un poco más abstractos.

   ❖ Ejercicios de práctica: Los estudiantes tienen acceso a una variedad de ejercicios de práctica, que se organizan por operación y nivel de dificultad.

   ❖ Evaluaciones con retroalimentación: Con la finalidad de evaluar el progreso de los estudiantes y determinar su comprensión del material, es importante dar una retroalimentación de forma que él mismo pueda identificar sus avances.

   ❖ Recopilación de datos: Con una adecuada recopilación de datos, se puede obtener una comprensión de cómo está funcionando el software para así poder mejorarlo y adaptarlo cada vez más a las necesidades de los usuarios.

4. **Requisitos no funcionales**: Son aquellos que no tienen que ver con las funciones, sino el cómo se ve y cómo funciona:

   ◆ Interfaz de usuario intuitiva: Fácil de usar para todos los usuarios con audición.

   ◆ Seguridad: Los datos recopilados en las pruebas y uso del software únicamente tendrán uso académico y no serán compartidos en ningún caso.

5. **Priorización de requisitos**: Tiene preeminencia el aprendizaje de los usuarios

6. **Validación de requisitos**: Validan a los usuarios que lo realicen.

El análisis de requisitos es un croquis de acción sobre el software que se realizó. Una vez puesto en la mesa la estructura medular, a continuación se explica su implementación.

3.2. Diseño de la arquitectura

Después de haber hecho un análisis tanto del contenido como del software, a continuación se explica detalladamente su diseño de programación.

3.2.1. Modelo de componentes y servicios

El modelo de componentes y servicios para este proyecto se divide en dos capas, la interfaz y los datos. Cada capa se compone de varios componentes:

- **Interfaz**: Este software no está contemplado para tener una interfaz gráfica, pues no es necesaria. No cuenta con un sistema de ventanas y botones como la mayoría de otras aplicaciones o páginas web, más bien su navegación es por medio de comandos. Debido a que es un software dirigido a personas ciegas, se trabaja con Python3, se ejecuta un script con comandos de BASH en Linux que manda a llamar archivos de Python, aunque bien podrían llamarse desde otro programa de Python. Por otro lado, teniendo instalado Python3 en cualquier sistema operativo, se puede ejecutar cada operación por separado. En ese sentido, proporciona una ventaja accesible. De esta manera, el software se vuelve multiplataforma, ya sea en Linux, Windows, o en MacOS.

- **Datos**: El software genera una base de datos en formato *.csv*, que son un tipo de documento, hoja de cálculo, sencillo y ligero para representar datos en forma de tabla. Las columnas se separan por comas y las filas por saltos de línea. Todas las variables que se consideraron necesarias para el análisis estadístico se guardan automáticamente en el archivo. Con esto, se puede proporcionar una retroalimentación de lo aprendido al usuario y, por otra parte, se podrá revisar los resultados que está ofreciendo el software.

3.2.2. Tecnologías y herramientas utilizadas

- Veinte computadoras de escritorio Lenovo ThinkCentre M800, con un procesador Intel® Core™ i5-6500 x 4 y una memoria RAM de 8.0 GiB: Los dispositivos que se encuentran en el Laboratorio de Cómputo para la Enseñanza de las Ciencias (LACECI), parte de la Universidad Autónoma de la Ciudad de México, lugar donde se realizó el desarrollo y pruebas del software para la enseñanza.

- Python3: Es un lenguaje de programación muy versátil, flexible y de alto nivel que se enfoca en la simplicidad y la claridad del código, lo que lo hace más fácil de entender y mantener, pero sobre todo de código libre. Tiene una gran comunidad de

desarrolladores y usuarios que contribuyen constantemente con bibliotecas y paquetes de software. Esto genera la facilidad de que otras personas puedan integrarse en un futuro a contribuir en este proyecto. Cabe destacar, que una de las librerías de este lenguaje que se usó fue **soundplay**, la cual fue una valiosa herramienta para el manejo del audio. Dicho lo anterior, el programa fue escrito en este lenguaje.

♠ R: Es un lenguaje multiplataforma especializado en el análisis de datos y estadística, además de contar con herramientas de visualización de datos muy robustos. Permite manipular los datos y graficar adecuadamente para fines como los de este proyecto.

♠ Bash-Shell: Bash es un lenguaje utilizado para la automatización de tareas en sistemas como Linux, Unix, y MacOS. Se usará en esta ocasión para optimizar la ejecución de los códigos que componen el software completo.

3.2.3. Explicación de la estructura de la aplicación

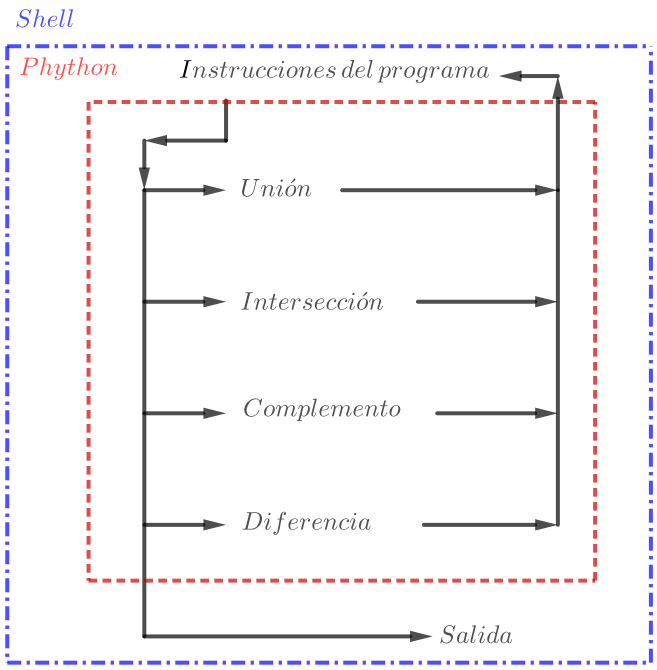


Figura 3.1: Estructura general del programa

La estructura de este software se divide en dos capas. La primera está en Bash mientras la segunda en Python. No obstante, ambas trabajan en paralelo. En la figura 3.1 se ve este diseño; si se hiciera un zoom en cada operación uno encuentra la estructura que se muestra en la figura 3.2.

La primera capa es un script realizado en lenguaje BASH-SHELL. Este automatiza la integración de los programas de las operaciones de teoría de conjuntos, uno por cada ope-

ración. La división del programa principal se puede ver en el gráfico 3.1, y el recuadro azul corresponde a dicha etapa. Esto es, se corre el *.sh*, el cual manda a llamar a alguna operación, después de terminar la ejecución de ese programa, vuelve a entrar al archivo *.sh*, así sucesivamente hasta que el usuario decida salirse.

La segunda capa, que corresponde al cuadro rojo de la figura 3.1, son los programas de cada una de las operaciones de la teoría de conjuntos, cuentan con una organización dividida en módulos. Cada operación tiene su bienvenida, instrucciones generales acerca del funcionamiento de este programa y un menú para el desplazamiento entre una sección de aprendizaje, de ejercicios de práctica y para la evaluación. Lo anterior se puede observar en el gráfico 3.2.

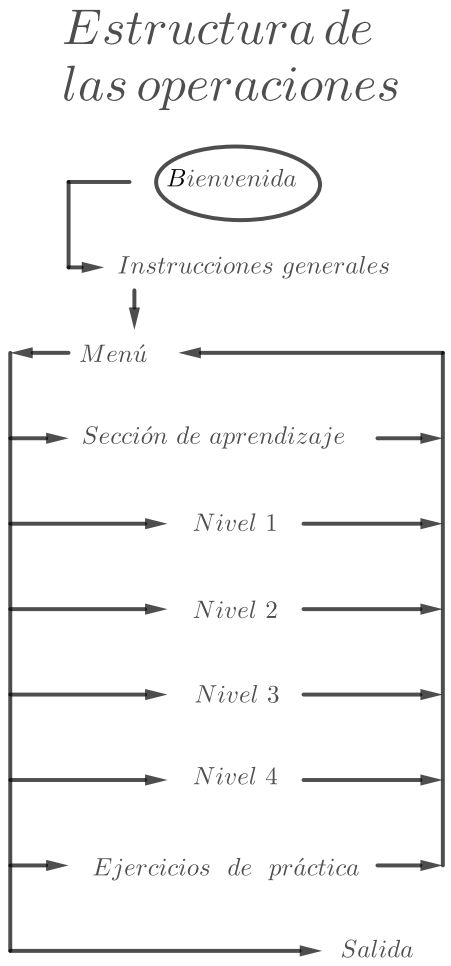


Figura 3.2: Estructura general de los niveles

Los módulos que componen al programa de cada operación se integran de la siguiente manera :

- Bienvenida: Se da un saludo al usuario, y se comenta la operación a la que se acaba de ingresar.

- Instrucciones generales: Se dan indicaciones para el correcto uso del programa y una definición formal de la operación que se está trabajando.

- Menú: Tiene la finalidad de indicar las diferentes secciones de aprendizaje, evaluación y práctica. Siempre que acaba un nivel se regresa a esta sección, en el mismo puede salir para ir a otra operación.

- Sección de aprendizaje: Se explica la definición formal de la operación en palabras sencillas, se dan ejemplos de su uso o aplicación de la operación con situaciones o cosas de la vida cotidiana, cabe destacar que se buscaron ejemplos que fueran accesibles a los ciegos, a saber, situaciones que fueran comunes para ellos. También se realizaron ejercicios de práctica guiados para que se fueran familiarizando con las evaluaciones.

- Evaluaciones: Al entrar a cada sección de evaluación se

da una explicación de las propiedades que se evaluaron, junto con un ejemplo de cómo se hace; posteriormente se realizaron 10 ejercicios con dos o tres conjuntos a operar completamente aleatorios, y por último se proporciona una retroalimentación al usuario de su desempeño, con la posibilidad de solicitar que se vuelva a realizar la evaluación si el resultado de ésta fue insuficiente.
Los datos generados en las evaluaciones son guardados en una base de datos .csv para su posterior análisis.

- Ejercicios de práctica: Los ejercicios de práctica tienen la finalidad de permitirle al usuario repasar la operación, ya sea porque el usuario quiere dominar la operación o por sugerencia de alguna de sus evaluaciones.

Con esto se puede asegurar que dentro de la estructura de cada operación que muestra la imagen 2.4 se cumple con los requisitos que se mencionaron en la sección 3.1, y se integran las partes estudiadas en el capítulo anterior.

3.3. Pruebas y verificación

En esta sección hablaremos de la metodología y el proceso que se llevó a cabo para probar el software hasta poder verificar que estaba libre de errores en la programación, que cumple con los parámetros de enseñanza vistos en el capítulo anterior y que es entendible para los usuarios para los que va dirigido.

3.3.1. Elección de la muestra de participantes para revisar el software

Primero se buscó un grupo de personas ciegas que al menos supiera la posición del teclado numérico, y de las letras: a, s, d, *, - y enter, dentro del teclado con una distribución QWERTY, en este caso fueron 18 participantes. Ya que, es necesario al menos tener este conocimiento, para filtrar el tipo de error del usuario. El nivel de estudio y edad son arbitrarias, pues es un grupo del DIF y con la dinámica que maneja el software, todos los usuarios pueden aprender o recordar lo ya aprendido. Es decir, es un grupo bastante heterogéneo respecto a sus conocimientos matemáticas. No obstante, el programa está diseñado para que esto no sea un impedimento, pues la teoría se enseña a partir de situaciones cotidianas .

Debido a que la población de personas con discapacidad visual que cuentan con dominio de la computadora y que estén interesadas en el aprendizaje de las matemáticas es re-

ducido, además de las limitaciones que muchas de estas personas tienen para acercarse a realizar las pruebas, se reclutaron un total de 18 estudiantes, pertenecientes a los talleres de Informática para personas con discapacidad visual impartidos en el DIF unidad Coyoacán, CDMX, y al programa de Letras Habladas parte de la Universidad Autónoma de la Ciudad de México.

Para poder realizar las pruebas fue necesario contactar al encargado de impartir los talleres de Informática para personas con discapacidad visual impartidos en el DIF unidad Coyoacán , con el Sr. Aurelio Hernández Trejo, a quien se le explicaron las intenciones y la necesidad de realizar estas pruebas con un grupo que tiene a su cargo. Después de revisar los tiempos y la disponibilidad de las personas que participarán en la prueba, se solicitó el apoyo del DIF con un medio de transporte para el traslado desde el DIF unidad Coyoacán hasta la Universidad Autónoma de la Ciudad de México, plantel San Lorenzo Tezonco. Puso a disposición un autobús con una capacidad de pasajeros suficiente para el traslado de los estudiantes del taller. Dicho autobús consiguió entrar hasta el laboratorio donde se realizaron las evaluaciones, gracias al permiso y atención otorgados por parte de la coordinación de la Universidad Autónoma de la Ciudad de México.

En la parte de la Universidad Autónoma de la Ciudad de México, se realizó una invitación a los miembros del programa de Letras Habladas que estuvieran interesados en realizar estas pruebas, para que pudieran hacer un espacio de tiempo entre sus clases y así asistir a las evaluaciones.

3.3.2. Condiciones de la evaluación final

Las pruebas se realizaron en el Laboratorio de Cómputo para la Enseñanza de las Ciencias (LACECI) los días jueves 9 y viernes 10 de febrero del 2023. El grupo de personas ciegas ejecutó el software bajo supervisión, además de profesores y otros estudiantes del laboratorio LACECI , para resolver cualquier duda o inconveniente. Se les explicó el propósito de la evaluación y se les proporcionó una breve explicación sobre el uso del software.

El primer día de testeo del software se solicitó que los 18 usuarios realizaran las primeras dos operaciones, unión e intersección, y el segundo día, diferencia y complemento. Los participantes demoraron un promedio de dos horas por operación, y se les permitió tomar descansos si así lo requerían. Además de un receso de unos 20 minutos cada día después de terminar la primera operación correspondiente a ese día, para tomar un refrigerio.

En esta parte, es necesario señalar que, los usuarios fueron acompañados por familiares y/o tutores para realizar estas pruebas. Mismos que también fueron informados de las condiciones de estas evaluaciones, y estuvieron pendientes en todo momento de la realización de éstas.

3.4. Datos

3.4.1. Variables que se recopilaron en las pruebas

Se hablará brevemente de la recopilación de datos, con la finalidad de entender las variables que se ocupan en el siguiente capítulo. Primero se identificaron todas las variables de datos que se consideraron importantes de medir para analizar los objetivos del aprendizaje de acuerdo a lo visto en el capítulo anterior, ya que hay que hacer un análisis para poder darles una interpretación. Esto incluye información sobre las respuestas a las preguntas, el tiempo de respuesta, el número de errores, etc. En la tabla 3.1 se desglosan las variables obtenidas, junto con una breve explicación de cómo funcionan.

Del cuadro anterior, se pueden en realidad quitar todas las variables de acumulación ya que son datos repetidos. Se pensó que era importante ponerlas para facilitar los cálculos en el análisis de varianza. Por otro lado, también se despreció la variable de exploración ya que no fue usada por los usuarios de manera contundente, es decir, a lo más dos personas lo usaron. Tres variables son las que se van a ocupar: el tiempo total en contestar la pregunta, el número de repeticiones y el número de errores. Son suficientes para tener un estudio exploratorio de la funcionalidad del software.

3.4.2. Integración de las bases de datos

La recopilación de datos tiene que ser una tarea muy meticulosa y automatizada que permita fácilmente el manejo de los datos. En esta sección se explica brevemente la logística de los datos. Cada que un usuario termina un nivel, se genera una base de datos; en total por persona se crean 16 archivos, de los cuales se dividen en carpetas según la operación que corresponda. Se hizo una distribución dada por los nombres, como se ve en la tabla 3.2, donde la 'n' es el identificador del usuario o el id. Los archivos se guardaron en un mismo directorio y gracias a los patrones en los nombres se pudieron importar adecuadamente para darles su respectivo tratamiento.

Con los patrones anteriores, se pueden juntar todos los archivos en una sola carpeta. A la hora de importarlos, sólo se necesitó usar el patrón correcto para extraer toda la información necesaria.

3.4.3. Limpieza de datos

Organizar los datos como se muestra en la tabla 3.2 hace que los niveles y los usuarios no se mezclen: de esta manera, se pueden trabajar los niveles y operaciones sin que se confundan. Así, se revisaron todas las bases de datos, que nos dieron un total de 288; archivos, fue necesario contabilizar que estuvieran completas y que no se hubiera generado ningún error en la captura de los datos. Los datos se trabajaron con R-Studio; en el anexo B se puede encontrar el código que se utilizó para trabajarlas. La idea básica de cómo se tuvieron que acomodar, fue la siguiente:

Nombre de la variable	Descripción
Conjunto respuesta	Se guarda la respuesta a la pregunta.
Tiempo por pregunta sin repetición ni exploración	Toma el tiempo que tardó en responder correctamente la pregunta, quitando el tiempo de repetición y exploración.
Tiempo acumulado sin repetición ni exploración	Mide el tiempo acumulado que tardó en responder correctamente todas las preguntas del nivel hasta donde va, quitando el tiempo de repetición y exploración.
Tiempo con repetición por pregunta	Toma el tiempo que tardó en responder correctamente la pregunta, quitando el tiempo de exploración.
Tiempo con repetición acumulado	Mide el tiempo acumulado que tardó en responder correctamente todas las preguntas del nivel hasta donde va, quitando el tiempo de exploración.
Tiempo con exploración por pregunta	Toma el tiempo que tardó en responder correctamente la pregunta, quitando el tiempo de repetición.
Tiempo con exploración acumulado	Mide el tiempo acumulado que tardó en responder correctamente todas las preguntas del nivel hasta donde va, quitando el tiempo de repetición.
Tiempo con repetición y exploración por pregunta	Toma el tiempo total que tardó en responder correctamente la pregunta.
Tiempo con repetición y exploración acumulado	Mide el tiempo acumulado que tardó en responder correctamente todas las preguntas del nivel hasta donde va.
Número de repeticiones por pregunta	Determina el número de veces que se repitió esa pregunta.
Número de repeticiones acumulado	Cuantifica el número acumulado de veces que se repitieron todas las preguntas del nivel hasta donde va.
Número de exploraciones por pregunta	Determina el número de veces que se entró a la exploración en esa pregunta.
Número de exploraciones acumulado	Cuantifica el número acumulado de veces que se entró a la exploración en todas las preguntas del nivel hasta donde va.
Número de repeticiones más exploraciones por pregunta	Determina el número de veces que se entró a la exploración y a la repetición en esa pregunta.
Número de repeticiones más exploraciones acumulado	Cuantifica el número acumulado de veces que se entró a la exploración y a la repetición en todas las preguntas del nivel hasta donde va.
Errores por pregunta	Determina el número acumulado de veces que el usuario se equivocó en esa pregunta.
Errores acumulados	Cuantifica el número acumulado de veces que el usuario se equivocó en todas las preguntas del nivel hasta donde va.

Cuadro 3.1: Variables

El problema que se presentó a la hora de leer los datos fue que se tuvieron que agregar más variables que describen la operación a la pertenecen y el nivel del mismo, de tal forma que desde una misma tabla se pudiera leer toda la información. Lo anterior, conllevó a resolver un problema de planeación con la obtención de datos. Por otro lado, organizar

Operación	Nivel 1	Nivel 2	Nivel 3	Nivel 4
Unión	dat1_U_n.csv	dat2_U_n.csv	dat3_U_n.csv	dat4_U_n.csv
Intersección	dat1_I_n.csv	dat2_I_n.csv	dat3_I_n.csv	dat4_I_n.csv
Diferencia	dat1_D_n.csv	dat2_D_n.csv	dat3_D_n.csv	dat4_D_n.csv
Complemento	dat1_C_n.csv	dat2_C_n.csv	dat3_C_n.csv	dat4_C_n.csv

Cuadro 3.2: Organización de archivos que se crearon para guardar la información de cada persona ciega por nivel

todo en una sola tabla, agilizó la movilidad de los datos.

Las gráficas también se realizaron en R-Studio, usando la librería de `ggplot2`, la cual se especializa en la gráfica de bases de datos, donde cada columna puede ser una dimensión. Imaginemos que tenemos que graficar cuatro dimensiones, a saber: tiempo, posición, temperatura y tipo de elemento en un plano cartesiano. Se puede resolver graficando la posición contra el tiempo y los demás datos se pueden agregar en la forma del punto, en el tamaño del punto, etc. En este caso, la temperatura pude agregarse con un gradiente de color mientras que el tipo de elemento puede ser la forma del punto. Hacer este tipo de gráficas es sencillo cuando la base de datos está bien acomodada: entre más detalle tenga la base de datos, más datos podrás agregar en el plano. Esta es la idea con la que se construyeron los gráficos que se presentan en el siguiente capítulo.

3.5. Conclusión

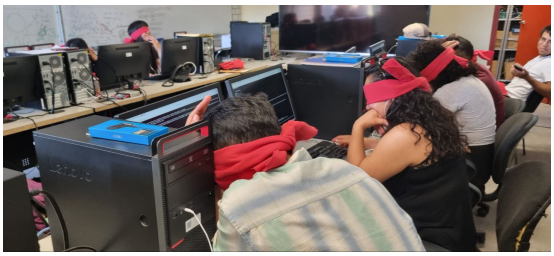
Este capítulo trató todo el proceso de programación, desde su idea, hasta las problemáticas que se tuvieron que resolver. Se describieron las bases de cómo se trabajó la información y en qué condiciones se adquirió. El siguiente capítulo se enfocará sólo a la parte estadística.

Capítulo 4

Análisis estadístico

Los experimentos diseñados estadísticamente permiten eficiencia y economía en el proceso experimental, y la aplicación de métodos estadísticos para examinar los datos resulta **en objetividad científica** al llegar a conclusiones.

D.C. Montgomery y G.C. Runger, *Probabilidad y estadística aplicadas a la ingeniería*



El grupo de control eran universitarios que simularon ser ciegos.

Introducción

En este capítulo se hará un análisis estadístico, desde distintos puntos de vista. Primero se realizará una visualización del experimento, posteriormente un análisis de varianza. Se verán las diferentes relaciones que existen entre las operaciones cognitivamente; por otro lado, se indagará en las distintas relaciones que hay de los usuarios, para corroborar que la brecha entre lo visual y lo no visual se desdibuja poco a poco, además se hará una comparación con un grupo de control.

4.1. Resumen de los datos

En el apéndice C se encuentran las tablas de todos los promedios. El objetivo de dicho apéndice es mostrar un resumen de los datos que se están analizando en este capítulo; las demás variables no se presentarán en forma de tabla, sino en los gráficos.

Es menester primero hablar de los datos desde sus promedios, empezaremos con la unión. La tabla C.1, muestra que casi todos tardaron más en el nivel uno; al contrario del nivel dos, el avance fue mucho más eficiente, lo cual puede interpretarse como el tiempo que tardaron en adaptarse o en entender más la operación. Por otra parte también se ve un incremento de tiempo al pasar del nivel dos al tres e inclusive respecto al cuarto, ya que, en ese nivel se ve la conmutabilidad y asociatividad, respectivamente, de la operación. Por otro lado, hubo usuarios que tardaron más, como el $U1$, quien tuvo una dificultad al entender, pero posteriormente se ven sus mejoras. Usuarios que fueron muy sobresalientes, como el $U17$, cuyos datos marcan casi cero, pero esto se debe a que iba respondiendo exactamente cuando iba sonando la respuesta, se puede entender como si hubiera entendido el concepto bastante bien.

En el caso de la intersección, en la tabla, C.2 fue diferente, el nivel que más se les dificultó en general fue el tres, un ejemplo de ello es el usuario $U1$, no obstante el $U14$ tardo casi lo mismo en el segundo nivel. Empero, se observa una mejoría respecto a la unión.

La diferencia sí mostró un aumento de tiempo considerablemente, como se ve en C.3, y en particular el nivel tres, pues al contrario de las dos operaciones anteriores, la diferencia no es conmutativa: hubo varios usuarios que tardaron en comprender el concepto. Esta operación fue la introducción al complemento, el cual es cognitivamente más difícil, porque para realizar la operación se debe entender primero el conjunto universo, de esa manera, no sólo es hacer una operación, sino entender de qué se esta hablado.

Por su parte, en la tabla C.4, el promedio del complemento, en su nivel más fácil, fue entendido sin muchas complicaciones casi por todos los usuarios. Respecto a la comparación de todas las operaciones, en la tabla C.12 el tiempo donde tardaron más fue la diferencia y el complemento, siendo más la primera. Hecho que sustenta la diferencia entre la división que se hace de operaciones básicas y complejas.

4.2. Gráficos del experimento

Como un primer acercamiento , primero debemos entender el comportamiento del experimento. Poder esbozar el experimento, no sólo permite un análisis rápido, sino también mostrar un retrato, ver las huellas de cada participante, en ese sentido, ver el camino que dejan trazado. Para ello, se construyó un gráfico que expresará en su totalidad el experimento. De esta forma, en esta sección se expondrá todo el experimento gráficamente, por último, veremos una comparación con el grupo de control.

La figura 4.2, representa una línea del tiempo del experimento; en el eje horizontal está el tiempo expresado en segundo; mientras en el vertical se encuentran las operaciones de teoría de conjuntos con sus respectivos niveles. En ese sentido, se piensa el experimento como cuatro sucesos en paralelo de cada operación, que a su vez se vuelven a subdividir en 4 niveles. Es importante subrayar que estos eventos no ocurrieron simultáneamente, pero para efectos de la investigación se tomarán de esa manera. De este modo, se puede observar fácilmente lo que ocurrió a grandes rasgos. Cada punto representa el tiempo total de respuesta del usuario por pregunta, es decir, cada usuario aporto 40 puntos en cada operación y en total 160 puntos.

Es importante hacer notar que ciertas cosas del gráfico, pues el tiempo pudo subir por el número de repeticiones o el número de errores. Sin embargo, podemos advertir qué fue lo que más se les complicó, en donde tuvieron que dedicarle más tiempo para poder responderlo correctamente, ya sea por memoria o por falta de entendimiento.

Dicho lo anterior, la gráfica se acomodó respecto a su dispersión o su varianza, para poder hablar sobre las hipótesis que se hicieron en el capítulo dos, es decir, responder a primera vista ¿Qué operación se les complicó más?

Asimismo, se muestran las medias o promedios (los cuales están representados por una linea horizontal del color respectivo a su nivel), donde se ve su dispersión. Con la finalidad de observar cuál es la operación qué más se les complicó, se muestra la información de esa manera.

Las operaciones en las que más tardaron fueron el complemento y la diferencia, mientras que la intersección fue la operación que más rápido respondieron; este fue un resultado que no era tan esperado, pues intuitivamente se esperaba que tardaran más en la intersección que en la unión y esto se debe a que en la unión se requiere más memoria como veremos más adelante.

Parece señalar que tanto el complemento como la diferencia de operaciones son más complejas de entender, a primera vista. Mientras que la intersección fue una de las operaciones que más rápidamente respondieron, la diferencia (que se entiende como el complemento de la intersección) fue una operación que no fue procesada rápidamente. Por otro lado, la unión, que se necesita ejercitar la memoria, fue la segunda operación en resolverse más rápido. El complemento resultó el más complicado respecto al tiempo.

Se volvió a repetir el experimento, ahora con un grupo de control con 20 participantes. Este nuevo grupo fue integrado por participantes de licenciatura que ya tenía conocimientos previos de teoría de conjuntos. No obstante lo anterior, podemos encontrar algunas similitudes y diferencias. Es importante notar, que cuando se comparen los grupos se debe

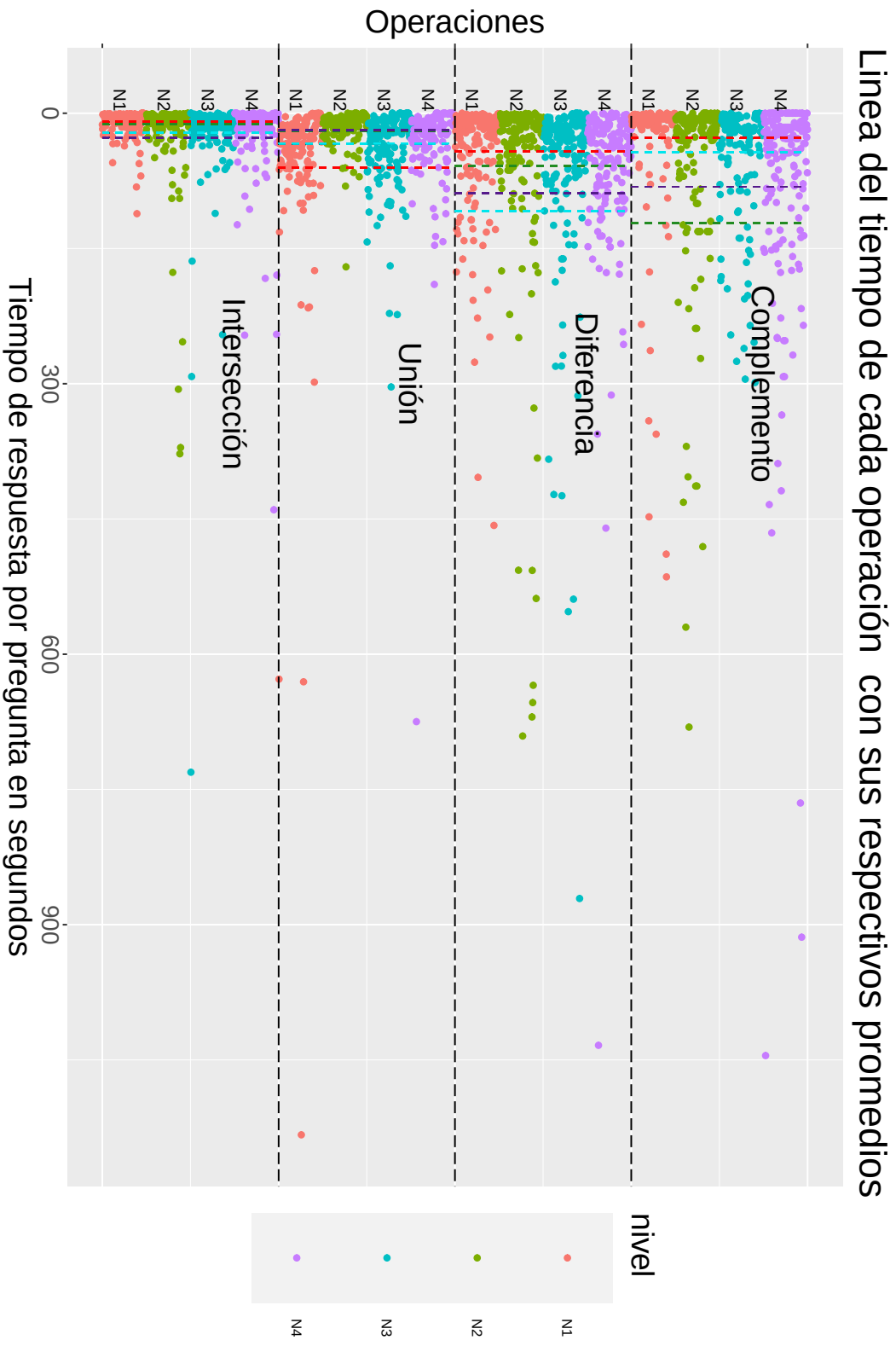


Figura 4.1: Retrato de experimento. Se muestra el tiempo (en segundos) total del experimento. Se ponen en paralelo las cuatro operaciones

pensar proporcionalmente, dado que los tiempos son extremadamente diferentes. La primera diferencia es que el primer grupo, el de personas ciegas, se adaptó mucho más rápido al software, mientras en el grupo de control hay un tiempo de adaptación, el cual se ve en el primer nivel de la unión. La intersección también fue una operación que se realizó más rápidamente, sin embargo la unión, en donde implica más memoria subió al tercer nivel de dificultad. Hay dos motivos que deben considerarse: primero hubo mayor tiempo de adaptabilidad a la aplicación, segundo, se requiere más memoria al contestar las preguntas. Lo interesante, es que en ambos grupos por distintas razones, fue el complemento la operación en donde más tiempo tardaron.

4.2.1. Boxplot

En la sección anterior, pudimos ver rápidamente o crearnos una idea de experimento; no obstante, los datos que se presentaron fueron los de las medias y las varianzas. En este apartado se esbozará la mediana, para poder ver dónde esta el mayor número de población. Para ello se ocupó la herramienta de boxplot, con la cual se pueden ver los cuatro cuartiles de la población.

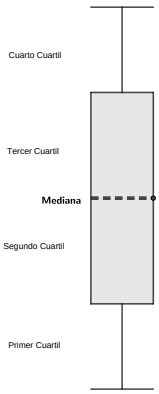


Figura 4.3: Boxplot

La figura 4.2.1 es un gráfico de caja (boxplot) de una distribución normal, donde todo se ve simétrico. Se trata de una visualización exploratoria que sirve para encontrar elementos que se alejan mucho de la población y que modifican drásticamente las medias y varianzas; en otras palabras, datos cargados: con la finalidad de tratar disminuir el sesgo. Cuando no es una muestra normal el gráfico pensarno es simétrico como los que nos aparecen en el experimento.

Al ser un estudio exploratorio es necesario encontrar donde está el mayor porcentaje de la población. Además también se puede estudiar la dispersión, pues la información se ve por cuartiles. Es decir, que si en el experimento hubo puntos aberrantes o puntos que se alejaran mucho de la población, son más fáciles de localizar, porque aparecen lejos. En el caso de los gráficos que se presentan a continuación puntos rojos sobre una línea central de la caja, los cuales representan esos puntos; mientras que los puntos azules más pequeños son puntos en sí que estás representados por los rojos.También podemos ver un histograma dentro del boxplot, siendo los puntos azules vistos desde el eje vertical.

Las variables que se trabajan con este tipo de gráficos son los *errores* y las *repeticiones* de cada pregunta. Esto para comprender mejor el movimiento del tiempo, donde el primero nos habla de la dificultad de comprensión; el segundo, de diversas circunstancias, por ejemplo, memoria, momentos de distracción, etc. Primero, se verán los errores y luego las repeticiones del primer grupo, luego la comparación con el segundo grupo.

La figura 4.2.1 indica todos los errores que hubo. Esta es una de las variables más importantes que trata sobre cómo fueron aprendiendo, dado que los errores pueden tener dos razones: o no se entendió el ejercicio o fue un error de dedo. En el trabajo se supondrá o se les denotará de la primera forma, como si no se hubiera entendido, ya que generalmente

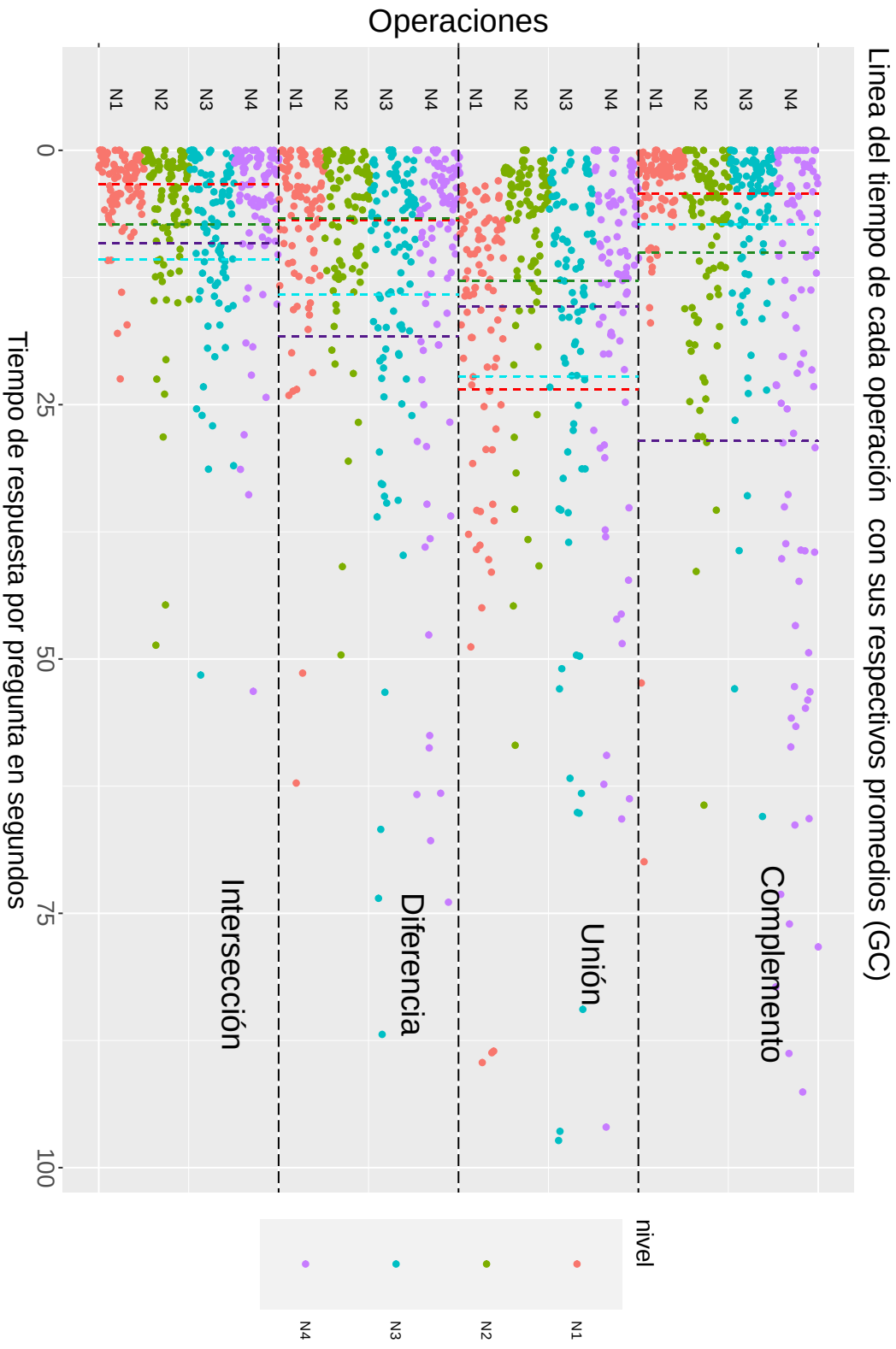


Figura 4.2: Retrato del grupo de control

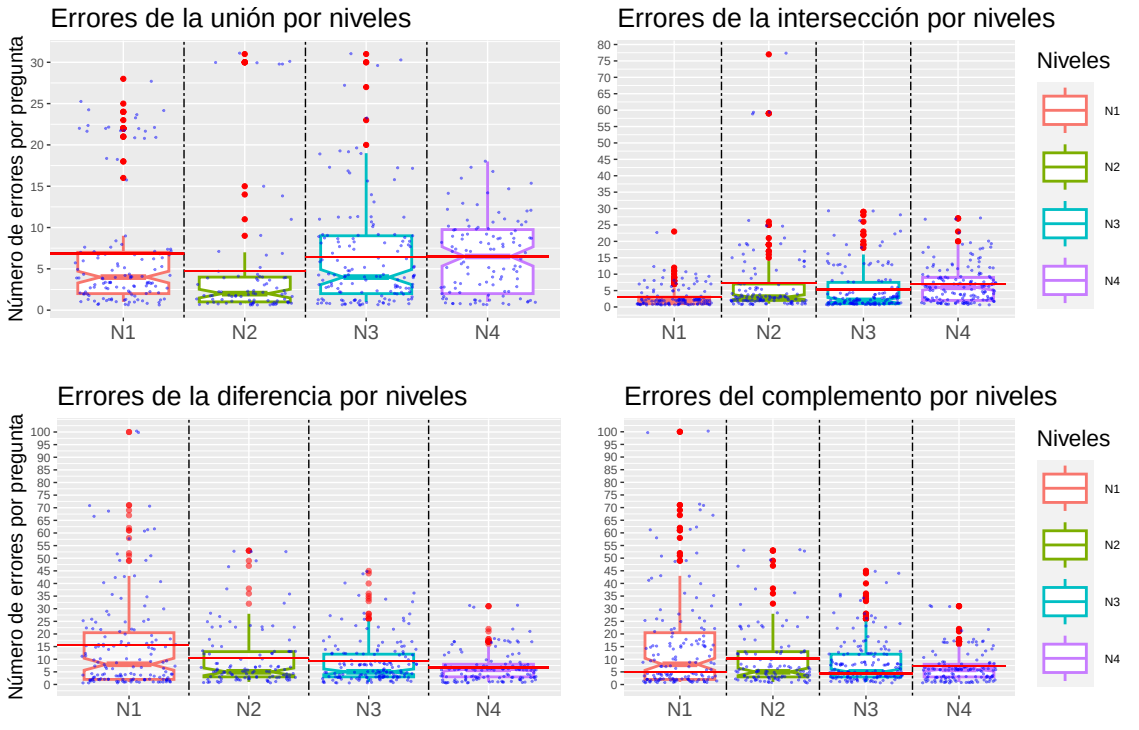


Figura 4.4: Gráfico que nos muestra los errores de la población

se cometen más de dos errores en la pregunta y por simplicidad de las hipótesis. Conviene enfatizar que, posteriormente se hará un análisis de varianza donde se podrán verificar con más rigurosidad las aseveraciones de las proposiciones.

Dicho lo anterior, se expondrá sobre la interpretación que surge de los boxplot. Una de las primeras cosas a notar es que la unión tuvo a lo más 35 errores por una pregunta, siendo los primeros niveles donde más se equivocaban, lo cual nos habla de una curva de aprendizaje, donde al principio, ocurren varios errores, pero al finalizar empiezan a disminuir. No ocurre tan claramente en la intersección, se ven más o menos al mismo nivel, pero sucede que hubo casos en el segundo nivel donde aumentaron gradualmente los errores, siendo interesante que en ese nivel se introduce el conjunto vacío. En los otros niveles de la intersección se mantuvieron los errores más o menos iguales, pero es interesante notar que contrario a nuestra suposición del principio efectivamente se equivocaron más en la intersección que en la unión, pues a lo más hubo 75 errores.

Sobre la diferencia, parece que también hubo una curva de aprendizaje bastante eficiente, siendo en los primeros niveles donde se adquiere el conocimiento (un gran número de errores) y después en los últimos se ve un mejor dominio del conocimiento, pues el número de errores disminuyó considerablemente en el último nivel. Cabe señalar que el número de errores aumentó respecto de la intersección; es decir, sí se les complicó más. Por último, el complemento también se comportó de la misma manera. A primera vista, es una señal de que la aplicación sí está logrando el objetivo de que se vayan comprendiendo poco a poco los conceptos que se estudian.

Sobre las repeticiones, se pueden ver en la figura 4.2.1. Esta variable en realidad se refiere

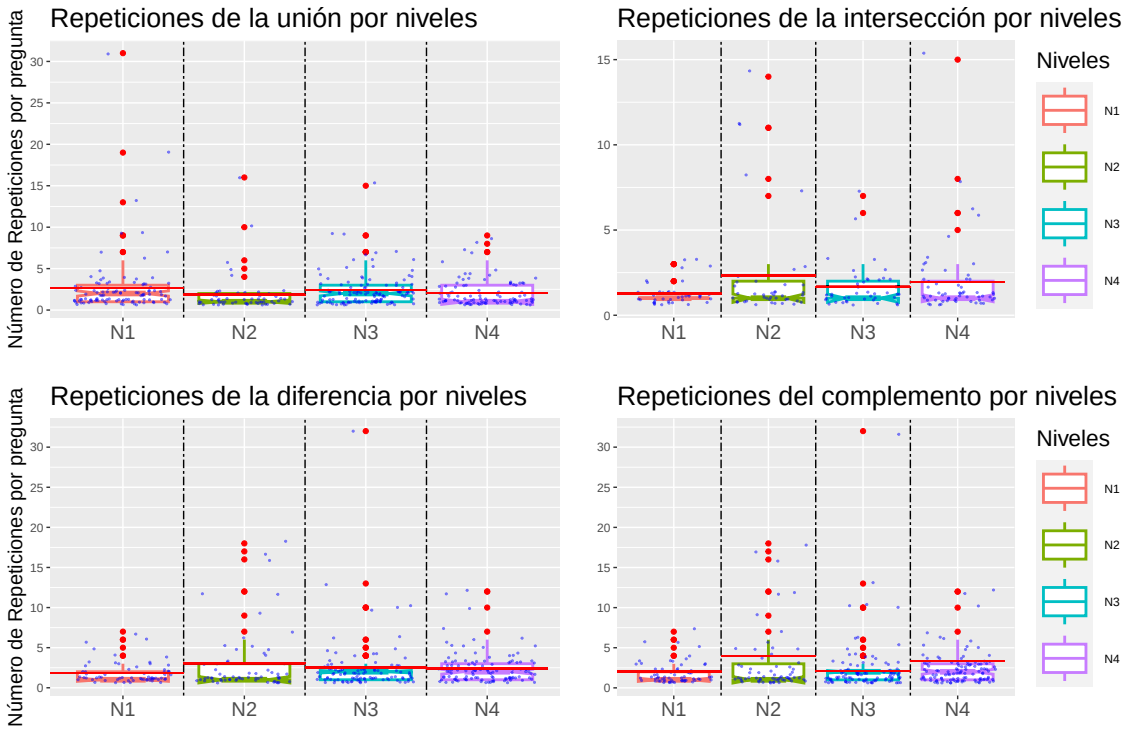


Figura 4.5: Gráfico que muestra la las repeticiones por cada pregunta

a las habilidades cognitivas que presentan los usuarios sobre sus capacidades de memoria, retención, habilidad geográfica, etc. Es importante fijarse en su comportamiento, porque habla por qué varia tanto el tiempo en ciertas operaciones que no deberían variar tan bruscamente.

Las repeticiones de la unión empiezan a disminuir conforme avanzan los niveles, lo cual quiere decir que al inicio se fortaleció o se mejoraron técnicas de memoria, pues a pesar de que en los últimos niveles el número de elementos de cada pregunta aumentaba considerablemente, disminuyeron. Por otra parte, también es notable que tanto los errores como las repeticiones se mantuvieron en los mismos rangos. Respecto a la intersección, hubo más repeticiones tanto en el nivel dos como en el cuatro, lo cual significa que el tiempo del este último aumentó por las repeticiones, no por los errores.

Las repeticiones del complemento y la diferen lo que refiere a los conceptos que se vieron en este nivel. En el nivel tres donde se revisa la no conmutabilidad en la diferencia y la idempotencia en el complemento, se sugiere que al ser temas diferentes a las otras operaciones, tuvieron que repetir más veces.

Cuando revisamos el gráfico de la comparación entre ambos grupos, el de los las personas ciegas y el de control, vemos que tardan menos tiempo, pero esto se debe a que el grupo de control ya tenia un antecedente en el conocimiento de teoría de conjuntos. ¿Entonces para qué tomar en cuenta esta comparación? ¿O en qué sentido se pueden comparar? Como se mencionó anteriormente, se debe pensar proporcionalmente, en cómo sus boxplots varían entre el mismo grupo con las diferentes operaciones. Para ello, se comentará el gráfico desde las operaciones *grosso modo*.

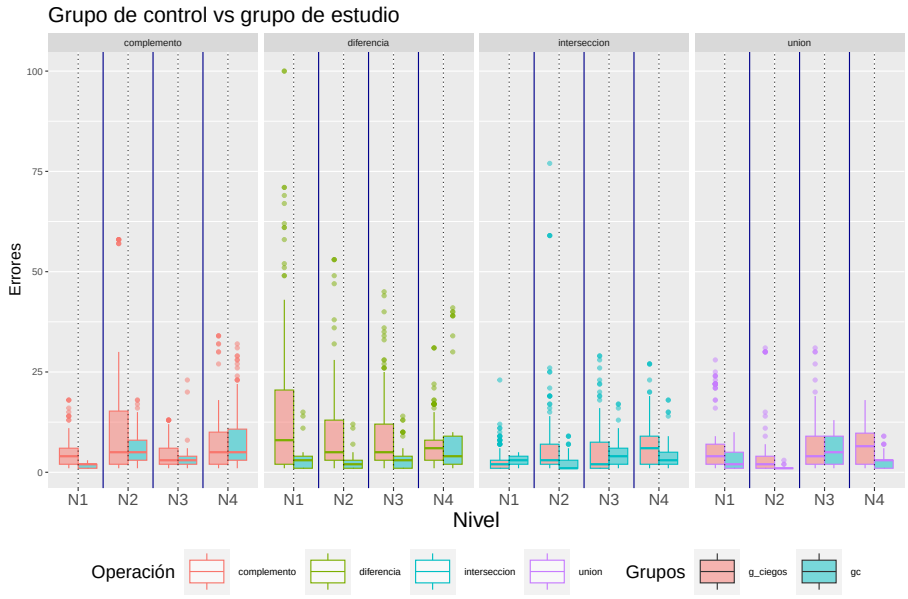


Figura 4.6: Gráfico que compara el grupo de control con el grupo de experimento en los errores

Veamos que el último nivel de la unión fue más complicado para el grupo de control. Sin embargo, en general, su comportamiento respecto a la operación es similar. En la intersección no ocurre lo mismo, pues en el caso de las personas ciegas parece ser cada vez más sencillo, en tanto que en el grupo de control el último nivel también resulto más complicado. En la diferencia, también fue bastante similar la forma en la que fueron aprendiendo. Por último, en el complemento, sólo varió en el último nivel, donde se concentraba todo lo aprendido en el software. Es interesante notar que aunque los tiempos fueran totalmente diferentes hay algunas constantes que se encuentran en los dos grupos.

Respecto las repeticiones, fueron menos que los errores; en particular, hubo un mayor porcentaje de personas ciegas que repitieron las preguntas. Fue similar el número de repeticiones en la unión; mientras que en la intersección el grupo de control casi no repitió, caso que también ocurrió en la diferencia. En el complemento se comportaron similar.

Lo anterior nos habla de que hay operaciones que por sí mismas apuntan a ser más complicadas mientras que otras parecen más sencillas de entender.

Por último, la comparación de los promedios de las operaciones respecto al tiempo total que se tardaron en cada pregunta se muestra en 4.2.1. En este gráfico podemos ver la figura que hace cada operación con su promedio y comparar respectivamente con cada grupo. Como se observa todas las operaciones tienen la misma forma, a excepción de la intersección en el último nivel donde efectivamente se le complicó más al grupo de control. De ahí en fuera, se tiene el mismo comportamiento. Por ello en la sección siguiente se trabaja sólo con el grupo de personas ciegas.

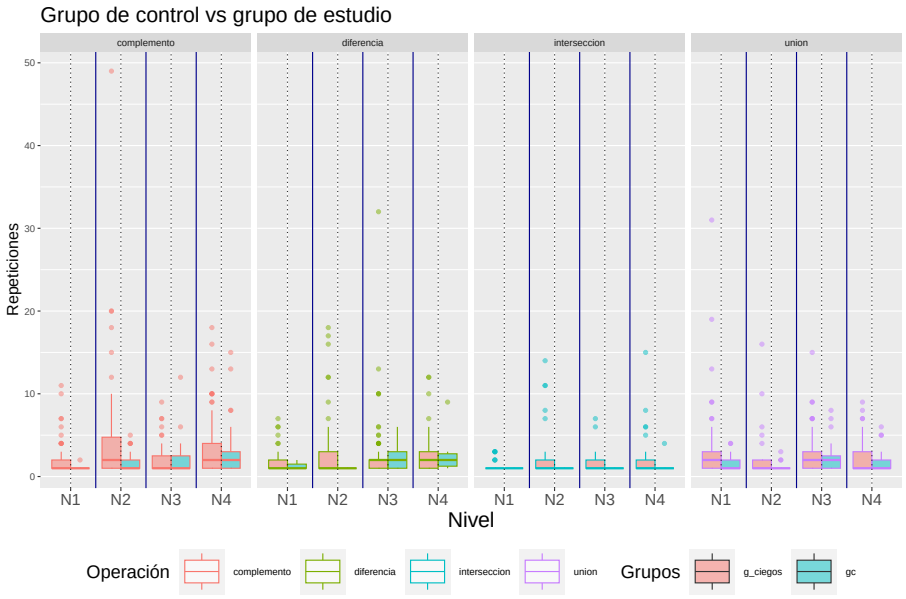


Figura 4.7: Gráfico que compara el grupo de control con el grupo de experimento en las repeticiones

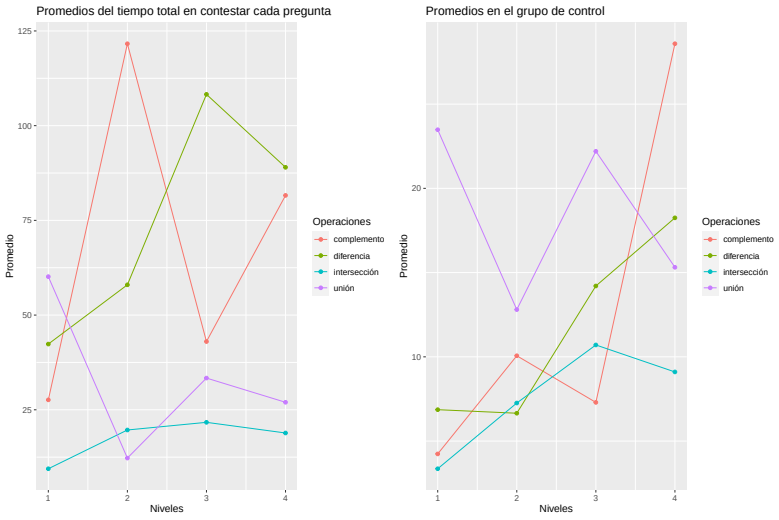


Figura 4.8: Gráfica de medias del tiempo de las operaciones por nivel

4.3. Variabilidad de los datos

De todas las variables mencionadas, en esta sección sólo se trabaja con la del tiempo total ¿Por qué estudiar está variable? Es la más representativa respecto a las demás porque las engloba, es decir, esta variable da una idea de qué esta pasando en general con las operaciones, los usuarios y cómo se relacionan.

Como se observa en la figura 4.9, los datos se acomodaron en forma matricial, donde los renglones representan los niveles; las columnas, las operaciones, que a su vez se subdividen en el número de participantes y de preguntas, respectivamente. En ese sentido, en cada

cuadro hay un total de 180 divisiones y en total 2880. Cabe destacar que el tiempo está medido por segundos y las preguntas son aleatorias respecto al tema que se está revisando.

Los datos se trabajaron en el lenguaje de R, primero se acomodaron en un data frame de tres columnas: operaciones, niveles y tiempo, para poder realizar el análisis de varianza. El acomodo anterior respeta el mismo orden que en la figura 4.9, sólo que viéndolo desde triadas en R. En la figura 4.2 se puede ver la línea de tiempo general del experimento, lo podemos pensar como eventos paralelos, como se mencionó anteriormente.

4.4. Análisis de varianza

Cuando no se tiene toda la población y se quieren estimar parámetros, es necesario acudir a la estadística inferencial, la cual se divide en estimación de parámetros y prueba de hipótesis. Estimar los parámetros ayuda a un siguiente estudio desde las ecuaciones diferenciales o alguna otra rama de las matemáticas. Sin embargo, en esta investigación sólo se verá la estimación y la interpretación del comportamiento de las variables.

En el diseño de experimentos se hace un estudio de la varianza para saber cómo se comportan las medias, es decir, se realiza una prueba de hipótesis, para aceptar o rechazar enunciados acerca de un parámetro que pertenece a una distribución en específico. Con lo anterior, se podrá verificar el modelo. Cabe aclarar que se trata de un estudio estadístico exploratorio, en otras palabras, con este análisis se harán nuevos planteamientos para mejorar el software e incluso vislumbrar cuál es la operación más sencilla de aprender o si se debe de reforzar alguna actividad en concreto. Se podrá hacer una toma de decisiones de cómo mejorar el proceso de aprendizaje, o si se quiere entender, el rendimiento y eficiencia de la aplicación.

Al tratarse de una proporción de la población no se puede garantizar la certeza, a menos que se cuente con toda la población. La prueba de hipótesis presupone que los enunciados solo pueden ser o verdaderos o falsos, por lo que hay dos tipos de hipótesis: la nula y la alternativa, esta última puede ser de dos colas o de una. Cuando el estadístico está en la región de aceptación se puede afirmar un enunciado. Sin embargo, eso no significa que no pueda ocurrir un error: concluir resultados incorrectos. A este tipo de errores se le clasifican de la siguiente manera:

Decisión	H_0 **es verdadeda**	H_0 **es falsa**	**Simbolo**
No se puede rechazar H_0	No hay error	error tipo II	α
Se rechaza H_0	error tipo I	No hay error	$P\|B$

Puede haber dos tipos de errores, de tipo I y de tipo II o β, los cuales están inversamente relacionados, respecto al nivel de significancia y potencia de la prueba. Para resolver el primero se define un $\alpha = P(rechazar\, H_0\, cuando\, es\, verdadera)$ para definir los límites de la región de aceptación, generalmente se toma a $\alpha = 0{,}5, 0{,}01$ (% 5, el % 1 cometer el error tipo 1). En el segundo error es la probabilidad de rechazar la hipótesis nula cuando es falsa es igual a $1 - \beta$, se estima el valor-pi; este es un valor que se considera como el nivel α más bajo en el que los datos son significativos, si el valor.pi es más pequeño que α entonces se acepta la hipótesis nula; de lo contrario, se rechaza.

El análisis de varianza es una técnica desarrollada por Fischer, se usa en diseño de experimentos, para ver qué tanto varían las medias unas entre otras. Consiste en una prueba de hipótesis usando el estadístico de Fischer (F), cuya hipótesis nula es que las medias son iguales, mientras que en alternativa difieren. Sirve para hacer toma de decisiones y mejorar del proceso del experimento, pues estima la variabilidad de las variables (llamadas tratamientos o factores), son las que se quieren predecir. Si se llegase a aceptar la hipótesis nula, entonces podríamos afirmar que no hay una diferencia conceptual entre las operaciones y los niveles, en caso contrario se afirmaría que sí existe una diferencia y que se pueden distinguir los efectos que producen los distintos tratamientos, los efectos a los que se exponen los factores.

Estos se pueden hacer tanto para un factor como para múltiples factores. Lo que difiere son los grados de libertad y el número de relaciones entre los distintos tratamientos, pues se tienen que sacar los datos en sus distintas combinaciones para ver la interacción entre las variables.

$$H_o : \alpha_i = 0 H_a : almenos\, una\, diferente\, a\, cero$$

Donde α_i representan las medias estandarizadas.

	Bloque				
	1	2	...	b	
Tratamiento 1	X_{11}	X_{12}	...	X_{1b}	$\bar{X}_1$
Tratamiento 2	X_{21}	X_{22}	...	X_{2b}	$\bar{X}_2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Tratamiento a	X_{a1}	X_{a2}	...	X_{ab}	$\bar{X}_a$
	$\bar{X}_1$	$\bar{X}_2$		$\bar{X}_b$	

Cuadro 4.1: Tabla cuando hay dos factores en el diseño de experimentos

La tabla 4.1 muestra cómo se deben acomodar los datos cuando se trata de dos factores. En la tabla no hay repeticiones, es decir, cada elemento corresponde a un tratamiento y a un bloque. Sin embargo como veremos en nuestro experimento cada pregunta corresponde a una repetición.

En la figura 4.9, se ve cómo se tienen que acomodar los datos cuando se trata de dos factores: las operaciones y los niveles. Se trata de un mapeo de la variable a analizar, en este caso del tiempo total. Cabe notar que en la programación este tipo de mapeo se puede considerar como un vector en $\mathbb{R}^3$, donde la primer componente corresponde a la operación, la segunda al nivel y la tercera al tiempo. Aunque la información se acomoda diferente, da los mismos resultados.

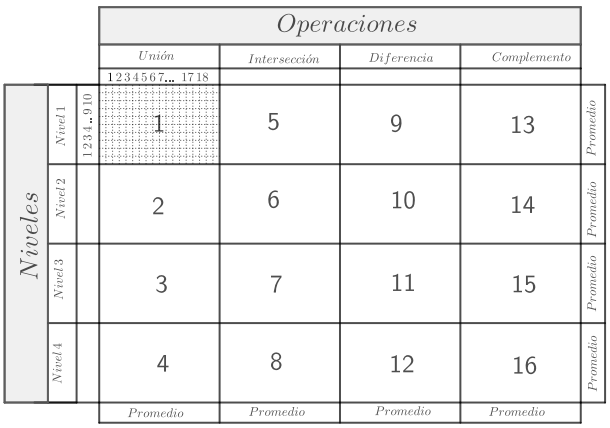


Figura 4.9: Diseño de experimentos

4.5. Análisis de varianza multifactorial

En el análisis de varianza se trabajó con dos factores, α y *beta* y niveles I y J, respectivamente, con $n_ij > 1$ observaciones de sus combinaciones, siendo $y_{ij1}, y_{ij2}, \ldots > n_ij$, planteando el siguiente modelo:

$$y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk} \tag{4.1}$$

Donde μ es la media y ϵ el error, la interacción entre las variables está dada por los términos $(\alpha\beta)_{ij}$, si no se quiere ver la interacción de la ecuación anterior nos queda:

$$y_{ijk} = \mu + \alpha_i + \beta_j + +\epsilon_{ijk} \tag{4.2}$$

El análisis de varianza, se realizó con una herramienta de R. De esta manera se pueden analizar los dos errores que se pueden cometer en la prueba de hipótesis, a saber: α que es el valor de significancia y el valor de pi.

Este primer análisis se hizo sin interacción de los factores.

```
#Análsis de varianza con efectos principales
Modelo1<-aov(ii~nivel+op, data=datostodos)
Análisis de Varianza.  Tabla
             Df    Sum Sq   Mean Sq   F value     Pr(>F)
    operación        3   1634308    544769    8.9454   6.753e-06 ***
    Nivel            3     92792     30931    0.5079     0.67685
    operación:nivel  8   1312388    164049    2.6938     0.00595 **
    Residuals     2865 174477560     60900
    ---
    Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Donde `Df` son los grados de libertad; `Sum Sq` es la suma de cuadrados; `Mean Sq` es la media de la suma de cuadrados; `F value` es el valor estadístico para la prueba de hipótesis, que se calcula como la división entre suma de cuadrados de las variables independientes entre la

suma de cuadrados del error estándar; y `Pr(>F)` es el valor de pi para aceptar o rechazar la hipótesis; mientras los asteriscos nos hablan de qué tan significativa es la prueba, cuando tiene asteriscos se rechaza la hipótesis nula. Como se ver en las operaciones las medias son diferentes hasta con 99 % de probabilidad, eso quiere decir que son significativas y para efectos de la investigación significa que los usuarios perciben diferentes las operaciones. Rechazar la hipótesis nula sólo nos sugiere que no son iguales pero ¿cuál es la que varia más?

Se realizó una prueba de Tukey con R para ver

```
    Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Modelo1)

$nivel
        diff       lwr      upr     p adj
N2-N1 17.996538 -15.50756 51.50063 0.5115468
N3-N1 16.697846 -16.80625 50.20194 0.5750659
N4-N1 19.224755 -14.27934 52.72885 0.4528197
N3-N2 -1.298692 -34.80279 32.20540 0.9996459
N4-N2  1.228217 -32.27588 34.73231 0.9997004
N4-N3  2.526910 -30.97718 36.03100 0.9974187

$op
                             diff       lwr        upr     p adj
diferencia-complemento     5.939861 -27.56423  39.443955 0.9685261 (M iguales)
interseccion-complemento -51.044127 -84.54822 -17.540032 0.0005331 (M  diff)
union-complemento        -35.273943 -68.77804  -1.769848 0.0345329
interseccion-diferencia  -56.983988 -90.48808 -23.479893 0.0000752
union-diferencia         -41.213803 -74.71790  -7.709709 0.0086015
union-interseccion        15.770184 -17.73391  49.274279 0.6205581
```

Donde diff son la diferencia de las medias, lwr es el valor más bajo, upr es el valor más alto y muestra los valores p de cada combinación, es decir, nos muestra si son o no entre ellas significativas. Lo que se muestra es que entre el complemento y la diferencia sienten la misma dificultada, en cambio de la intersección al complemento sí sienten una gran diferencia, la más alejada. Es interesante notar que entre la unión y la intersección no hay gran diferencia, pero entre la intersección y la diferencia sí la hay. Lo cual parece indicar que efectivamente hay dos grupos de operaciones, el primero conformado por la unión y la intersección y el segundo por la diferencia y el complemento. Entre los niveles no hubo gran diferencia, lo cual apunta dos cosas o que no se encuentra gran diferencia entre cursar el nivel 1 al dos o que fueron aprendiendo conforme los niveles.

El siguiente análisis de varianza analiza la interacción entre las variables, el cual podrá ayudarnos a solucionar el cuestionamiento anterior.

```
    Modelo2<-aov(ii~nivel*op,data=datostodos)
    Análisis de Varianza tipo 2
Variable: tiempo total de respuesta por pregunta
               Df  Sum Sq    Mean Sq    F value    Pr(>F)
operación       3   1634308   544769     8.9557     6.654e-06 ***
nivel           3   176736    58912      0.9685     0.406590
```

```
nivel:operación  9   1490555   165617     2.7226     0.003672 **
Resudial 2864 174215450 60829
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
```

De este análisis se sigue que si hay una relación entre las operaciones y las niveles, pero no tan significativa, es decir solo con $alpha = 0{,}05$. Si volvemos a hacer una prueba de Turkey, veremos todas las relaciones que pueden existir:

```
    Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Modelo2)

$nivel
        diff       lwr      upr     p adj
N2-N1 17.996538 -15.41758 51.41066 0.5091957
N3-N1 16.697846 -16.71627 50.11197 0.5728559
N4-N1 19.224755 -14.18936 52.63887 0.4503888
N3-N2 -1.298692 -34.71281 32.11543 0.9996431
N4-N2  1.228217 -32.18590 34.64234 0.9996980
N4-N3  2.526910 -30.88721 35.94103 0.9973979

$op
                           diff       lwr        upr     p adj
diferencia-complemento     5.939861 -27.47426  39.353980 0.9682841
interseccion-complemento -51.044127 -84.45825 -17.630007 0.0005107
union-complemento        -35.273943 -68.68806  -1.859823 0.0338341
interseccion-diferencia  -56.983988 -90.39811 -23.569868 0.0000713
union-diferencia         -41.213803 -74.62792  -7.799684 0.0083633
union-interseccion        15.770184 -17.64394  49.184304 0.6184818

$`nivel:op`
                                      diff         lwr         upr     p adj
N2:complemento-N1:complemento     94.0014961    4.851976 183.1510165 0.0269627
N3:complemento-N1:complemento     15.3959455  -73.753575 104.5454658 0.9999998
N4:complemento-N1:complemento     53.9660288  -35.183492 143.1155491 0.7806987
N3:complemento-N2:complemento    -78.6055507 -167.755071  10.5439697 0.1595860
N4:complemento-N2:complemento    -40.0354674 -129.184988  49.1140530 0.9778168
N4:complemento-N3:complemento     38.5700833  -50.579437 127.7196036 0.9844065

N2:diferencia-N1:diferencia       15.6424275  -73.507093 104.7919479 0.9999998
N3:diferencia-N1:diferencia       65.9134938  -23.236026 155.0630142 0.4455368
N4:diferencia-N1:diferencia       46.6556409  -42.493879 135.8051613 0.9185894
N3:diferencia-N2:diferencia       50.2710663  -38.878454 139.4205866 0.8602900
N4:diferencia-N2:diferencia       31.0132134  -58.136307 120.1627337 0.9984299
N4:diferencia-N3:diferencia      -19.2578529 -108.407373  69.8916674 0.9999961

N2:interseccion-N1:interseccion   10.2311318  -78.918389  99.3806522 1.0000000
N3:interseccion-N1:interseccion   12.2528596  -76.896661 101.4023799 1.0000000
N4:interseccion-N1:interseccion    9.4401753  -79.709345  98.5896957 1.0000000
N3:interseccion-N2:interseccion    2.0217277  -87.127793  91.1712481 1.0000000
N4:interseccion-N2:interseccion   -0.7909565  -89.940477  88.3585638 1.0000000
N4:interseccion-N3:interseccion   -2.8126842  -91.962205  86.3368361 1.0000000

N2:union-N1:union                -47.8889042 -137.038425  41.2606161 0.9009413
```

N3:union-N1:union	-26.7709164	-115.920437	62.3786039	0.9997228
N4:union-N1:union	-33.1628239	-122.312344	55.9866965	0.9966946
N3:union-N2:union	21.1179878	-68.031533	110.2675081	0.9999865
N4:union-N2:union	14.7260803	-74.423440	103.8756006	0.9999999
N4:union-N3:union	-6.3919075	-95.541428	82.7576128	1.0000000
N1:diferencia-N1:complemento	14.7278379	-74.421682	103.8773582	0.9999999
N2:diferencia-N1:complemento	30.3702654	-58.779255	119.5197857	0.9987636
N3:diferencia-N1:complemento	80.6413317	-8.508189	169.7908521	0.1301893
N4:diferencia-N1:complemento	61.3834788	-27.766042	150.5329991	0.5774301
N1:interseccion-N1:complemento	-18.1843009	-107.333821	70.9652194	0.9999982
N2:interseccion-N1:complemento	-7.9531691	-97.102689	81.1963512	1.0000000
N3:interseccion-N1:complemento	-5.9314414	-95.080962	83.2180790	1.0000000
N4:interseccion-N1:complemento	-8.7441256	-97.893646	80.4053947	1.0000000
N1:union-N1:complemento	32.5225861	-56.626934	121.6721064	0.9973294
N2:union-N1:complemento	-15.3663181	-104.515838	73.7832022	0.9999998
N3:union-N1:complemento	5.7516697	-83.397851	94.9011900	1.0000000
N4:union-N1:complemento	-0.6402378	-89.789758	88.5092825	1.0000000
N1:diferencia-N2:complemento	-79.2736583	-168.423179	9.8758621	0.1494398
N2:diferencia-N2:complemento	-63.6312307	-152.780751	25.5182896	0.5113967
N3:diferencia-N2:complemento	-13.3601644	-102.509685	75.7893559	1.0000000
N4:diferencia-N2:complemento	-32.6180173	-121.767538	56.5315030	0.9972420
N1:interseccion-N2:complemento	-112.1857971	-201.335317	-23.0362767	0.0017657
N2:interseccion-N2:complemento	-101.9546652	-191.104186	-12.8051449	0.0088336
N3:interseccion-N2:complemento	-99.9329375	-189.082458	-10.7834172	0.0118691
N4:interseccion-N2:complemento	-102.7456217	-191.895142	-13.5961014	0.0078530
N1:union-N2:complemento	-61.4789100	-150.628430	27.6706103	0.5746266
N2:union-N2:complemento	-109.3678142	-198.517335	-20.2182939	0.0028029
N3:union-N2:complemento	-88.2498264	-177.399347	0.8996939	0.0557560
N4:union-N2:complemento	-94.6417339	-183.791254	-5.4922136	0.0247614
N1:diferencia-N3:complemento	-0.6681076	-89.817628	88.4814127	1.0000000
N2:diferencia-N3:complemento	14.9743199	-74.175200	104.1238403	0.9999999
N3:diferencia-N3:complemento	65.2453862	-23.904134	154.3949066	0.4645900
N4:diferencia-N3:complemento	45.9875333	-43.161987	135.1370537	0.9272012
N1:interseccion-N3:complemento	-33.5802464	-122.729767	55.5692739	0.9962154
N2:interseccion-N3:complemento	-23.3491146	-112.498635	65.8004057	0.9999501
N3:interseccion-N3:complemento	-21.3273869	-110.476907	67.8221335	0.9999846
N4:interseccion-N3:complemento	-24.1400711	-113.289591	65.0094493	0.9999236
N1:union-N3:complemento	17.1266406	-72.022880	106.2761610	0.9999992
N2:union-N3:complemento	-30.7622636	-119.911784	58.3872568	0.9985683
N3:union-N3:complemento	-9.6442758	-98.793796	79.5052446	1.0000000
N4:union-N3:complemento	-16.0361833	-105.185704	73.1133371	0.9999997
N1:diferencia-N4:complemento	-39.2381909	-128.387711	49.9113294	0.9816284
N2:diferencia-N4:complemento	-23.5957634	-112.745284	65.5537570	0.9999429
N3:diferencia-N4:complemento	26.6753029	-62.474217	115.8248233	0.9997346
N4:diferencia-N4:complemento	7.4174500	-81.732070	96.5669704	1.0000000

N1:interseccion-N4:complemento	-72.1503297	-161.299850	16.9991906	0.2838972
N2:interseccion-N4:complemento	-61.9191979	-151.068718	27.2303225	0.5616812
N3:interseccion-N4:complemento	-59.8974701	-149.046990	29.2520502	0.6208445
N4:interseccion-N4:complemento	-62.7101544	-151.859675	26.4393660	0.5384155
N1:union-N4:complemento	-21.4434427	-110.592963	67.7060777	0.9999835
N2:union-N4:complemento	-69.3323469	-158.481867	19.8171735	0.3526548
N3:union-N4:complemento	-48.2143591	-137.363879	40.9351613	0.8958992
N4:union-N4:complemento	-54.6062666	-143.755787	34.5432538	0.7650597
N1:interseccion-N1:diferencia	-32.9121388	-122.061659	56.2373815	0.9969569
N2:interseccion-N1:diferencia	-22.6810070	-111.830527	66.4685134	0.9999657
N3:interseccion-N1:diferencia	-20.6592792	-109.808800	68.4902411	0.9999899
N4:interseccion-N1:diferencia	-23.4719635	-112.621484	65.6775569	0.9999466
N1:union-N1:diferencia	17.7947482	-71.354772	106.9442686	0.9999987
N2:union-N1:diferencia	-30.0941560	-119.243676	59.0553644	0.9988869
N3:union-N1:diferencia	-8.9761682	-98.125689	80.1733522	1.0000000
N4:union-N1:diferencia	-15.3680757	-104.517596	73.7814447	0.9999998
N1:interseccion-N2:diferencia	-48.5545663	-137.704087	40.5949540	0.8904554
N2:interseccion-N2:diferencia	-38.3234345	-127.472955	50.8260858	0.9853431
N3:interseccion-N2:diferencia	-36.3017068	-125.451227	52.8478136	0.9914403
N4:interseccion-N2:diferencia	-39.1143910	-128.263911	50.0351293	0.9821707
N1:union-N2:diferencia	2.1523207	-86.997200	91.3018410	1.0000000
N2:union-N2:diferencia	-45.7365835	-134.886104	43.4129368	0.9302671
N3:union-N2:diferencia	-24.6185957	-113.768116	64.5309246	0.9999020
N4:union-N2:diferencia	-31.0105032	-120.160024	58.1390171	0.9984314
N1:interseccion-N3:diferencia	-98.8256326	-187.975153	-9.6761123	0.0139070
N2:interseccion-N3:diferencia	-88.5945008	-177.744021	0.5550195	0.0534877
N3:interseccion-N3:diferencia	-86.5727731	-175.722293	2.5767473	0.0679867
N4:interseccion-N3:diferencia	-89.3854573	-178.534978	-0.2359370	0.0485773
N1:union-N3:diferencia	-48.1187456	-137.268266	41.0307747	0.8973973
N2:union-N3:diferencia	-96.0076498	-185.157170	-6.8581295	0.0205898
N3:union-N3:diferencia	-74.8896620	-164.039182	14.2598583	0.2252875
N4:union-N3:diferencia	-81.2815695	-170.431090	7.8679508	0.1218592
N1:interseccion-N4:diferencia	-79.5677797	-168.717300	9.5817406	0.1451292
N2:interseccion-N4:diferencia	-69.3366479	-158.486168	19.8128724	0.3525439
N3:interseccion-N4:diferencia	-67.3149202	-156.464441	21.8346002	0.4064169
N4:interseccion-N4:diferencia	-70.1276044	-159.277125	19.0219159	0.3324449
N1:union-N4:diferencia	-28.8608927	-118.010413	60.2886276	0.9993167
N2:union-N4:diferencia	-76.7497969	-165.899317	12.3997234	0.1904096
N3:union-N4:diferencia	-55.6318091	-144.781329	33.5177112	0.7390263
N4:union-N4:diferencia	-62.0237166	-151.173237	27.1258037	0.5586065
N1:union-N1:interseccion	50.7068870	-38.442633	139.8564074	0.8519198
N2:union-N1:interseccion	2.8179828	-86.331538	91.9675032	1.0000000
N3:union-N1:interseccion	23.9359706	-65.213550	113.0854910	0.9999314
N4:union-N1:interseccion	17.5440631	-71.605457	106.6935835	0.9999989
N1:union-N2:interseccion	40.4757552	-48.673765	129.6252755	0.9754620
N2:union-N2:interseccion	-7.4131490	-96.562669	81.7363713	1.0000000

N3:union-N2:interseccion	13.7048388	-75.444682	102.8543592	1.0000000
N4:union-N2:interseccion	7.3129313	-81.836589	96.4624517	1.0000000
N1:union-N3:interseccion	38.4540275	-50.695493	127.6035478	0.9848530
N2:union-N3:interseccion	-9.4348767	-98.584397	79.7146436	1.0000000
N3:union-N3:interseccion	11.6831111	-77.466409	100.8326314	1.0000000
N4:union-N3:interseccion	5.2912036	-83.858317	94.4407239	1.0000000
N1:union-N4:interseccion	41.2667117	-47.882809	130.4162320	0.9707500
N2:union-N4:interseccion	-6.6221925	-95.771713	82.5273278	1.0000000
N3:union-N4:interseccion	14.4957953	-74.653725	103.6453156	0.9999999
N4:union-N4:interseccion	8.1038878	-81.045633	97.2534081	1.0000000

Con esta prueba se observa ver la interacción entre todas las variables, además de ver si hay una diferencia entre los niveles y las operaciones y cuáles son. De esta manera se podrán modificar los niveles para que queden en forma de más dificultad. Recordemos que si el valor de p es mayor a 0.5 entonces las medias son iguales y en el caso contrario son diferentes.

En el caso del complemento sólo hubo una gran diferencia entre el nivel 1 y el 2. Esto quiere decir que al principio fue complicado y después ya no tanto, pues en los niveles siguientes ya no hay una diferencia como tal.

En los niveles de la diferencia fueron casi todos iguales, a excepción del 1 con el 3, lo cual indica que se pueden hacer algunas adaptaciones a estos niveles para que no haya un brinco tan marcado. Por otro lado también indicaría que la no conmutabilidad fue un tema que confundió a los usuarios.

Sobre la intersección no hubo un cambio entre los niveles, e por lo que no tuvieron problemas en ningún nivel, al igual que la unión. Lo cual fortalece más la hipótesis de que efectivamente las operaciones que llamamos complejas son más difíciles de adquirir.

Sobre la interacción de las operaciones y los niveles podemos decir que la más diferente fue el nivel 1 de la diferencia con el nivel dos del complemento

4.6. Conclusión

Existe una diferencia de aprendizaje en las operaciones básicas de teoría de conjuntos. Tanto el complemento como la diferencia representaron una mayor dificultad de aprendizaje.

Capítulo 5

Conclusiones

La discriminación de la educación puede ir mermando lentamente con el aporte de todos. A continuación se enlistan las conclusiones:

☛ **Cualitativas**

- El análisis de los datos indica que el software de enseñanza de matemáticas es efectivo para mejorar el rendimiento de los estudiantes en teoría de conjuntos y que es bien recibido. Estos resultados son consistentes con la literatura previa sobre la eficacia de los softwares educativos en el aprendizaje de las matemáticas.
- Además, los participantes informaron actitudes positivas hacia el software. En una encuesta informal posterior a las pruebas, la mayoría de los participantes señalaron que el software fue útil para mejorar su comprensión de los conceptos matemáticos de teoría de conjuntos, y que les gustó usar el software para aprender.
- Aprender las operaciones básicas de teoría de conjuntos es el primer paso para extrapolar ese conocimiento a la lógica proposicional u otras ramas de las matemáticas.

☛ **Cuantitativas**

- En el análisis de varianza de efectos principales, se rechaza la hipótesis nula de que las medias en las operaciones sean iguales con un 99 % de confianza. Mientras en los niveles se acepta la hipotesis nula.
- En ambos grupos se muestra un comportamiento similar (proporcionalmente) en los promedios de las operaciones
- En la prueba de Turkey, las medias del complemento-diferencia y unión-intersección se acepta la hipotesis nula con un 95 % de confianza. Es decir, cada grupo de operaciones esta relacionado respecto a la dificultad que presento el usuario. En todas las demás operaciones vario el resultado.

Es importante tener en cuenta que la muestra fue un grupo de 18 participantes para probar el software. Se necesitan estudios adicionales para generalizar estos resultados a otras poblaciones de estudiantes. Sin embargo, este trabajo es el inicio de un proyecto de enseñanza de las matemáticas a personas ciegas y en general al público que quiera tener un

acercamiento diferente, se espera que en un futuro cercano se pueda ampliar a otras áreas de las matemáticas. Respecto al modelo de aprendizaje de teoría de conjuntos resulto más útil de lo esperado, promete mejores modelos.

Anexo A

Códigos del programa de teoría de Conjuntos

A.1. Unión

```
#bibliotecas
import time
from datetime import datetime
import pandas as pd
from playsound import playsound
from tkinter import Tk, Label, Button
import random as rd
import math
import os

# directorio de trabajo
direc=os.getcwd()

#definiciones-funciones

#Audio de los números
def elemetosconj(conj):
    for r in conj:
        if r=="0":
            audiosn1=playsound(direc + '/0.mp3')
            uno="1"
        elif r=="1":
            audiosn1=playsound(direc + '/1.mp3')
            dos="2"
        elif r=="2":
            audiosn2=playsound(direc + '/2.mp3')
            tres="3"
        elif r=="3":
            audiosn3=playsound(direc + '/3.mp3')
            print("3")
        elif r=="4":
            audiosn4=playsound(direc + '/4.mp3')
            cuatro="4"
        elif r=="5":
            audiosn5=playsound(direc + '/5.mp3')
            cinco="5"
        elif r=="6":
            audiosn5=playsound(direc + '/6.mp3')
            seis="6"
        elif r=="7":
            audiosn5=playsound(direc + '/7.mp3')
            siete="7"
        elif r=="8":
            audiosn5=playsound(direc + '/8.mp3')
            ocho="8"
        elif r=="9":
            audiosn5=playsound(direc + '/9.mp3')
            nueve="9"
    return(conj)

#función para crear conjuntos con números aleatorios
def rellenacon(conj,a,b,n):
    for i in range(n):
        agre=rd.randint(a,b)
        agre=str(agre)
        conj.add(agre)

#Crea conjuntos disjuntos
def rellenacondife(conj,a,b,n):
    for i in range(n):
        t=rd.randint(a,b)
        t=str(t)
        if t not in conj:
            conj.add(t)
        else:
            while t not in conj:
                t=rd.randint(a,b)
                t=str(t)

#Cread dos números diferentes
def creanumdif(a,b):
    x=rd.randint(a,b)
    y=rd.randint(a,b)
    while y==x:
        x=rd.randint(a,b)
        y=rd.randint(a,b)
    x=str(x)
    y=str(y)
    return(x,y)

#Crea conjuntos disjuntos
#def creadosconjdis1(conj1,conj2,a,b,n):
#    for i in range(n):
#        r,w=creanumdif(a,b)
#        conj3=conj1 & conj2
#        if len(conj3)==0:
#            conj1.add(r)
#            conj2.add(w)
#    return(conj1,conj2)


#crea conjuntos disjuntos
def creadosconjdis(conj1,conj2,a,b,n):
    a=math.floor(a/2)
    b=math.floor(b/2)
    for i in range(n):
        a1=2*rd.randint(a,b)
        b1=1+(2*rd.randint(a,b))
        a1=str(a1)
        b1=str(b1)
        conj1.add(a1)
        conj2.add(b1)
    return(conj1,conj2)

#función de explorar dos conjuntos
def explorar(a,b,repe):
    conjAlist=list(a)
    conjBlist=list(b)
    cambiarconj="s"
    tamA=len(conjAlist)
    tamB=len(conjBlist)
    muevedere="d"
    mueveizqu="a"
    posicionA=0
    posicionB=0
    print(conjAlist[0])
    while repe=='*':
        audioseje5au=playsound(direc + '/estas1.mp3')
        audioseje5au=playsound(direc + '/mover.mp3')
        while cambiarconj=="s":
            print("Estás en el primer conjunto")
            usumover=input("¿A dónde te quiere mover?")
            usumover.lower()
            if (usumover=="d" or usumover=="D"):
                posicionA=(posicionA+1) % tamA
                elemetosconj(conjAlist[posicionA])
                print(conjAlist[posicionA])
            elif (usumover=="a" or usumover=="A"):
```

```
                posicionA=(posicionA-1) % tamA
                elemetosconj(conjAlist[posicionA])
                print(conjAlist[posicionA])
            elif (usumover=="s" or usumover=="S"):
                cambiarconj="ss"
            elif usumover=="-":
                print("Estás saliendo de la exploración")
                audioseje5au=playsound(direc + '/salirexplo.mp3')
                break
            elif usumover is not ('d' or 'a' or 'ss' or 'D'
            or 'A' or 'SS'):
                print("Tu opción no es valida")
                audioseje5au=playsound(direc + '/novalida.mp3')
        if usumover=='-':
            break
        audioseje5au=playsound(direc + '/estas2.mp3')
        audioseje5au=playsound(direc + '/mover.mp3')
        while cambiarconj=="ss":
            print("Estás en el segundo conjunto")
            usumover=input("¿A dónde te quiere mover?")
            usumover.lower()
            if (usumover=="d" or usumover=="D"):
                posicionB=(posicionB+1) % tamB
                elemetosconj(conjBlist[posicionB])
                print(conjBlist[posicionB])
            elif (usumover=="a" or usumover=="A"):
                posicionB=(posicionB-1) % tamB
                elemetosconj(conjBlist[posicionB])
                print(conjBlist[posicionB])
            elif (usumover=="s" or  usumover=="S"):
                cambiarconj="s"
            elif usumover=="-":
                print("Estás saliendo de la exploración")
                audioseje5au=playsound(direc + '/salirexplo.mp3')
                break
            elif usumover is not ('d' or 'a' or 's' or 'D' or
            'A' or 'S'):
                print("Tu opción no es valida")
                audioseje5au=playsound(direc + '/novalida.mp3')
        if repe != '-':
            repe='*'
        else:
             audioseje5au=playsound(direc + '/salirexplo.mp3')
             repe='-'
             break
        if usumover =='-':
            break
    return(a,b)

#Función para explorar tres conjuntos
def explorar3(a,b,c,repe):
    conjAlist=list(a)
    conjBlist=list(b)
    conjClist=list(c)
    cambiarconj="s"
    tamA=len(conjAlist)
    tamB=len(conjBlist)
    tamC=len(conjClist)
    muevedere="d"
    mueveizqu="a"
    posicionA=0
    posicionB=0
    posicionC=0
    print(conjAlist[0])
    while repe=='*':
        audioseje5au=playsound(direc + '/estas1.mp3')
        audioseje5au=playsound(direc + '/mover.mp3')
        while cambiarconj=="s":
            print("Estás en el primer conjunto")
            usumover=input("¿A dónde te quieres mover?")
            usumover.lower()
            if (usumover=="d" or usumover=="D"):
                posicionA=(posicionA+1) % tamA
                elemetosconj(conjAlist[posicionA])
                print(conjAlist[posicionA])
            elif (usumover=="a" or usumover=="A"):
                posicionA=(posicionA-1) % tamA
                elemetosconj(conjAlist[posicionA])
                print(conjAlist[posicionA])
            elif (usumover=="s" or usumover=="S"):
                cambiarconj="ss"
            elif usumover=="-":
                print("Estás saliendo de la exploración")
                audioseje5au=playsound(direc + '/salirexplo.mp3')
                break
            elif usumover is not ('d' or 'a' or 'ss'
            or 'D' or 'A' or 'SS'):
                print("Tu opcion no es valida")
                audioseje5au=playsound(direc + '/novalida.mp3')
        if usumover=='-':
            break
        audioseje5au=playsound(direc + '/estas2.mp3')
        audioseje5au=playsound(direc + '/mover.mp3')
        while cambiarconj=="ss":
            print("Estás en el segundo conjunto")
            usumover=input("¿A dónde te quieres mover?")
            if (usumover=="d" or usumover=="D"):
                posicionB=(posicionB+1) % tamB
                elemetosconj(conjBlist[posicionB])
                print(conjBlist[posicionB])
    #mover a la izquierda
            elif (usumover=="a" or usumover=="A"):
                posicionB=(posicionB-1) % tamB
                elemetosconj(conjBlist[posicionB])
                print(conjBlist[posicionB])
            elif (usumover=="s" or usumover=="S"):
                cambiarconj="sss"
            elif usumover=="-":
                print("Estás saliendo de la exploración")
                audioseje5au=playsound(direc + '/salirexplo.mp3')
                break
            elif usumover is not ('d' or 'a' or 's'
            or 'D' or 'A' or 'S'):
                print("Tu opción no es valida")
                audioseje5au=playsound(direc + '/novalida.mp3')
        if usumover=='-':
            break
        audioseje5au=playsound(direc + '/estas3.mp3')
        audioseje5au=playsound(direc + '/mover.mp3')
        while cambiarconj=="sss":
            print("Estás en el tercer conjunto")
            usumover=input("¿A Dónde te quiere mover?")
            usumover.lower()
            if (usumover=="d" or usumover=="D"):
                posicionC=(posicionC+1) % tamC
                elemetosconj(conjClist[posicionC])
                print(conjClist[posicionC])
            elif (usumover=="a" or usumover=="A"):
                posicionC=(posicionC-1) % tamC
                elemetosconj(conjClist[posicionC])
                print(conjClist[posicionC])
            elif (usumover=="s" or usumover=="S"):
                cambiarconj="s"
            elif usumover=="-":
                print("Estás saliendo de la exploración")
                audioseje5au=playsound(direc + '/salirexplo.mp3')
                break
            elif usumover is not ('d' or 'a'
            or 's' or 'D' or 'A' or 'S'):
                print("Tu opción no es valida")
                audioseje5au=playsound(direc + '/novalida.mp3')
        if usumover !='-':
            repe='*'
        else:
            audioseje5au=playsound(direc + '/salirexplo.mp3')
            repe='-'
            break
    return(a,b,c)

#Conjunto respuesta
def respconj(conj, n,a,b):
    numr=0.
    tiem=0.
    nt=0.
    while len(conj)<n:
        print("Ingresa los elementos con un enter")
        uu=input("")
        if uu=='+':
            print("Se volvera a repetir el ejercicio")
            audioseje5au=playsound(direc + '/repetirejercicio.mp3')

            break
        if uu=='*':
            numr=numr+1
            ta=time.time()
            a,b=explorar(a,b,'*')
            tb=time.time()
            tiem=tb-ta
            audioss11=playsound(direc + '/pide.mp3')
        else:
            conj.add(uu)

        nt=tiem+nt
    return(conj, nt, numr)


def respconj3(conj, n,a,b,c):
    numr=0.
    tiem=0.
    nt=0.
    while len(conj)<n:
        print("Ingresa los elementos con un enter")
        uu=input("")
        if uu=='+':
            print("Se volverá a repetir el ejercicio")
            audioseje5au=playsound(direc + '/repetirejercicio.mp3')
            break
        if uu=='*':
            numr=numr+1
            ta=time.time()
            a,b,c=explorar3(a,b,c,'*')
            tb=time.time()
```

```
            tiem=tb-ta
            audioss11=playsound(direc + '/pide.mp3')
        else:
            conj.add(uu)
        nt=tiem+nt
    return(conj,nt,numr)

#Inicio del programa
audioss11=playsound(direc + '/bienvenido.mp3')

#numero de preguntas
numpreg=10
repetir="+"

print("menu")
#numpreg=(input("ingrese el numero de preguntas"))

print('''Bienvenido a las actividades de la unión''')

print('''En este programa realizaras diversas actividades que te
↪ guíen en el aprendizaje
de la unión, la dificultad de los niveles agrega matices
↪ conceptuales. Se recomienda
las hagas en orden. ''')
audioss14=playsound(direc + '/instru0.mp3')
audioss14=playsound(direc + '/menu.mp3')
#variable menu
menu="0"
print(''' Aprieta el número del nivel que deseas cursas. Has
↪ ingresado al menú:
presiona cero para la actividad de aprendizaje; uno para la
↪ evaluación en su nivel
más básico; dos para la unión de conjuntos con elementos repetidos;
↪ tres para la
propiedad de la conmutatividad;  cuatro para la propiedad de
↪ asociatividad;  y en el
cinco aprenderás a ingresar respuestas, seis para salir
  ''') #cambiar el final

while (menu=="0" or menu=="1" or menu=="2"
                 or menu=="3" or menu=="4"
                 or menu=="5" or menu=="6"
                 or menu=="7"):
    audioss14=playsound(direc + '/menu1.mp3')
    print("Estás en el menú")
    menu=input("Ingresa tu opción ")
    if menu=="0":
        print("---Actividad didáctica---")
        audiosresc=playsound(direc + '/actividaddica.mp3')

        eleccion='''¿Cómo quieres estudiar de forma ordenada, es
        ↪ decir, primero la
        definición, posteriormente ejemplos, o quieres entrar a una
↪ parte especifica
        de la didáctica donde elegirás tu orden? Presiona A para la
↪ primera opción
        o D para la segunda '''
        audioseje5au=playsound(direc + '/eleccion.mp3')
        bande=True
        while bande==True:
            elec=input("Ingresa tu opción ")
            if (elec=='a' or elec=='d' or elec=='A' or elec=='D' ):
                bande=False
            else:
                print("Opcion no valida")
                audioseje5au=playsound(direc + '/novalida.mp3')
        if (elec=='a' or elec=='A'):
            audioseje5au=playsound(direc + '/opcionA.mp3')
            print("Has elegido la opcion A")
            definicion='''La unión es una operación lógica que
            ↪ consiste en unir
            los elementos de dos o más conjuntos, lo cual conforma un
↪ nuevo conjunto.'''
            audiosdefiau=playsound(direc + '/defi.mp3')
            bande1=True
            while bande1==True:
                audioss14=playsound(direc + '/repemasenter.mp3')
                respdida=input('''Si quieres repetir la definición
                ↪ ingresa el signo +,
                de lo contrario da un enter para continuar''')
                if respdida=="+":
                    print(definicion)
                    audiosdefiau=playsound(direc + '/defi.mp3')
                elif respdida != "+":
                    bande1=False
                    print("Sin repetición")
            Ejemplo1='''Ejemplo uno, supongamos dos conjuntos, el de
            ↪ los hombres y
            el de las mujeres, si realizamos la operación de la unión
↪ de los dos
            conjuntos creamos un nuevo conjunto que estaría
↪ conformado por todos los
            seres humanos, es decir, hombres y mujeres.'''
            audioseje1au=playsound(direc + '/ejem1.mp3')
            bande1=True
            while bande1==True:
                audioss14=playsound(direc + '/repemasenter.mp3')
                respdida=input('''si quieres repetir el ejemplo uno
                ↪ ingresa el signo
                +''')
                if respdida=="+":
                    print(definicion)
                    audiosdefiau=playsound(direc + '/ejem1.mp3')
                elif respdida != "+":
                    bande1=False

            Ejemplo2=''' Ejemplo dos, tenemos tres conjuntos. El
            ↪ primero es conformado
            por un lápiz y una goma; el segundo, una pluma y una
↪ mochila; y el
            tercero, un cuaderno. Hagamos la operación de unión de
↪ los conjuntos,
            del cual resulta un nuevo conjunto conformado por un
↪ lápiz, una goma,
            una pluma, una mochila y un cuaderno. Como podemos ver la
↪ unión une todos
            los elementos de los conjuntos que participan en la
↪ operación.'''
            audioseje2au=playsound(direc + '/ejem2.mp3')

            bande1=True
            while bande1==True:
                audioss14=playsound(direc + '/repemasenter.mp3')
                respdida=input('''Si quieres repetir el ejemplo dos
                ↪ ingresa el signo
                +''')
                if respdida=="+":
                    print(definicion)
                    audiosdefiau=playsound(direc + '/ejem2.mp3')
                elif respdida != "+":
                    bande1=False



            Ejemplo3='''Ejemplo tres con números. Tenemos el conjunto
            ↪ A={1,3}
            y el conjunto B={4,7}.  Realizando la unión nos forma el
↪ conjunto
            C={1,3,4,7}. Como podemos ver el conjunto C tiene todos
↪ los elementos de
            ambos conjuntos'''
            audioseje3au=playsound(direc + '/ejem3.mp3')
            bande1=True
            while bande1==True:
                audioss14=playsound(direc + '/repemasenter.mp3')
                respdida=input('''Si quieres repetir el ejemplo tres
                ↪ ingresa el
                signo +''')
                if respdida=="+":
                    print(definicion)
                    audiosdefiau=playsound(direc + '/ejem3.mp3')
                elif respdida != "+":
                    bande1=False

            Ejemplo4='''Ejemplo 4. Tenemos el conjunto de W={1,3,7} y
            ↪ el conjunto X={1,2}
            realizando la unión nos forma el conjunto Y={1,2,3,7}.
↪ Notemos que en este
            ejemplo,  aunque ambos conjuntos tiene el elemento uno,
↪ el conjunto respuesta
            sólo lo tiene una vez. Recordemos que en un conjunto no
↪ importa el orden
            y si se repite el mismo elemento, representa sólo ese
↪ mismo elemento'''
            audioseje4au=playsound(direc + '/ejem4.mp3')
            bande1=True
            while bande1==True:
                audioss14=playsound(direc + '/repemasenter.mp3')
                respdida=input('''Si quieres repetir el ejemplo 4
                ↪ ingresa el signo +''')
                if respdida=="+":
                    print(definicion)
                    audiosdefiau=playsound(direc + '/ejem4.mp3')
                elif respdida != "+":
                    bande1=False

            Ejemplo5='''Ejemplo cinco. Unamos el conjunto vacío con
            ↪ el conjunto
            A={1,2}, del cual resulta el conjunto A, pues el conjunto
↪ vació no aporta
            elementos. '''
            audioseje5au=playsound(direc + '/ejem5.mp3')
            bande1=True
            while bande1==True:
                audioss14=playsound(direc + '/repemasenter.mp3')
                respdida=input('''Si quieres repetir el ejemplo cinco
                ↪ ingresa el
                signo +''')
                if respdida=="+":
                    print(definicion)
                    audiosdefiau=playsound(direc + '/ejem5.mp3')
                elif respdida != "+":
```

```
                bande1=False
            Ejemplo6='''Ejemplo seis. Se dice que la unión es
            ↪ asociativa y
            conmutativa. Hagamos un ejemplo con tu cuerpo. Supongamos
↪ que el conjunto
            A son tus dedos de la derecha y el conjunto B los dedos
↪ de la izquierda,
            si hacemos la unión de los dos conjuntos se obtienen los
↪ 10 dedos.
        Con el conjunto A y B tienes sólo tus dedos, pero te falta
↪ tu cuerpo,
        ahora pensemos. El conjunto C es tu cabeza y tus pies, y el
↪ conjunto D es el
        resto de tus partes no mencionadas. Si realizamos la
↪ operación de la unión,
        obtendremos el conjunto universo, en otras palabras, todas
↪ las partes de tu
        cuerpo. En este ejemplo podemos notar que el orden para
↪ realizar la unión,
        no afecta el resultado, pues podemos unir primero el
↪ conjunto D, el C y
        después el A y B y notaríamos que el resultado seria tu
↪ cuerpo.'''
            audioseje6au=playsound(direc + '/ejem6.mp3')
            bande1=True
            while bande1==True:
                audioss14=playsound(direc + '/repemasenter.mp3')
                respdida=input('''Si quieres repetir el ejemplo seis
                ↪ ingresa el
                signo +''')
                if respdida=="+":
                    print(definicion)
                    audiosdefiau=playsound(direc + '/ejem6.mp3')
                elif respdida != "+":
                    bande1=False
        banderae=True
        while banderae==True:
            if (elec=='d' or elec=='D'):
                print('''Has elegido tu orden, el uno corresponde a
                ↪ la definición,
                del dos al seis a los ejemplos, para salir aprieta
↪ cualquier otro
                número.''')
                audiosdefiau=playsound(direc + '/opcionD.mp3')
            if (elec=='a' or elec=='A'):
                repe='''Si deseas volver a escuchar la definición
                ↪ formal aprieta
                cero o del uno al seis para volver a escuchar algún
↪ ejemplo en
                concreto. Para ilustrar, si gustas escuchar el
↪ ejemplo tres aprieta
                el número tres. Para continuar
sin repetición presiona  cualquier otro numero. '''
                audiosdefiau=playsound(direc + '/repeejem.mp3')
            respdida=input('''Ingresa tu opción de ejemplos.''')
            if respdida=="0":
                audiosdefiau=playsound(direc + '/defi.mp3')
                print("0")
            elif respdida=="1":
                audioseje1au=playsound(direc + '/ejem1.mp3')
                print("0")
            elif respdida=="2":
                audioseje2au=playsound(direc + '/ejem2.mp3')
                print("0")
            elif respdida=="3":
                audioseje3au=playsound(direc + '/ejem3.mp3')
                print("0")
            elif respdida=="4":
                audioseje4au=playsound(direc + '/ejem4.mp3')
                print("0")
            elif respdida=="5":
                audioseje5au=playsound(direc + '/ejem5.mp3')
                print("0")
            elif respdida=="6":
                audioseje5au=playsound(direc + '/ejem6.mp3')
                print("0")
            else:
                print("presiona enter")
                banderae=False
        ejemplo_instrucciones='''Con la finalidad de que te
        ↪ familiarices más con el
        programa y el ingreso de respuestas realizaremos un ejercicio
↪ similar al del
        programa. Escucharas los conjuntos y después este sonido que
↪ te indicara que
        puedes ingresar tu respuestas. '''
        audiosresc=playsound(direc + '/ejeminstru.mp3')
        audiosresc=playsound(direc + '/didaintrudos.mp3')
        audiosresc=playsound(direc + '/didaintrudos2.mp3')
        audiosresc=playsound(direc + '/explorar.mp3')
        audioseje5au=playsound(direc + '/explirepe.mp3')
        conjeje1=set()
        conjeje=set()
        #llenar conjuntos
        print('''Cada que ingreses un elemento da un enter, es decir,
        ↪ si el conjunto
        consta de los elementos 1,2,3, entonces agrega el 1, enter, 2
↪ enter, 3 y
        enter.''')
        rellenacon(conjeje,0,5,2)
        rellenacon(conjeje1,0,5,2)
        conjdi=conjeje|conjeje1
        bandera=True
        while bandera==True:
            print(conjeje)
            audiosresc=playsound(direc + '/conjA.mp3')
            elemetosconj(conjeje)
            print(conjeje1)
            audiosresc=playsound(direc + '/conjB.mp3')
            elemetosconj(conjeje1)

            audioss11=playsound(direc + '/pide.mp3')
            print('''¿Cuál es la unión de los elementos de los
            ↪ conjuntos
            anteriores?''')
            conjres=set()
            conjres,nt,numr=respconj(conjres,
            ↪ len(conjdi),conjeje,conjeje1)
            if conjdi==conjres:
                bandera=False
                audioss11=playsound(direc + '/rescorre.mp3')
                print("Es correcta tu respuesta")
            else:
                if len(conjres)==len(conjdi):
                    audioss11=playsound(direc + '/resincorre.mp3')
                    print("vuelve a intentarlo")
                conjres.clear()
        audioss11=playsound(direc + '/evauni.mp3')
        banderae=True

        print('''Estás listo para iniciar las actividades de
        ↪ evaluación para la
        unión. Están divididas en cuatro niveles escoge el nivel que
↪ te gustaría
        practicar las veces que sean necesarias. ''')


##############---Nivel uno---#####################################
    elif menu=="1":
        print("Nivel uno")
        a1=time.time() #para tomar el tiempo completo

        print('''Bienvenido al nivel uno. En este ejercicio te
        ↪ aparecerán dos
        conjuntos. Debes realizar la operación de la unión. Para ello
↪ debes ingresar
        elemento por elemento con un enter''')

        ###################listas de las variables
        conjuntorespuesta=[]
        #1 conjunto respuesta
        tiempoxpreguntasin=[]
        #2 tiempo por pregunta sin repetición ni acumulado
        tiemacumupoxpreguntasin=[]
        #3 tiempo acumulado sin repetición ni acumulado
        tiemporepexpreguntacon=[]
        #4 tiempo por repeticiones por pregunta
        tiempoacumurepexpreguntacon=[]
        #5 tiempo por repeticiones aculudado
        tiempoexploestrella=[]
        #6 tiempo con exploración por pregunta
        tiempoexploestrellaacumulado=[]
        #7 tiempo acumulado con exploración
        repesumaestrella=[]
        #8 tiempo con repetición y exploración por pregunta
        repesumaestrellaacumulado=[]
        #9 tiempo acumulado con repetición y exploración
        numrepesuma=[]
        #10 número de repeticiones por pregunta
        numrepesumaacumu=[]
        #11 número de repeticiones acumualdo
        numexploxpreg=[]
        #12 numero de exploraciones por pregunta
        numexploacumu=[]
        #13 numero de exploraciones acumulada
        numsumaexploxpregunta=[]
        #14 numero de suma de repetición y exploración por pregunta
        numsumaexploacumu=[]
        #15 numero de suma de repetición y exploración acumulado
        incorrestas1=[]
        #16 errores por pregunta
        incoacumuladas=[]
        #17 errores por pregunta acumulada

        mal=0
        acuti=0
        acutire=0
        repemunacu=0
        ntacumu=0
        tsumarepeacumu=0
        ndexacumu=0
        ndexyrepeacumu=0
```

```
        audioss11=playsound(direc + '/instru1.mp3')
        audioss13=playsound(direc + '/nivel1.mp3')

        for i in range(numpreg):
            #crear dos conjuntos aleatorios de un elemento
            conj1=set()
            conj2=set()
            creadosconjdis(conj1,conj2,0,9,3)
            bandera=True
            conj3=set()
            conj4=conj1 | conj2
            c=conj4
            malacumu=0
            timecal=0
            repenum=0
            ntxp=0
            tsumarepe=0
            ndexp=0
            ndexyrepe=0
            while bandera==True:
                print("Une el conjunto A")
                audiosresc=playsound(direc + '/conjA.mp3')
                print(conj1)
                elemetosconj(conj1)
                print("Con el siguiente conjunto B")
                audiosresc=playsound(direc + '/conjB.mp3')
                print(conj2)
                print("Sonido de respuesta")
                elemetosconj(conj2)
                audioss11=playsound(direc + '/pide.mp3')
                a=time.time()
                conj3,nt,numr=respconj(conj3,len(conj4),conj1,conj2)
                b=time.time()
                cal=b-a-nt
                calre=0
                ntxp=ntxp+nt
                ntacumu=ntacumu+nt
                ndexp=ndexp+numr
                if conj3==conj4:
                    bandera=False
                    print("Respuesta correcta")
                    audiosresc=playsound(direc + '/rescorre.mp3')
                    acutire=acutire+timecal+cal
                    acuti=acuti+cal
                    calre=cal+timecal
                    tsumarepe=ntxp+calre
                    tsumarepeacumu=tsumarepeacumu+tsumarepe
                    ndexacumu=ndexacumu+ndexp
                    ndexyrepe=repenum+ndexp
                    ndexyrepeacumu=ndexyrepeacumu+ndexyrepe
                else:
                    if len(conj3)==len(conj4):
                        print("Vuelve a intentar")
                        audiosresin=playsound(direc +
                        '/resincorre.mp3')
                        mal=mal+1
                        malacumu=malacumu+1
                    timecal=timecal+cal
                    conj3.clear()
                    repenum=repenum+1
            nt=ntxp

            repemunacu=repemunacu+repenum
            conjuntorespuesta.append(c)
            tiempoxpreguntasin.append(cal)
            tiemporepexpreguntacon.append(calre)
            tiemacumupoxpreguntasin.append(acuti)
            tiempoacumurepexpreguntacon.append(acutire)
            tiempoexploestrella.append(nt)
            tiempoexploestrellaacumulado.append(ntacumu)
            repesumaestrella.append(tsumarepe)
            repesumaestrellaacumulado.append(tsumarepeacumu)
            numrepesuma.append(repenum)
            numrepesumaacumu.append(repemunacu)
            numexploxpreg.append(ndexp)
            numexploacumu.append(ndexacumu)
            numsumaexploxpregunta.append(ndexyrepe)
            numsumaexploacumu.append(ndexyrepeacumu)
            incorrestas1.append(mal)
            incoacumuladas.append(malacumu)

        print('''Como te diste cuenta uniste conjuntos con diferentes
        ↪  elementos pero
        ¿Qué pasa con los elementos en común? Lo veremos en los
↪   siguientes niveles
        ''')
        audiosresin=playsound(direc + '/retro1.mp3')
        promedio=(mal/numpreg)

        if mal==0:
            print("Increíble no te has equivocado")
            audiosresin=playsound(direc + '/increible.mp3')
        elif promedio>=.8:
            print("Buen trabajo")
        elif .8>promedio>=.6:
            print("Te sugerimos volver a realizar el nivel")
            audiosresin=playsound(direc + '/repenivel.mp3')
        else:
            print('''Te sugerimos volver a estudiar la actividad
            ↪  didáctica por el
            número de respuestas incorrectas''')
            audiosresin=playsound(direc + '/repeincorre.mp3')
        names=['Conjunto respuesta',
        'Tiempo final en contestar correctamente',
        'Tiempo final en contestar correctamente acumulado',
        'Tiempo sin exploración por pregunta',
        'Tiempo sin exploración acumulado',
        'Tiempo de exploración por pregunta',
        'Tiempo de exploración acumulado',
        'Tiempo con exploración por pregunta',
        'Tiempo con exploración acumulado',
        'Número de repeticiones por pregunta',
        'Número de repeticiones acumulado',
        'Número de exploraciones por pregunta',
        'Número de exploraciones acumulado',
        'Número de repeticiones más exploraciones por pregunta',
        'Número de repeticiones más exploraciones acumulado',
        'Errores por pregunta',
        'Errores acumulados']

        df=pd.DataFrame(list(zip(conjuntorespuesta,
                        tiempoxpreguntasin,
                        tiemacumupoxpreguntasin,
                        tiemporepexpreguntacon,
                        tiempoacumurepexpreguntacon,
                        tiempoexploestrella,
                        tiempoexploestrellaacumulado,
                        repesumaestrella,
                        repesumaestrellaacumulado,
                        numrepesuma, numrepesumaacumu,
                        numexploxpreg, numexploacumu,
                        numsumaexploxpregunta, numsumaexploacumu,
                        incorrestas1, incoacumuladas)), columns=names)

        print(df)
        df.to_csv(direc + '/CU1/dat1_U_.csv')

##############---Nivel dos---#####################################
    elif menu=="2":
        print("Nivel 2")
        audioss14=playsound(direc + '/nivel2.mp3')
        print("nivel 2")

        print('''En el nivel 2, encontraras conjuntos que tienen
        ↪  elementos en
        común. Debes realizar la unión. Recuerda en un conjunto A que
↪   tiene los
        elementos A={1,1,1,1} es igual al conjunto b={1}.''')

        audioss14=playsound(direc + '/instru2.mp3')

        ###################listas de las variables
        conjuntorespuesta=[]
        #1 conjunto respuesta
        tiempoxpreguntasin=[]
        #2 tiempo por pregunta sin repetición ni acumulado
        tiemacumupoxpreguntasin=[]
        #3 tiempo acumulado sin repetición ni acumulado
        tiemporepexpreguntacon=[]
        #4 tiempo por repeticiones por pregunta
        tiempoacumurepexpreguntacon=[]
        #5 tiempo por repeticiones aculudado
        tiempoexploestrella=[]
        #6 tiempo con exploración por pregunta
        tiempoexploestrellaacumulado=[]
        #7 tiempo acumulado con exploración
        repesumaestrella=[]
        #8 tiempo con repetición y exploración por pregunta
        repesumaestrellaacumulado=[]
        #9 tiempo acumulado con repetición y exploración
        numrepesuma=[]
        #10 número de repeticiones por pregunta
        numrepesumaacumu=[]
        #11 número de repeticiones acumualdo
        numexploxpreg=[]
        #12 numero de exploraciones por pregunta
        numexploacumu=[]
        #13 numero de exploraciones acumulada
        numsumaexploxpregunta=[]
        #14 numero de suma de repetición y exploración por pregunta
        numsumaexploacumu=[]
        #15 numero de suma de repetición y exploración acumulado
        incorrestas1=[]
        #16 errores por pregunta
        incoacumuladas=[]
        #17 errores por pregunta acumulada

        mal=0
        acuti=0
        acutire=0
        repemunacu=0
        ntacumu=0
```

```
        tsumarepeacumu=0
        ndexacumu=0
        ndexyrepeacumu=0
        for i in range(numpreg):
            #print("Une los siguientes conjuntos")
            conj1=set()
            print("conjunto A")
            rellenacon(conj1,0,5,2)
            conj2=set()
            print("conjunto B")
            rellenacon(conj2,0,5,2)
            conj3=set()
            malacumu=0
            timecal=0
            repenum=0
            ntxp=0
            ndexp=0
            tsumarepe=0
            ndexyrepe=0
            conj3=conj1 | conj2
            c=conj3
            bandera=True
            conj4=set()
            while bandera==True:
                print("Une los conjuntos siguientes:")
                audioss14=playsound(direc + '/conjA.mp3')
                print(conj1)
                elemetosconj(conj1)
                audioss14=playsound(direc + '/conjB.mp3')
                print(conj2)
                print("Con el siguiente conjunto")
                elemetosconj(conj2)
                audioss11=playsound(direc + '/pide.mp3')
                a=time.time()
                conj4,nt,numr=respconj(conj4,len(conj3),conj1,conj2)
                b=time.time()
                cal=b-a-nt
                calre=0
                ntxp=+ntxp+nt
                ntacumu=ntacumu+nt
                ndexp=ndexp+numr
                if conj3==conj4:
                    bandera=False
                    audioss14=playsound(direc + '/rescorre.mp3')
                    acutire=acutire+timecal+cal
                    acuti=acuti+cal
                    calre=cal+timecal
                    tsumarepe=ntxp+calre
                    tsumarepeacumu=tsumarepeacumu+tsumarepe
                    ndexacumu=ndexacumu+ndexp
                    ndexyrepe=repenum+ndexp
                    ndexyrepeacumu=ndexyrepeacumu+ndexyrepe
                else:
                    if len(conj3)==len(conj4):
                        print("Vuelve a intentar")
                        audiosresin=playsound(direc +
                        ↪  '/resincorre.mp3')
                        mal=mal+1
                        malacumu=malacumu+1
                    conj4.clear()
                    timecal=timecal+cal
                    repenum=repenum+1
                    print("Se repetira el ejercicio")

            repemunacu=repemunacu+repenum
            conjuntorespuesta.append(c)
            tiempoxpreguntasin.append(cal)
            tiemporepexpreguntacon.append(calre)
            tiemacumupoxpreguntasin.append(acuti)
            tiempoacumurepexpreguntacon.append(acutire)
            tiempoexploestrella.append(nt)
            tiempoexploestrellaacumulado.append(ntacumu)
            repesumaestrella.append(tsumarepe)
            repesumaestrellaacumulado.append(tsumarepeacumu)
            numrepesuma.append(repenum)
            numrepesumaacumu.append(repemunacu)
            numexploxpreg.append(ndexp)
            numexploacumu.append(ndexacumu)
            numsumaexploxpregunta.append(ndexyrepe)
            numsumaexploacumu.append(ndexyrepeacumu)
            incorrestas1.append(mal)
            incoacumuladas.append(malacumu)


        print('''Como observaste, en los conjuntos no importa el
        ↪  orden ni la repetición
        de los elementos. Por último, ¿Existirán los conjuntos
↪  infinitos? La respuesta
        es sí, un ejemplo de ellos son los números naturales,
↪  1,2,3,,5,etc. ''')
        audiosresin=playsound(direc + '/retro2.mp3')
        promedio=(mal/numpreg)
        if mal==0:
            print("Increíble no te has equivocado")
            audiosresin=playsound(direc + '/increible.mp3')
        elif promedio>=.8:
            print("Buen trabajo")
        elif .8>promedio>=.6:
            print("Te sugerimos volver a realizar el nivel")
            audiosresin=playsound(direc + '/repenivel.mp3')
        else:
            print('''te sugerimos volver a estudiar la actividad
            ↪  didáctica por el
            número de respuestas incorrectas''')
            audiosresin=playsound(direc + '/repeincorre.mp3')


        names=['Conjunto respuesta',
        'Tiempo final en contestar correctamente',
        'Tiempo final en contestar correctamente acumulado',
        'Tiempo sin exploración por pregunta',
        'Tiempo sin exploración acumulado',
        'Tiempo de exploración por pregunta',
        'Tiempo de exploración acumulado',
        'Tiempo con exploración por pregunta',
        'Tiempo con exploración acumulado',
        'Número de repeticiones por pregunta',
        'Número de repeticiones acumulado',
        'Número de exploraciones por pregunta',
        'Número de exploraciones acumulado',
        'Número de repeticiones más exploraciones por pregunta',
        'Número de repeticiones más exploraciones acumulado',
        'Errores por pregunta',
        'Errores acumulados']

        df=pd.DataFrame(list(zip(conjuntorespuesta,
                        tiempoxpreguntasin,
                        tiemacumupoxpreguntasin,
                        tiemporepexpreguntacon,
                        tiempoacumurepexpreguntacon,
                        tiempoexploestrella,
                        tiempoexploestrellaacumulado,
                        repesumaestrella,
                        repesumaestrellaacumulado,
                        numrepesuma, numrepesumaacumu,
                        numexploxpreg, numexploacumu,
                        numsumaexploxpregunta, numsumaexploacumu,
                        incorrestas1, incoacumuladas)), columns=names)
        df.to_csv(direc + '/CU2/dat2_U_.csv')


    ##############---Nivel
↪  tres---########################################
    elif menu=="3":
        print("Nivel 3")
        audioss14=playsound(direc + '/nivel3.mp3')
        audioss14=playsound(direc + '/instru3.mp3')
        print('''En este nivel te darás cuenta de que la unión de los
        ↪  conjuntos
        es conmutativo, es decir, que es lo mismo hacer AUB que BUA
↪  aunque parezca
        repetitivo el ejercicio, es importante notar esta propiedad
↪  en los conjuntos
        pues no todas las operaciones son conmutativas''')

        ##################listas de las variables
        conjuntorespuesta=[]
        #1 conjunto respuesta
        tiempoxpreguntasin=[]
        #2 tiempo por pregunta sin repetición ni acumulado
        tiemacumupoxpreguntasin=[]
        #3 tiempo acumulado sin repetición ni acumulado
        tiemporepexpreguntacon=[]
        #4 tiempo por repeticiones por pregunta
        tiempoacumurepexpreguntacon=[]
        #5 tiempo por repeticiones aculudado
        tiempoexploestrella=[]
        #6 tiempo con exploración por pregunta
        tiempoexploestrellaacumulado=[]
        #7 tiempo acumulado con exploración
        repesumaestrella=[]
        #8 tiempo con repetición y exploración por pregunta
        repesumaestrellaacumulado=[]
        #9 tiempo acumulado con repetición y exploración
        numrepesuma=[]
        #10 número de repeticiones por pregunta
        numrepesumaacumu=[]
        #11 número de repeticiones acumualdo
        numexploxpreg=[]
        #12 numero de exploraciones por pregunta
        numexploacumu=[]
        #13 numero de exploraciones acumulada
        numsumaexploxpregunta=[]
        #14 numero de suma de repetición y exploración por pregunta
        numsumaexploacumu=[]
        #15 numero de suma de repetición y exploración acumulado
        incorrestas1=[]
        #16 errores por pregunta
        incoacumuladas=[]
        #17 errores por pregunta acumulada

        mal=0
```

```
acuti=0
acutire=0
repemunacu=0
ntacumu=0
tsumarepeacumu=0
ndexacumu=0
ndexyrepeacumu=0
for k in range(numpreg):
    conj1=set()
    conj2=set()
    rellenacon(conj1,0,7,3)
    rellenacon(conj2,0,7,3)
    conj4=conj1 | conj2
    c=conj4
    bandera=True
    malacumu=0
    timecal=0
    repenum=0
    ntxp=0
    ndexp=0
    tsumarepe=0
    ndexyrepe=0
    conj3=set()
    while bandera==True:
        print("Une el conjunto ")
        audioss14=playsound(direc + '/conjA.mp3')
        print(conj1)
        elemetosconj(conj1)
        audioss14=playsound(direc + '/conjB.mp3')
        print("Con")
        print(conj2)
        elemetosconj(conj2)
        print("Ingresa los elementos uno por uno")
        audioss11=playsound(direc + '/pide.mp3')
        a=time.time()
        conj3,nt,numr=respconj(conj3,len(conj4),conj1,conj2)
        b=time.time()
        cal=b-a-nt
        calre=0
        ntxp=+ntxp+nt
        ntacumu=ntacumu+nt
        ndexp=ndexp+numr
        if conj3==conj4:
            bandera=False
            print("correcto")
            audiosresc=playsound(direc + '/rescorre.mp3')
            acutire=acutire+timecal+cal
            acuti=acuti+cal
            calre=cal+timecal
            tsumarepe=ntxp+calre
            tsumarepeacumu=tsumarepeacumu+tsumarepe
            ndexacumu=ndexacumu+ndexp
            ndexyrepe=repenum+ndexp
            ndexyrepeacumu=ndexyrepeacumu+ndexyrepe
        else:
            if len(conj3)==len(conj4):
                print("Vuelve a intentar")
                audiosresin=playsound(direc +
                '/resincorre.mp3')
                malacumu=malacumu+1
                mal=mal+1
            conj3.clear()
            timecal=timecal+cal
            conj3.clear()
            repenum=repenum+1
            print("Vuelve a intentar")

    conj3=set()
    print('''Usemos la  conmutatividad. cambiemos el orden de
    los conjuntos''')
    audiosresc=playsound(direc + '/nive3ex.mp3')
    bandera=True
    while bandera==True:
        print("une el conjunto ")
        print(conj2)
        audioss14=playsound(direc + '/conjB1.mp3')
        elemetosconj(conj2)
        print("con")
        print(conj1)
        audioss14=playsound(direc + '/conjA2.mp3')
        elemetosconj(conj1)
        print("ingresa los elementos 1 por 1")
        audioss11=playsound(direc + '/pide.mp3')
        a=time.time()
        conj3, nt, numr= respconj(conj3, len(conj4),\
        conj2,conj1)
        b=time.time()
        cal=b-a-nt
        calre=0
        ntxp=+ntxp+nt
        ntacumu=ntacumu+nt
        ndexp=ndexp+numr
        if conj3==conj4:
            bandera=False
            print("Correcto")
            audiosresc=playsound(direc + '/rescorre.mp3')
            acutire=acutire+timecal+cal
            acuti=acuti+cal
            calre=cal+timecal
            tsumarepe=ntxp+calre
            tsumarepeacumu=tsumarepeacumu+tsumarepe
            ndexacumu=ndexacumu+ndexp
            ndexyrepe=repenum+ndexp
            ndexyrepeacumu=ndexyrepeacumu+ndexyrepe
        else:
            if len(conj3)==len(conj4):
                print("vuelve a intentar")
                audiosresin=playsound(direc +
                '/resincorre.mp3')
                mal=mal+1
                malacumu=malacumu+1
            conj3.clear()
            timecal=timecal+cal
            repenum=repenum+1
            print("Se repetira el ejercicio")

    repemunacu=repemunacu+repenum
    conjuntorespuesta.append(c)
    tiempoxpreguntasin.append(cal)
    tiemporepexpreguntacon.append(calre)
    tiemacumupoxpreguntasin.append(acuti)
    tiempoacumurepexpreguntacon.append(acutire)
    tiempoexploestrella.append(nt)
    tiempoexploestrellaacumulado.append(ntacumu)
    repesumaestrella.append(tsumarepe)
    repesumaestrellaacumulado.append(tsumarepeacumu)
    numrepesuma.append(repenum)
    numrepesumaacumu.append(repemunacu)
    numexploxpreg.append(ndexp)
    numexploacumu.append(ndexacumu)
    numsumaexploxpregunta.append(ndexyrepe)
    numsumaexploacumu.append(ndexyrepeacumu)
    incorrestas1.append(mal)
    incoacumuladas.append(malacumu)

audioss14=playsound(direc + '/felicidades.mp3')
df=pd.DataFrame()

names=['Conjunto respuesta',
'Tiempo final en contestar correctamente',
'Tiempo final en contestar correctamente acumulado',
'Tiempo sin exploración por pregunta',
'Tiempo sin exploración acumulado',
'Tiempo de exploración por pregunta',
'Tiempo de exploración acumulado',
'Tiempo con exploración por pregunta',
'Tiempo con exploración acumulado',
'Número de repeticiones por pregunta',
'Número de repeticiones acumulado',
'Número de exploraciones por pregunta',
'Número de exploraciones acumulado',
'Número de repeticiones más exploraciones por pregunta',
'Número de repeticiones más exploraciones acumulado',
'Errores por pregunta',
'Errores acumulados']

df=pd.DataFrame(list(zip(conjuntorespuesta,
                tiempoxpreguntasin,
                tiemacumupoxpreguntasin,
                tiemporepexpreguntacon,
                tiempoacumurepexpreguntacon,
                tiempoexploestrella,
                tiempoexploestrellaacumulado,
                repesumaestrella,
                repesumaestrellaacumulado,
                numrepesuma, numrepesumaacumu,
                numexploxpreg, numexploacumu,
                numsumaexploxpregunta, numsumaexploacumu,
                incorrestas1, incoacumuladas)), columns=names)

df.to_csv(direc + '/CU3/dat3_U_.csv')
print('''En La unión no importa el orden de la operación, si
primero A o B
o C, sino unirlos''')
promedio=(mal/numpreg)
if mal==0:
    print("Increíble no te has equivocado")
    audiosresin=playsound(direc + '/increible.mp3')
elif promedio>=.8:
    print("Buen trabajo")
elif .8>promedio>=.6:
    print("Te sugerimos volver a realizar el nivel")
    audiosresin=playsound(direc + '/repenivel.mp3')
else:
    print("Te sugerimos volver a estudiar la actividad
    didáctica por el numero
    de respuestas incorrectas")
    audiosresin=playsound(direc + '/repeincorre.mp3')
audiosresin=playsound(direc + '/retro3.mp3')

print(df)
```

```
##############---Nivel cuatro---####################################
    elif menu=="4":
        print("Nivel 4")
        #nivel 4 conjuntos de 3 elementos
        audioss15=playsound(direc + '/nivel4.mp3')
        audioss14=playsound(direc + '/instru4.mp3')
        print('''en este nivel aprenderás que la unión se puede hacer
        ↪  con varios
        conjuntos a la vez. En ese sentido, si unes todos los
↪  conjuntos  obtendrás el
        conjunto universo. También aprenderás que la unión es
↪  asociativa, es decir
        (AUB)UC es igual a (A)U(BUC)''')

        ##################listas de las variables
        conjuntorespuesta=[]
        #1 conjunto respuesta
        tiempoxpreguntasin=[]
        #2 tiempo por pregunta sin repetición ni acumulado
        tiemacumupoxpreguntasin=[]
        #3 tiempo acumulado sin repetición ni acumulado
        tiemporepexpreguntacon=[]
        #4 tiempo por repeticiones por pregunta
        tiempoacumurepexpreguntacon=[]
        #5 tiempo por repeticiones aculudado
        tiempoexploestrella=[]
        #6 tiempo con exploración por pregunta
        tiempoexploestrellaacumulado=[]
        #7 tiempo acumulado con exploración
        repesumaestrella=[]
        #8 tiempo con repetición y exploración por pregunta
        repesumaestrellaacumulado=[]
        #9 tiempo acumulado con repetición y exploración
        numrepesuma=[]
        #10 número de repeticiones por pregunta
        numrepesumaacumu=[]
        #11 número de repeticiones acumualdo
        numexploxpreg=[]
        #12 numero de exploraciones por pregunta
        numexploacumu=[]
        #13 numero de exploraciones acumulada
        numsumaexploxpregunta=[]
        #14 numero de suma de repetición y exploración por pregunta
        numsumaexploacumu=[]
        #15 numero de suma de repetición y exploración acumulado
        incorrestas1=[]
        #16 errores por pregunta
        incoacumuladas=[]
        #17 errores por pregunta acumulada

        mal=0
        acuti=0
        acutire=0
        repemunacu=0
        ntacumu=0
        tsumarepeacumu=0
        ndexacumu=0
        ndexyrepeacumu=0
        for k in range(numpreg):
            conj1=set()
            conj2=set()
            conj3=set()
            rellenacon(conj1,0,7,2)
            rellenacon(conj2,0,7,2)
            rellenacon(conj3,0,7,2)
            conj4=conj1 | conj2 | conj3
            c=conj4
            bandera=True
            malacumu=0
            timecal=0
            repenum=0
            ntxp=0
            ndexp=0
            tsumarepe=0
            ndexyrepe=0
            while bandera==True:
                print("Une el conjunto ")
                audioss14=playsound(direc + '/conjA.mp3')
                print(conj1)
                elemetosconj(conj1)
                print("con")
                audioss14=playsound(direc + '/conjB.mp3')
                print(conj2)
                elemetosconj(conj2)
                print("con")
                audioss14=playsound(direc + '/conjC.mp3')
                print(conj3)
                elemetosconj(conj3)
                audioss11=playsound(direc + '/pide.mp3')
                conj5=set()
                a=time.time()

                conj5,nt,numr=respconj3(conj5,len(conj4),conj1,conj2,conj3)
                #falta el
                cojunto 3
                b=time.time()
                cal=b-a-nt
                calre=0
                ntxp=+ntxp+nt
                ntacumu=ntacumu+nt
                ndexp=ndexp+numr
                if conj5==conj4:
                    bandera=False
                    print("correcto")
                    audiosresc=playsound(direc + '/rescorre.mp3')
                    acutire=acutire+timecal+cal
                    acuti=acuti+cal
                    calre=cal+timecal
                    tsumarepe=ntxp+calre
                    tsumarepeacumu=tsumarepeacumu+tsumarepe
                    ndexacumu=ndexacumu+ndexp
                    ndexyrepe=repenum+ndexp
                    ndexyrepeacumu=ndexyrepeacumu+ndexyrepe
                else:
                    if len(conj5)==len(conj4):
                        print("Vuelve a intentar")
                        audiosresin=playsound(direc +
                        ↪  '/resincorre.mp3')
                        mal=mal+1
                        malacumu=malacumu+1
                    conj5.clear()
                    timecal=timecal+cal
                    repenum=repenum+1

                    print("Se volverá a repetir")

            repemunacu=repemunacu+repenum
            conjuntorespuesta.append(c)
            tiempoxpreguntasin.append(cal)
            tiemporepexpreguntacon.append(calre)
            tiemacumupoxpreguntasin.append(acuti)
            tiempoacumurepexpreguntacon.append(acutire)
            tiempoexploestrella.append(nt)
            tiempoexploestrellaacumulado.append(ntacumu)
            repesumaestrella.append(tsumarepe)
            repesumaestrellaacumulado.append(tsumarepeacumu)
            numrepesuma.append(repenum)
            numrepesumaacumu.append(repemunacu)
            numexploxpreg.append(ndexp)
            numexploacumu.append(ndexacumu)
            numsumaexploxpregunta.append(ndexyrepe)
            numsumaexploacumu.append(ndexyrepeacumu)
            incorrestas1.append(mal)
            incoacumuladas.append(malacumu)

        audioss14=playsound(direc + '/felicidades.mp3')
        df=pd.DataFrame()

        names=['Conjunto respuesta',
        'Tiempo final en contestar correctamente',
        'Tiempo final en contestar correctamente acumulado',
        'Tiempo sin exploración por pregunta',
        'Tiempo sin exploración acumulado',
        'Tiempo de exploración por pregunta',
        'Tiempo de exploración acumulado',
        'Tiempo con exploración por pregunta',
        'Tiempo con exploración acumulado',
        'Número de repeticiones por pregunta',
        'Número de repeticiones acumulado',
        'Número de exploraciones por pregunta',
        'Número de exploraciones acumulado',
        'Número de repeticiones más exploraciones por pregunta',
        'Número de repeticiones más exploraciones acumulado',
        'Errores por pregunta',
        'Errores acumulados']

        df=pd.DataFrame(list(zip(conjuntorespuesta,
                    tiempoxpreguntasin,
                    tiemacumupoxpreguntasin,
                    tiemporepexpreguntacon,
                    tiempoacumurepexpreguntacon,
                    tiempoexploestrella,
                    tiempoexploestrellaacumulado,
                    repesumaestrella,
                    repesumaestrellaacumulado,
                    numrepesuma, numrepesumaacumu,
                    numexploxpreg, numexploacumu,
                    numsumaexploxpregunta, numsumaexploacumu,
                    incorrestas1, incoacumuladas)), columns=names)

        df.to_csv(direc + '/CU4/dat4_U_.csv')
        print('''La unión es como hacer una mezcla solida, no importa
        ↪  el orden de
        los ingredientes siempre dará el mismo resultado''')
        audiosresin=playsound(direc + '/retro4.mp3')
        promedio=(mal/numpreg)
        if mal==0:
            print("Increible no te has equivocado")
            audiosresin=playsound(direc + '/increible.mp3')
        elif promedio>=.8:
            print("Buen trabajo")
        elif .8>promedio>=.6:
```

```
            print("Te sugerimos volver a realizar el nivel")
            audiosresin=playsound(direc + '/repenivel.mp3')
        else:
            print('''Te sugerimos volver a estudiar la actividad
            didáctica por el
            numero de respuestas incorrectas''')
            audiosresin=playsound(direc + '/repeincorre.mp3')

##############---Práctica para ingresar respuestas---###############
    elif menu=='5':

        ejemplo_instrucciones='''Con la finalidad de que te
        familiarices más con el
        programa y el ingreso de respuestas, realizaremos un
ejercicio similar al del
        programa. Escucharas los conjuntos y después este sonido que
te indicara que
        puedes ingresar tu respuestas. '''
        audiosresc=playsound(direc + '/ejeminstru.mp3')
        audiosresc=playsound(direc + '/didaintrudos.mp3')
        audiosresc=playsound(direc + '/didaintrudos2.mp3')
        audiosresc=playsound(direc + '/explorar.mp3')
        audioseje5au=playsound(direc + '/explirepe.mp3')
        conjeje1=set()
        conjeje=set()
        print('''cada que ingreses un elemento da un enter, es decir,
        si el conjunto
        consta de los elementos 1,2,3, entonces agrega el 1, enter, 2
 enter, 3 y
        enter.''')
        rellenacon(conjeje,0,5,2)
        rellenacon(conjeje1,0,5,2)
        conjdi=conjeje|conjeje1
        bandera=True
        while bandera==True:
            print(conjeje)
            audiosresc=playsound(direc + '/conjA.mp3')
            elemetosconj(conjeje)
            print(conjeje1)
            audiosresc=playsound(direc + '/conjB.mp3')
            elemetosconj(conjeje1)
            audioss11=playsound(direc + '/pide.mp3')
            print('''¿Cuál es la unión de los elementos de los
            conjuntos
            anteriores?''')
            conjres=set()
            conjres,nt,numr=respconj(conjres,
            len(conjdi),conjeje,conjeje1)
            if conjdi==conjres:
                bandera=False
                audioss11=playsound(direc + '/rescorre.mp3')
                print("Es correcta tu respuesta")
            else:
                if len(conjres)==len(conjdi):
                    audioss11=playsound(direc + '/resincorre.mp3')
                    print("Vuelve a intentarlo")
                conjres.clear()

        audioss11=playsound(direc + '/evauni.mp3')
        banderae=True

##############---Salir del programa---##############################

    elif menu=="6":
        audioss14=playsound(direc + '/adios.mp3')
        print("Salida")
        break

##############---Opción valida---###################################
    else:
        menu="7"
        print("tu opción no es valida, ")
        audioseje5au=playsound(direc + '/novalida.mp3')
        audioss14=playsound(direc + '/menu.mp3')
        audioss14=playsound(direc + '/menu1.mp3')
##############---Fin del la
 unión---#################################
```

A.2. Intersección

```
#Este código es una actividad didáctica para la enseñanza y la
↪  evaluación de la intersección
entre conjuntos.
import time
from datetime import datetime
import pandas as pd
import numpy as np
from playsound import playsound
from tkinter import Tk, Label, Button
import random as rd
import os
ruta = os.getcwd()


#Funciones:
def LLENARCONJUNTOS(conj, a, b): #Función para llenar los conjuntos.
  for m in range(a):
    x=rd.randint(0,b)
    conj.add(str(x))

def DICTACONJUNTO (conj):
  for i in conj: #Mencionamos los elementos del conjunto.
    if i=='0':
      sound0 = playsound(ruta+'/0.mp3')
    if i=='1':
      sound1 = playsound(ruta+'/1.mp3')
    if i=='2':
      sound2 = playsound(ruta+'/2.mp3')
    if i=='3':
      sound3 = playsound(ruta+'/3.mp3')
    if i=='4':
      sound4 = playsound(ruta+'/4.mp3')
    if i=='5':
      sound5 = playsound(ruta+'/5.mp3')
    if i=='6':
      sound6 = playsound(ruta+'/6.mp3')
    if i=='7':
      sound7 = playsound(ruta+'/7.mp3')
    if i=='8':
      sound8 = playsound(ruta+'/8.mp3')
    if i=='9':
      sound9 = playsound(ruta+'/9.mp3')

def DICTANUMERO (n):
  ConjRetro=set()
  ConjRetro.add(n)
  DICTACONJUNTO(ConjRetro)

def EXPLORACONJUNTOS (listA, listB, n):
  mov='a'
  i=0
  if (n==1):
    print('Estás en el Conjunto A')
    soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')
    print('''Conjunto A.''')
  elif (n==2):
    print('Estás en el Conjunto B')
    soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')
    print('''Conjunto B.''')
  while(mov!='-'):
    print(listA[i])
    DICTANUMERO(listA[i])
    mov=input('Movimiento:')
    if (mov=='d' or mov=='D'):
      if (i<(len(listA)-1)):
        i+=1
      else:
        i=0
    elif (mov=='a' or mov=='A'):
      if (i==0):
        i=len(listA)-1
      else:
        i-=1
    elif (mov=='s' or mov=='S'):
      if (n==1):
        EXPLORACONJUNTOS (listB, listA, 2)
        break
      else:
        EXPLORACONJUNTOS (listB, listA, 1)
        break

def EXPLORACONJUNTOS2 (listA, listB, listC, n):
  mov='a'
  i=0
  if (n==1):
    print('Estás en el Conjunto A')
    soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')
    print('''Conjunto A.''')
  elif (n==2):
    print('Estás en el Conjunto B')
    soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')
    print('''Conjunto B.''')
  elif (n==3):
    print('Estás en el Conjunto C')
    soundconjuntoC = playsound(ruta+'/ConjuntoC.mp3')
    print('''Conjunto C.''')
  while(mov!='-'):
    print(listA[i])
    DICTANUMERO(listA[i])
    mov=input('Movimiento:')
    if (mov=='d' or mov=='D'):
      if (i<(len(listA)-1)):
        i+=1
      else:
        i=0
    elif (mov=='a' or mov=='A'):
      if (i==0):
        i=len(listA)-1
      else:
        i-=1
    elif (mov=='s' or mov=='S'):
      if (n==1):
        EXPLORACONJUNTOS2 (listB, listC, listA, 2)
        break
      elif (n==2):
        EXPLORACONJUNTOS2 (listB, listC, listA, 3)
        break
      else:
        EXPLORACONJUNTOS2 (listB, listC, listA, 1)
        break

#Variables globales
npreg0=1
npreg1=10
npreg2=10
npreg3=10
npreg4=10


#Bienvenida al programa (Audios):
soundbienvenida = playsound(ruta+'/Bienvenida.mp3')
print('''Bienvenido al programa de actividades didácticas para la
↪  enseñanza y evaluación de
la intersección de dos conjuntos.''')
sounddefinicionformal = playsound(ruta+'/DefinicionFormal.mp3')
print('''Definición Formal: La intersección de dos conjuntos es una
↪  operación que resulta
en otro conjunto que contiene los elementos comunes a los conjuntos
↪  de partida.''')
soundexplicacion = playsound(ruta+'/Explicacion.mp3')
print('''Este programa cuenta con 5 actividades diferentes, una de
↪  ellas tiene la finalidad de
enseñarte, y las otras cuatro son evaluaciones que van aumentando su
↪  dificultad e incluyendo
cada vez más conceptos. ''')
soundenter = playsound(ruta+'/Enter.mp3')
print('''Para interactuar con este programa deberás teclear Enter
↪  después de cada número.''')


#Menú de actividades:
act='0'
while (act!='6'):
  soundmenu = playsound(ruta+'/Menu.mp3')
  print('''Presiona cero para la actividad de aprendizaje, uno para
  ↪  la evaluación en su nivel
  más básico, dos para la evaluación del nivel básico incluyendo
  ↪  conjuntos vacíos, tres
  para la evaluación de la conmutatividad, cuatro para la evaluación
  ↪  con tres conjuntos,
  cinco para realizar ejercicios de práctica, seis para salir.''')
  act=input('Opción Menú: ')
  if act=='0':
    print('Actividad Didáctica')
    sounddidactica1 = playsound(ruta+'/Didactica1.mp3')
    print('''Haz ingresado a la actividad didáctica de la
    ↪  intersección. En esta parte te
    enseñaremos cómo funciona, te daremos algunos ejemplos y te
    ↪  guiaremos a que realices
    algunos ejercicios similares a las actividades de evaluación.''')
    soundmenudidac = playsound(ruta+'/Didactica1_1.mp3')
    print('''Si quieres estudiar la actividad didáctica de manera
    ↪  fluida presiona 'a' seguido de
    un enter, o si prefieres poder moverte entre las diferentes
    ↪  secciones presiona 'd' y enter.''')
    menudidac=input('Fluidez didáctica:')
    didac=''
    while(didac=='' or didac=='1' or didac=='2' or didac=='3' or
    ↪  didac=='4'):
      if (menudidac=='d'):
```

```
        print ('Avance por partes')
        soundrep1 = playsound(ruta+'/Rep1.mp3')
        print('''Presiona uno para la definición informal, dos para los ejemplos en la vida
        cotidiana, tres para las instrucciones de los ejercicios, o cuatro para los ejercicios
        prácticos. Cualquier otro número para salir.''')
        didac=input('Opción menú didáctica:')
      else:
        print ('Avance fluido')
      if(didac=='' or didac=='1'):
        print ('Definición Informal')
        sounddidactica2 = playsound(ruta+'/Didactica2.mp3')
        print('''La intersección es una operación que 'revisa' cuales elementos del primer
        conjunto, también están en el segundo conjunto, es decir, que comparten las propiedades
        de ambos conjuntos.''')
      if(didac=='' or didac=='2'):
        print ('Ejemplos')
        ejemp='0'
        if (menudidac=='d'):
          soundselecejem = playsound(ruta+'/Didactica3_1.mp3')
          print('''Hay cinco ejemplos, presiona el número de ejemplo que quieres escuchar
          seguido de un enter o cero si quieres escucharlos todos.''')
          ejemp=input('Número de ejemplo:')
        if (ejemp=='0' or ejemp=='1'):
          print ('Ejemplo 1')
          sounddidactica3 = playsound(ruta+'/Didactica3.mp3')
          print('''Ejemplo 1: El conjunto A tendrá la propiedad que serán todas las personas a
          las que les gusta lo dulce, y el conjunto B tendrá la propiedad que serán todas las
          personas a las que les gusta lo salado. Como sabrás, hay personas a las que les gusta
          tanto lo dulce como lo salado, o sea, que hay personas que cumplen con las propiedades
          de los dos conjuntos, y son estas las que conforman el conjunto intersección.''')
        if (ejemp=='0' or ejemp=='2'):
          print ('Ejemplo 2')
          sounddidactica4 = playsound(ruta+'/Didactica4.mp3')
          print('''Ejemplo 2: El conjunto A tendrá la propiedad que serán todas las personas
          a las que les gusta el rock, y el conjunto B tendrá la propiedad que serán todas
          las personas a las que les gustan los boleros. Como sabrás, hay personas a las que
          les gusta el Rock y también les gustan los boleros, es decir, que hay personas a las
          que les gustan ambas cosas, o visto de otra manera, que cumplen con las propiedades
          de los dos conjuntos, y son estas las que conforman el conjunto intersección.''')
        if (ejemp=='0' or ejemp=='3'):
          print ('Ejemplo 3')
          sounddidactica5 = playsound(ruta+'/Didactica5.mp3')
          print('''Ejemplo 3: Ahora revisemos un caso donde la intersección sea vacía. Es
          decir, que no hay ningún elemento que cumpla las propiedades del conjunto A que
          también cumpla con las propiedades del conjunto B. Pongamos que el conjunto A tendrá
          la propiedad que serán todas las personas que saben Braille, y el conjunto B tendrá
          la propiedad que serán todas las personas que no saben Braille. Como podrás notar,
          no hay manera que una persona sepa Braille y no sepa Braille al mismo tiempo, es decir,
          que no hay elemento que cumpla con las propiedades de ambos conjuntos al mismo tiempo,
          por lo tanto, la intersección es vacía.''')
        if (ejemp=='0' or ejemp=='4'):
          print ('Ejemplo 4')
          sounddidactica6 = playsound(ruta+'/Didactica6.mp3')
          print('''Ejemplo 4: Aquí vamos a observar que la operación intersección es
          conmutativa, esto quiere decir que, no importa si hacemos la operación intersección del
          conjunto A con el conjunto B, o si la hacemos al revés, que hagamos la intersección
          del conjunto B con el conjunto A, porque nos da el mismo resultado: Si el conjunto
          A tiene la propiedad que son todas las personas que tienen más de 25 años, y el
          conjunto B tiene la propiedad que son todas las personas que tienen hijos, el conjunto
          intersección del conjunto A con el conjunto B, son todas las personas que tienen
          más de 25 años y tienen hijos; pero si hacemos la intersección en diferente orden,
          es decir, intersectamos el conjunto B con el conjunto A, nos daremos cuenta que es
          lo mismo, también son todas las personas que tienen más de 25 años y tienen hijos.''')
        if (ejemp=='0' or ejemp=='5'):
          print ('Ejemplo 5')
          sounddidactica7 = playsound(ruta+'/Didactica7.mp3')
          print('''Ejemplo 5: Veamos un caso en el que haremos la intersección de tres conjuntos:
          Pongamos el conjunto A con la propiedad de que son todas las personas que trabajan, el
          conjunto B con la propiedad de que son todas las personas que estudian la universidad,
          y el conjunto C con la propiedad que son todas las personas que hacen ejercicio,
          el resultado serán todas las personas que cumplen las tres propiedades, o sea,
          todas las personas que trabajan, van a la universidad y hacen ejercicio.''')
      if(didac=='' or didac=='3'):
        print ('Instrucciones')
        sounddidactica8 = playsound(ruta+'/Didactica8.mp3')
        print('''A continuación te guiaremos a realizar algunos ejercicios similares a las
        actividades de evaluación, dándote las respuestas, para que te vayas familiarizando. Cabe
        aclarar que para las actividades de evaluación solo utilizarás las teclas del 0 al
        9 y el Enter. Si respondes incorrectamente, se te volverá a hacer la misma pregunta
        hasta que respondas correctamente.''')
        sounddidactica9 = playsound(ruta+'/Didactica9.mp3')
        print('''Instrucciones para los ejercicios: Primero te vamos a dictar los conjuntos
        A y B con sus respectivos elementos, deberás revisar qué elementos del conjunto A
        están en el conjunto B, porque esa será la respuesta, después de eso escucharás el
        siguiente sonido para indicarte que ya puedes responder:''')
        soundpideresp = playsound(ruta+'/PideResp.mp3')
        print('''Ding.''')
        soundsirespuestacorrecta = playsound(ruta+'/SiRespuestaCorrecta.mp3')
        print('''Si tu respuesta es correcta se escuchará:''')
        soundrespuestacorrecta = playsound(ruta+'/RespuestaCorrecta.mp3')
        print('''Ding.''')
        soundsirespuestaincorrecta = playsound(ruta+'/SiRespuestaIncorrecta.mp3')
        print('''Y si tu respuesta es incorrecta se escuchará:''')
        soundrespuestaincorrecta = playsound(ruta+'/RespuestaIncorrecta.mp3')
        print('''Wrrr.''')
        sounddidactica10 = playsound(ruta+'/Didactica10.mp3')
        print('''Los números de tu respuesta debererás ingresarlos uno a uno seguido de la
        tecla Enter, y en los casos que la respuesta es vacía, solo presiona Enter. Te sugerimos
        ayudarte con los dedos de las manos si se te dificulta memorizar. En esta actividad por
        ser de aprendizaje te daremos la respuesta, pero en las evaluaciones ya no será así.''')
        sounddidactica11 = playsound(ruta+'/Didactica11.mp3')
        print('''Si no alzanzaste a escuchar o simplemente quieres que te volvamos a repetir el
        ejercicio antes de ingresar tu respuesta, presiona la tecla de suma y después Enter.''')
        sounddidactica12 = playsound(ruta+'/Didactica12.mp3')
        print('''Si quieres entrar a explorar los elementos de los conjuntos mientras estás
        respondiendo porque se te olvidó o simplemente no escuchaste bien, presiona la tecla
        de asterisco, lo que te llevará al primer elemento del primer conjunto. Podrás moverte
        por el conjunto con las teclas a y d, y si quieres cambiar de conjunto, presiona s. Para
        salir de la exploración y continuar con tu respuesta, presiona la tecla menos.''')
      if(didac=='' or didac=='4'):
        i=0
        while i<npreg0: #Con esta instrucción se creará un bucle dependiendo que cuantas
        preguntas se requieran en la actividad.
          conjA=set() #Creamos un primer conjunto de números aleatorios.
          conjB=set() #Creamos un segundo conjunto de números aleatorios.
          LLENARCONJUNTOS(conjA,3,4)
          LLENARCONJUNTOS(conjB,3,4)
          conjC=set()
          conjC=conjA&conjB #Creamos el conjunto intersección (Respuesta).
          if len(conjC)!=0:
            i+=1
            print('  EJERCICIO DE PRÁCTICA', i)
            print('Conjunto A:', conjA)
            print('Conjunto B:', conjB)
            print('Conjunto Intersección (Respuesta):', conjC)
            incorrecto=True
```

```
            while incorrecto: #Ponemos un while para que se repita la
            ↪  pregunta hasta que sea
            contestada correctamente.
              soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')
              DICTACONJUNTO(conjA)
              soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')
              DICTACONJUNTO(conjB)
              conjD=set()
              soundconjuntoresp = playsound(ruta+'/ConjuntoResp.mp3')
              DICTACONJUNTO(conjC)
              print('Escribe tu respuesta:')
              soundpideresp = playsound(ruta+'/PideResp.mp3')
              repetir=False
              j=0
              while (j<(len(conjC))): #Recibimos el conjunto
              ↪  respuesta del teclado.
                j+=1
                resp=input()
                if (resp=='+'):
                  repetir=True
                  break
                if (resp==''):
                  break
                if (resp=='*'):
                  listA=list(conjA)
                  listB=list(conjB)
                  EXPLORACONJUNTOS (listA, listB, 1)
                  print('Continua tu respuesta:')
                  soundpideresp = playsound(ruta+'/PideResp.mp3')
                  j-=1
                else:
                  conjD.add(resp)
              if conjC==conjD: #Revisamos si el usuario respondió
              ↪  adecuadamente.
                incorrecto=False
                print("      ¡RESPUESTA CORRECTA!\n")
                soundrespuestacorrecta =
                ↪  playsound(ruta+'/RespuestaCorrecta.mp3')
              elif repetir==True:
                soundrepetir = playsound(ruta+'/Repetir.mp3')
                print('''Repitamos el ejercicio.''')
              else:
                print("      ¡RESPUESTA INCORRECTA!\n")
                soundrespuestaincorrecta =
                ↪  playsound(ruta+'/RespuestaIncorrecta.mp3')
          if (menudidac=='a'):
            menudidac='d'
        soundfelicitacionesdidac =
        ↪  playsound(ruta+'/FelicitacionesDidac.mp3')
        print('''Felicitaciones. Haz terminado la actividad para el
        ↪  aprendizaje de la intersección,
        y tú puedes decidir si estás listo para pasar a las actividades
↪       de evaluación o vuelves
        a repetir la actividad de aprendizaje.''')


      elif act=='1':
        Fila_Datos=[]
        Filas_Datos=[]
        timeFA=0
        timeSEA=0
        timeEA=0
        timeCEA=0
        nRA=0
        nEA=0
        nREA=0
        errA=0
        soundevaluacion1 = playsound(ruta+'/Evaluacion1.mp3')
        print('''Haz ingresado a la 'Evaluación 1'. En esta parte se
        ↪  evaluará la intersección
        de dos conjuntos en su nivel más básico.''')
        i=0
        while i<npreg1: #Con esta instrucción se creará un bucle
        ↪  dependiendo que cuantas preguntas
        se requieran en la actividad.
          conjA=set() #Creamos un primer conjunto de números aleatorios.
          conjB=set() #Creamos un segundo conjunto de números
          ↪  aleatorios.
          LLENARCONJUNTOS(conjA,3,5)
          LLENARCONJUNTOS(conjB,4,5)
          conjC=set()
          conjC=conjA&conjB #Creamos el conjunto intersección
          ↪  (Respuesta).
          if len(conjC)!=0:
            i+=1
            print('  PREGUNTA', i)
            print('Conjunto A:', conjA)
            print('Conjunto B:', conjB)
            print('Conjunto Intersección (Respuesta):', conjC)
            incorrecto=True
            repetir=False
            time1SE=time.time()
            timeE=0
            nR=0
            nE=0
            err=0
            while incorrecto: #Ponemos un while para que se repita la
            ↪  pregunta hasta que sea
            contestada correctamente.
              time1R=time.time()
              soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')
              DICTACONJUNTO(conjA)
              soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')
              DICTACONJUNTO(conjB)
              conjD=set()
              print('Escribe tu respuesta:')
              soundpideresp = playsound(ruta+'/PideResp.mp3')
              repetir=False
              time1F=time.time()
              time2R=time.time()
              time1SE+=time2R-time1R
              j=0
              while (j<(len(conjC))): #Recibimos el conjunto respuesta
              ↪  del teclado.
                j+=1
                resp=input()
                if (resp=='+'):
                  repetir=True
                  break
                if (resp=='*'):
                  listA=list(conjA)
                  listB=list(conjB)
                  time1E=time.time()
                  EXPLORACONJUNTOS (listA, listB, 1)
                  time2E=time.time()
                  time1F+=time2E-time1E
                  time1SE+=time2E-time1E
                  timeE+=time2E-time1E
                  nE+=1
                  print('Continua tu respuesta:')
                  soundpideresp = playsound(ruta+'/PideResp.mp3')
                  j-=1
                else:
                  conjD.add(resp)
              if conjC==conjD: #Revisamos si el usuario respondió
              ↪  adecuadamente.
                timeFF=time.time()
                incorrecto=False
                print("      ¡RESPUESTA CORRECTA!\n")
                soundrespuestacorrecta =
                ↪  playsound(ruta+'/RespuestaCorrecta.mp3')
              elif repetir==True:
                soundrepetir = playsound(ruta+'/Repetir.mp3')
                print('''Repitamos el ejercicio.''')
                nR+=1
              else:
                print("      ¡RESPUESTA INCORRECTA!\n")
                soundrespuestaincorrecta =
                ↪  playsound(ruta+'/RespuestaIncorrecta.mp3')
                err+=1
              timeFA+=timeFF-time1F
              timeSEA+=timeFF-time1SE
              timeEA+=timeE
              timeCEA+=(timeFF-time1SE)+timeE
              nRA+=nR
              nEA+=nE
              nREA+=nR+nE
              errA+=err
              Fila_Datos=[conjC, timeFF-time1F, timeFA, timeFF-time1SE,
              ↪  timeSEA, timeE, timeEA,
              (timeFF-time1SE)+timeE, timeCEA, nR, nRA, nE, nEA, nR+nE,
              ↪  nREA, err, errA]
              Filas_Datos.append(Fila_Datos)
        soundretroa1 = playsound(ruta+'/Retroa1.mp3')
        print('''En esta actividad se realizaron''')
        DICTANUMERO (str(npreg1))
        soundretroa2 = playsound(ruta+'/Retroa2.mp3')
        print('''preguntas, y tus errores fueron ''')
        DICTANUMERO (str(errA))
        if (errA<(npreg1/2)):
          soundfelicitaciones1 = playsound(ruta+'/Felicitaciones1.mp3')
          print('''Felicitaciones. Haz concluido la 'Evaluación 1'.''')
          soundsiguienteactividad =
          ↪  playsound(ruta+'/SiguienteActividad.mp3')
          print('''Puedes pasar a la siguiente actividad.''')
        else:
          soundintentanuevamente =
          ↪  playsound(ruta+'/IntentaNuevamente.mp3')
          print('''Por la cantidad de respuestas incorrectas que tuviste
          ↪  en esta actividad te
          sugerimos volver a realizarla antes de pasar a la
↪         siguiente.''')
        df_dat1=pd.DataFrame(Filas_Datos, columns=['Conjunto respuesta',
        ↪  'Tiempo final en contestar
        correctamente', 'Tiempo final en contestar correctamente
        ↪  acumulado', 'Tiempo sin exploración
        por pregunta', 'Tiempo sin exploración acumulado', 'Tiempo de
        ↪  exploración por pregunta',
        'Tiempo de exploración acumulado', 'Tiempo con exploración por
        ↪  pregunta', 'Tiempo con
        exploración acumulado', 'Número de repeticiones por pregunta',
        ↪  'Número de repeticiones
```

```
    acumulado', 'Número de exploraciones por pregunta', 'Número de
    ↪ exploraciones acumulado',
    'Número de repeticiones más exploraciones por pregunta', 'Número
    ↪ de repeticiones más
    exploraciones acumulado', 'Errores por pregunta', 'Errores
    ↪ acumulados'])
    df_dat1
    df_dat1.to_csv(ruta+'/CI1/dat1_I_.csv')


  elif act=='2':
    Fila_Datos=[]
    Filas_Datos=[]
    timeFA=0
    timeSEA=0
    timeEA=0
    timeCEA=0
    nRA=0
    nEA=0
    nREA=0
    errA=0
    soundevaluacion2 = playsound(ruta+'/Evaluacion2.mp3')
    print('''Haz ingresado a la 'Evaluación 2'. En esta parte se
    ↪ evaluará la intersección
    de dos conjuntos con la posibilidad de que haya conjuntos
↪   vacíos.''')
    i=0
    while i<npreg2: #Con esta instrucción se creará un bucle
    ↪ dependiendo que cuantas preguntas
    se requieran en la actividad.
      conjA=set() #Creamos un primer conjunto de números aleatorios.
      conjB=set() #Creamos un segundo conjunto de números
      ↪ aleatorios.
      LLENARCONJUNTOS(conjA,4,6)
      LLENARCONJUNTOS(conjB,5,6)
      conjC=set()
      conjC=conjA&conjB #Creamos el conjunto intersección
      ↪ (Respuesta).
      i+=1
      print('   PREGUNTA', i)
      print('Conjunto A:', conjA)
      print('Conjunto B:', conjB)
      print('Conjunto Intersección (Respuesta):', conjC)
      incorrecto=True
      repetir=False
      time1SE=time.time()
      timeE=0
      nR=0
      nE=0
      err=0
      while incorrecto: #Ponemos un while para que se repita la
      ↪ pregunta hasta que sea contestada
      correctamente.
        time1R=time.time()
        soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')
        DICTACONJUNTO(conjA)
        soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')
        DICTACONJUNTO(conjB)
        conjD=set()
        print('Escribe tu respuesta:')
        soundpideresp = playsound(ruta+'/PideResp.mp3')
        repetir=False
        time1F=time.time()
        time2R=time.time()
        time1SE+=time2R-time1R
        if (len(conjC)==0):
          resp=input()
          if (resp=='+'):
            repetir=True
          elif (resp==''):
            conjD=set()
          elif (resp=='*'):
            listA=list(conjA)
            listB=list(conjB)
            time1E=time.time()
            EXPLORACONJUNTOS (listA, listB, 1)
            time2E=time.time()
            time1F+=time2E-time1E
            time1SE+=time2E-time1E
            timeE+=time2E-time1E
            nE+=1
            print('Continua tu respuesta:')
            soundpideresp = playsound(ruta+'/PideResp.mp3')
            resp=input()
            if (resp==''):
              conjD=set()
            else:
              conjD.add(resp)
          else:
            conjD.add(resp)
        else:
          j=0
          while (j<(len(conjC))): #Recibimos el conjunto respuesta
          ↪ del teclado.
            j+=1
            resp=input()
            if (resp=='+'):
              repetir=True
              break
            if (resp=='*'):
              listA=list(conjA)
              listB=list(conjB)
              time1E=time.time()
              EXPLORACONJUNTOS (listA, listB, 1)
              time2E=time.time()
              time1F+=time2E-time1E
              time1SE+=time2E-time1E
              timeE+=time2E-time1E
              nE+=1
              print('Continua tu respuesta:')
              soundpideresp = playsound(ruta+'/PideResp.mp3')
              j-=1
            else:
              conjD.add(resp)
        if (conjC==conjD and repetir==False): #Revisamos si el
        ↪ usuario respondió adecuadamente.
          timeFF=time.time()
          incorrecto=False
          print("      ¡RESPUESTA CORRECTA!\n")
          soundrespuestacorrecta =
          ↪ playsound(ruta+'/RespuestaCorrecta.mp3')
        elif repetir==True:
          soundrepetir = playsound(ruta+'/Repetir.mp3')
          print('''Repitamos el ejercicio.''')
          nR+=1
        else:
          print("      ¡RESPUESTA INCORRECTA!\n")
          soundrespuestaincorrecta =
          ↪ playsound(ruta+'/RespuestaIncorrecta.mp3')
          err+=1
      timeFA+=timeFF-time1F
      timeSEA+=timeFF-time1SE
      timeEA+=timeE
      timeCEA+=(timeFF-time1SE)+timeE
      nRA+=nR
      nEA+=nE
      nREA+=nR+nE
      errA+=err
      Fila_Datos=[conjC, timeFF-time1F, timeFA, timeFF-time1SE,
      ↪ timeSEA, timeE, timeEA,
      (timeFF-time1SE)+timeE, timeCEA, nR, nRA, nE, nEA, nR+nE, nREA,
      ↪ err, errA]
      Filas_Datos.append(Fila_Datos)
    soundretroa1 = playsound(ruta+'/Retroa1.mp3')
    print('''En esta actividad se realizaron''')
    DICTANUMERO (str(npreg2))
    soundretroa2 = playsound(ruta+'/Retroa2.mp3')
    print('''preguntas, y tus errores fueron ''')
    DICTANUMERO (str(errA))
    if (errA<(npreg2/2)):
      soundfelicitaciones2 = playsound(ruta+'/Felicitaciones2.mp3')
      print('''Felicitaciones. Haz concluido la 'Evaluación 2'.''')
      soundsiguienteactividad =
      ↪ playsound(ruta+'/SiguienteActividad.mp3')
      print('''Puedes pasar a la siguiente actividad.''')
    else:
      soundintentanuevamente =
      ↪ playsound(ruta+'/IntentaNuevamente.mp3')
      print('''Por la cantidad de respuestas incorrectas que tuviste
      ↪ en esta actividad te
      sugerimos volver a realizarla antes de pasar a la
↪   siguiente.''')
    df_dat2=pd.DataFrame(Filas_Datos, columns=['Conjunto respuesta',
    ↪ 'Tiempo final en contestar
    correctamente', 'Tiempo final en contestar correctamente
    ↪ acumulado', 'Tiempo sin exploración
    por pregunta', 'Tiempo sin exploración acumulado', 'Tiempo de
    ↪ exploración por pregunta',
    'Tiempo de exploración acumulado', 'Tiempo con exploración por
    ↪ pregunta', 'Tiempo con
    exploración acumulado', 'Número de repeticiones por pregunta',
    ↪ 'Número de repeticiones
    acumulado', 'Número de exploraciones por pregunta', 'Número de
    ↪ exploraciones acumulado',
    'Número de repeticiones más exploraciones por pregunta', 'Número
    ↪ de repeticiones más
    exploraciones acumulado', 'Errores por pregunta', 'Errores
    ↪ acumulados'])
    df_dat2
    df_dat2.to_csv(ruta+'/CI2/dat2_I_.csv')


  elif act=='3':
    Fila_Datos=[]
    Filas_Datos=[]
    timeFA=0
    timeSEA=0
    timeEA=0
    timeCEA=0
    nRA=0
    nEA=0
    nREA=0
```

```
    errA=0
    soundevaluacion3 = playsound(ruta+'/Evaluacion3.mp3')
    print('''Haz ingresado a la 'Evaluación 3'. En esta parte se
    ↪  evaluará la intersección
    del Conjunto A con el Conjunto B, y posteriormente la
↪   intersección del Conjunto B con
    el Conjunto A, existiendo la posibilidad de que haya conjuntos
↪   vacíos. Nótese que la
    operación intersección es conmutativa, o sea, que da el mismo
↪   resultado independientemente
    del orden de los conjuntos al hacer la operación
↪   intersección.''')
    i=0
    while i<npreg3: #Con esta instrucción se creará un bucle
    ↪  dependiendo que cuantas preguntas
    se requieran en la actividad.
      conjA=set() #Creamos un primer conjunto de números aleatorios.
      conjB=set() #Creamos un segundo conjunto de números
      ↪  aleatorios.
      LLENARCONJUNTOS(conjA,4,7)
      LLENARCONJUNTOS(conjB,4,7)
      conjC=set()
      conjC=conjA&conjB #Creamos el conjunto intersección
      ↪  (Respuesta).
      i+=1
      print('   PREGUNTA', i, 'Parte 1:')
      print('Conjunto A:', conjA)
      print('Conjunto B:', conjB)
      print('Conjunto Intersección (Respuesta):', conjC)
      soundact3parte1 = playsound(ruta+'/Act3Parte1.mp3')
      print('''Parte 1: Intersección del Conjunto A con el Conjunto
      ↪  B.''')
      incorrecto=True
      repetir=False
      time1SE=time.time()
      timeE=0
      nR=0
      nE=0
      err=0
      while incorrecto: #Ponemos un while para que se repita la
      ↪  pregunta hasta que sea contestada
      correctamente.
        time1R=time.time()
        soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')
        DICTACONJUNTO(conjA)
        soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')
        DICTACONJUNTO(conjB)
        conjD=set()
        print('Escribe tu respuesta:')
        soundprideresp = playsound(ruta+'/PideResp.mp3')
        repetir=False
        time1F=time.time()
        time2R=time.time()
        time1SE+=time2R-time1R
        if (len(conjC)==0):
          resp=input()
          if (resp=='+'):
            repetir=True
          elif (resp==''):
            conjD=set()
          elif (resp=='*'):
            listA=list(conjA)
            listB=list(conjB)
            time1E=time.time()
            EXPLORACONJUNTOS (listA, listB, 1)
            time2E=time.time()
            time1F+=time2E-time1E
            time1SE+=time2E-time1E
            timeE+=time2E-time1E
            nE+=1
            print('Continua tu respuesta:')
            soundprideresp = playsound(ruta+'/PideResp.mp3')
            resp=input()
            if (resp==''):
              conjD=set()
            else:
              conjD.add(resp)
          else:
            conjD.add(resp)
        else:
          j=0
          while (j<(len(conjC))): #Recibimos el conjunto respuesta
          ↪  del teclado.
            j+=1
            resp=input()
            if (resp=='+'):
              repetir=True
              break
            if (resp=='*'):
              listA=list(conjA)
              listB=list(conjB)
              time1E=time.time()
              EXPLORACONJUNTOS (listA, listB, 1)
              time2E=time.time()
              time1F+=time2E-time1E
              time1SE+=time2E-time1E
              timeE+=time2E-time1E
              nE+=1
              print('Continua tu respuesta:')
              soundprideresp = playsound(ruta+'/PideResp.mp3')
              j-=1
            else:
              conjD.add(resp)
        if (conjC==conjD and repetir==False): #Revisamos si el
        ↪  usuario respondió adecuadamente.
          timeFF=time.time()
          incorrecto=False
          print("      ¡RESPUESTA CORRECTA!\n")
          soundrespuestacorrecta =
          ↪  playsound(ruta+'/RespuestaCorrecta.mp3')
        elif repetir==True:
          soundrepetir = playsound(ruta+'/Repetir.mp3')
          print('''Repitamos el ejercicio.''')
          nR+=1
        else:
          print("      ¡RESPUESTA INCORRECTA!\n")
          soundrespuestaincorrecta =
          ↪  playsound(ruta+'/RespuestaIncorrecta.mp3')
          err+=1
    print('Parte 2:')
    soundact3parte2 = playsound(ruta+'/Act3Parte2.mp3')
    print('''Parte 2: Intersección del Conjunto B con el Conjunto
    ↪  A. ''')
    incorrecto=True
    repetir=False
    time1SE2=time.time()
    timeE2=0
    nR2=0
    nE2=0
    err2=0
    while incorrecto: #Ponemos un while para que se repita la
    ↪  pregunta hasta que sea contestada
    correctamente.
      time1R2=time.time()
      soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')
      DICTACONJUNTO(conjB)
      soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')
      DICTACONJUNTO(conjA)
      conjD=set()
      print('Escribe tu respuesta:')
      soundprideresp = playsound(ruta+'/PideResp.mp3')
      repetir=False
      time1F2=time.time()
      time2R2=time.time()
      time1SE2+=time2R2-time1R2
      if (len(conjC)==0):
        resp=input()
        if (resp=='+'):
          repetir=True
        elif (resp==''):
          conjD=set()
        elif (resp=='*'):
          listA=list(conjA)
          listB=list(conjB)
          time1E2=time.time()
          EXPLORACONJUNTOS (listB, listA, 2)
          time2E2=time.time()
          time1F2+=time2E2-time1E2
          time1SE2+=time2E2-time1E2
          timeE2+=time2E2-time1E2
          nE2+=1
          print('Continua tu respuesta:')
          soundprideresp = playsound(ruta+'/PideResp.mp3')
          resp=input()
          if (resp==''):
            conjD=set()
          else:
            conjD.add(resp)
        else:
          conjD.add(resp)
      else:
        j=0
        while (j<(len(conjC))): #Recibimos el conjunto respuesta
        ↪  del teclado.
          j+=1
          resp=input()
          if (resp=='+'):
            repetir=True
            break
          if (resp=='*'):
            listA=list(conjA)
            listB=list(conjB)
            time1E2=time.time()
            EXPLORACONJUNTOS (listB, listA, 2)
            time2E2=time.time()
            time1F2+=time2E2-time1E2
            time1SE2+=time2E2-time1E2
            timeE2+=time2E2-time1E2
            nE2+=1
            print('Continua tu respuesta:')
            soundprideresp = playsound(ruta+'/PideResp.mp3')
            j-=1
```

```
            else:
              conjD.add(resp)
        if (conjC==conjD and repetir==False): #Revisamos si el
        ↪  usuario respondió adecuadamente.
          timeFF2=time.time()
          incorrecto=False
          print("      ¡RESPUESTA CORRECTA!\n")
          soundrespuestacorrecta =
          ↪  playsound(ruta+'/RespuestaCorrecta.mp3')
        elif repetir==True:
          soundrepetir = playsound(ruta+'/Repetir.mp3')
          print('''Repitamos el ejercicio.''')
          nR2+=1
        else:
          print("      ¡RESPUESTA INCORRECTA!\n")
          soundrespuestaincorrecta =
          ↪  playsound(ruta+'/RespuestaIncorrecta.mp3')
          err2+=1
      timeFA+=(timeFF-time1F)+(timeFF2-time1F2)
      timeSEA+=(timeFF-time1SE)+(timeFF2-time1SE2)
      timeEA+=timeE+timeE2
      timeCEA+=((timeFF-time1SE)+timeE)+((timeFF2-time1SE2)+timeE2)
      nRA+=nR+nR2
      nEA+=nE+nE2
      nREA+=(nR+nE)+(nR2+nE2)
      errA+=err+err2
      Fila_Datos=[conjC, (timeFF-time1F)+(timeFF2-time1F2), timeFA,
      (timeFF-time1SE)+(timeFF2-time1SE2), timeSEA, timeE+timeE2,
      ↪  timeEA,
      ((timeFF-time1SE)+timeE)+((timeFF2-time1SE2)+timeE2), timeCEA,
      ↪  nR+nR2, nRA, nE+nE2, nEA,
      (nR+nE)+(nR2+nE2), nREA, err+err2, errA]
      Filas_Datos.append(Fila_Datos)
    soundretroa1 = playsound(ruta+'/Retroa1.mp3')
    print('''En esta actividad se realizaron''')
    DICTANUMERO (str(npreg3))
    soundretroa2 = playsound(ruta+'/Retroa2.mp3')
    print('''preguntas, y tus errores fueron ''')
    DICTANUMERO (str(errA))
    if (errA<(npreg3/2)):
      soundrfelicitaciones3 = playsound(ruta+'/Felicitaciones3.mp3')
      print('''Felicitaciones. Haz concluido la 'Evaluación 3'.''')
      soundsiguienteactividad =
      ↪  playsound(ruta+'/SiguienteActividad.mp3')
      print('''Puedes pasar a la siguiente actividad.''')
    else:
      soundintentanuevamente =
      ↪  playsound(ruta+'/IntentaNuevamente.mp3')
      print('''Por la cantidad de respuestas incorrectas que tuviste
      ↪  en esta actividad te
      sugerimos volver a realizarla antes de pasar a la
↪   siguiente.''')
    df_dat3=pd.DataFrame(Filas_Datos, columns=['Conjunto respuesta',
    ↪  'Tiempo final en contestar
    correctamente', 'Tiempo final en contestar correctamente
    ↪  acumulado', 'Tiempo sin exploración
    por pregunta', 'Tiempo sin exploración acumulado', 'Tiempo de
    ↪  exploración por pregunta',
    'Tiempo de exploración acumulado', 'Tiempo con exploración por
    ↪  pregunta', 'Tiempo con
    exploración acumulado', 'Número de repeticiones por pregunta',
    ↪  'Número de repeticiones
    acumulado', 'Número de exploraciones por pregunta', 'Número de
    ↪  exploraciones acumulado',
    'Número de repeticiones más exploraciones por pregunta', 'Número
    ↪  de repeticiones más
    exploraciones acumulado', 'Errores por pregunta', 'Errores
    ↪  acumulados'])
    df_dat3
    df_dat3.to_csv(ruta+'/CI3/dat3_I_.csv')


  elif act=='4':
    Fila_Datos=[]
    Filas_Datos=[]
    timeFA=0
    timeSEA=0
    timeEA=0
    timeCEA=0
    nRA=0
    nEA=0
    nREA=0
    errA=0
    soundevaluacion4 = playsound(ruta+'/Evaluacion4.mp3')
    print('''Haz ingresado a la 'Evaluación 4'. En esta parte se
    ↪  evaluará la intersección
    de tres conjuntos con la posibilidad de que haya conjuntos
↪   vacíos.''')
    i=0
    while i<npreg4: #Con esta instrucción se creará un bucle
    ↪  dependiendo que cuantas preguntas
    se requieran en la actividad.
      conjA=set() #Creamos un primer conjunto de números aleatorios.
      conjB=set() #Creamos un segundo conjunto de números
      ↪  aleatorios.
      conjC=set() #Creamos un tercer conjunto de números aleatorios.
      LLENARCONJUNTOS(conjA,5,6)
      LLENARCONJUNTOS(conjB,5,6)
      LLENARCONJUNTOS(conjC,5,6)
      conjD=set()
      conjD=conjA&conjB&conjC #Creamos el conjunto intersección
      ↪  (Respuesta).
      i+=1
      print('   PREGUNTA', i)
      print('Conjunto A:', conjA)
      print('Conjunto B:', conjB)
      print('Conjunto C:', conjC)
      print('Conjunto Intersección (Respuesta):', conjD)
      incorrecto=True
      repetir=False
      time1SE=time.time()
      timeE=0
      nR=0
      nE=0
      err=0
      while incorrecto: #Ponemos un while para que se repita la
      ↪  pregunta hasta que sea contestada
      correctamente.
        time1R=time.time()
        soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')
        DICTACONJUNTO(conjA)
        soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')
        DICTACONJUNTO(conjB)
        soundconjuntoC = playsound(ruta+'/ConjuntoC.mp3')
        DICTACONJUNTO(conjC)
        conjE=set()
        print('Escribe tu respuesta:')
        time1=time.time()
        if (repetir==False):
          time1r=time.time()
        soundpideresp = playsound(ruta+'/PideResp.mp3')
        repetir=False
        time1F=time.time()
        time2R=time.time()
        time1SE+=time2R-time1R
        if (len(conjD)==0):
          resp=input()
          if (resp=='+'):
            repetir=True
          elif (resp==''):
            conjE=set()
          elif (resp=='*'):
            listA=list(conjA)
            listB=list(conjB)
            listC=list(conjC)
            time1E=time.time()
            EXPLORACONJUNTOS2 (listA, listB, listC, 1)
            time2E=time.time()
            time1F+=time2E-time1E
            time1SE+=time2E-time1E
            timeE+=time2E-time1E
            nE+=1
            print('Continua tu respuesta:')
            soundpideresp = playsound(ruta+'/PideResp.mp3')
            resp=input()
            if (resp==''):
              conjE=set()
            else:
              conjE.add(resp)
          else:
            conjE.add(resp)
        else:
          j=0
          while (j<(len(conjD))): #Recibimos el conjunto respuesta
          ↪  del teclado.
            j+=1
            resp=input()
            if (resp=='+'):
              repetir=True
              break
            if (resp=='*'):
              listA=list(conjA)
              listB=list(conjB)
              listC=list(conjC)
              time1E=time.time()
              EXPLORACONJUNTOS2 (listA, listB, listC, 1)
              time2E=time.time()
              time1F+=time2E-time1E
              time1SE+=time2E-time1E
              timeE+=time2E-time1E
              nE+=1
              print('Continua tu respuesta:')
              soundpideresp = playsound(ruta+'/PideResp.mp3')
              j-=1
            else:
              conjE.add(resp)
        if (conjD==conjE and repetir==False): #Revisamos si el
        ↪  usuario respondió adecuadamente.
          timeFF=time.time()
          incorrecto=False
          print("      ¡RESPUESTA CORRECTA!\n")
```

```
          soundrespuestacorrecta =
          ↪  playsound(ruta+'/RespuestaCorrecta.mp3')
        elif repetir==True:
          soundrepetir = playsound(ruta+'/Repetir.mp3')
          print('''Repitamos el ejercicio.''')
          nR+=1
        else:
          print("      ¡RESPUESTA INCORRECTA!\n")
          soundrespuestaincorrecta =
          ↪  playsound(ruta+'/RespuestaIncorrecta.mp3')
          err+=1
      timeFA+=timeFF-time1F
      timeSEA+=timeFF-time1SE
      timeEA+=timeE
      timeCEA+=(timeFF-time1SE)+timeE
      nRA+=nR
      nEA+=nE
      nREA+=nR+nE
      errA+=err
      Fila_Datos=[conjE, timeFF-time1F, timeFA, timeFF-time1SE,
      ↪  timeSEA, timeE, timeEA,
      (timeFF-time1SE)+timeE, timeCEA, nR, nRA, nE, nEA, nR+nE, nREA,
      ↪  err, errA]
      Filas_Datos.append(Fila_Datos)
    soundretroa1 = playsound(ruta+'/Retroa1.mp3')
    print('''En esta actividad se realizaron''')
    DICTANUMERO (str(npreg4))
    soundretroa2 = playsound(ruta+'/Retroa2.mp3')
    print('''preguntas, y tus errores fueron ''')
    DICTANUMERO (str(errA))
    if (errA<(npreg4/2)):
      soundrfelicitaciones4 = playsound(ruta+'/Felicitaciones4.mp3')
      print('''Felicitaciones. Haz concluido la 'Evaluación 4'.''')
      soundsiguienteactividad =
      ↪  playsound(ruta+'/SiguienteActividad.mp3')
      print('''Puedes pasar a la siguiente actividad.''')
    else:
      soundintentanuevamente =
      ↪  playsound(ruta+'/IntentaNuevamente.mp3')
      print('''Por la cantidad de respuestas incorrectas que tuviste
      ↪  en esta actividad te
      sugerimos volver a realizarla antes de pasar a la
↪  siguiente.''')
    df_dat4=pd.DataFrame(Filas_Datos, columns=['Conjunto respuesta',
    ↪  'Tiempo final en contestar
    correctamente', 'Tiempo final en contestar correctamente
    ↪  acumulado', 'Tiempo sin exploración
    por pregunta', 'Tiempo sin exploración acumulado', 'Tiempo de
    ↪  exploración por pregunta',
    'Tiempo de exploración acumulado', 'Tiempo con exploración por
    ↪  pregunta', 'Tiempo con
    exploración acumulado', 'Número de repeticiones por pregunta',
    ↪  'Número de repeticiones
    acumulado', 'Número de exploraciones por pregunta', 'Número de
    ↪  exploraciones acumulado',
    'Número de repeticiones más exploraciones por pregunta', 'Número
    ↪  de repeticiones más
    exploraciones acumulado', 'Errores por pregunta', 'Errores
    ↪  acumulados'])
    df_dat4
    df_dat4.to_csv(ruta+'/CI4/dat4_I_.csv')


  elif act=='5':
    i=0
    while i<npreg0: #Con esta instrucción se creará un bucle
    ↪  dependiendo que cuantas preguntas
    se requieran en la actividad.
      conjA=set() #Creamos un primer conjunto de números aleatorios.
      conjB=set() #Creamos un segundo conjunto de números
      ↪  aleatorios.
      LLENARCONJUNTOS(conjA,3,4)
      LLENARCONJUNTOS(conjB,3,4)
      conjC=set()
      conjC=conjA&conjB #Creamos el conjunto intersección
      ↪  (Respuesta).
      if len(conjC)!=0:
        i+=1
        print('  EJERCICIO DE PRÁCTICA', i)
        print('Conjunto A:', conjA)
        print('Conjunto B:', conjB)
        print('Conjunto Intersección (Respuesta):', conjC)
        incorrecto=True
        while incorrecto: #Ponemos un while para que se repita la
        ↪  pregunta hasta que sea
        contestada correctamente.
          soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')
          DICTACONJUNTO(conjA)
          soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')
          DICTACONJUNTO(conjB)
          conjD=set()
          soundconjuntoresp = playsound(ruta+'/ConjuntoResp.mp3')
          print('''La respuesta es:''')
          DICTACONJUNTO(conjC)
          print('Escribe tu respuesta:')
          soundpideresp = playsound(ruta+'/PideResp.mp3')
          repetir=False
          j=0
          while (j<(len(conjC))): #Recibimos el conjunto respuesta
          ↪  del teclado.
            j+=1
            resp=input()
            if (resp=='+'):
              repetir=True
              break
            if (resp==''):
              break
            if (resp=='*'):
              listA=list(conjA)
              listB=list(conjB)
              EXPLORACONJUNTOS (listA, listB, 1)
              print('Continua tu respuesta:')
              soundpideresp = playsound(ruta+'/PideResp.mp3')
              j-=1
            else:
              conjD.add(resp)
          if conjC==conjD: #Revisamos si el usuario respondió
          ↪  adecuadamente.
            incorrecto=False
            print("      ¡RESPUESTA CORRECTA!\n")
            soundrespuestacorrecta =
            ↪  playsound(ruta+'/RespuestaCorrecta.mp3')
          elif repetir==True:
            soundrepetir = playsound(ruta+'/Repetir.mp3')
            print('''Repitamos el ejercicio.''')
          else:
            print("      ¡RESPUESTA INCORRECTA!\n")
            soundrespuestaincorrecta =
            ↪  playsound(ruta+'/RespuestaIncorrecta.mp3')
        soundmas = playsound(ruta+'/Mas.mp3')
        print('''Si deseas otro ejercicio de práctica presiona la
        ↪  tecla de '+' seguido de un
        enter, sino solo enter.''')
        mas=input('¿Practicar de nuevo?:')
        if (mas=='+'):
          i-=1

  elif act=='6':
    print("Estás saliendo.")

  else:
    print("Opción erronea")
    soundrespuestaincorrecta =
    ↪  playsound(ruta+'/RespuestaIncorrecta.mp3')

soundsalida = playsound(ruta+'/Salida.mp3')
print('''Estás saliendo del programa para la enseñanza y evaluación
↪  de la intersección. Muchas
gracias por participar.''')
```

A.3. Diferencia

```
#Este código es una actividad didáctica para la enseñanza y la evaluación de la diferencia
entre dos conjuntos.
import time
from datetime import datetime
import pandas as pd
import numpy as np
from playsound import playsound
from tkinter import Tk, Label, Button
import random as rd
import os
ruta = os.getcwd()


#Funciones:
def LLENARCONJUNTOS(conj, a, b): #Función para llenar los conjuntos.
  for m in range(a):
    x=rd.randint(0,b)
    conj.add(str(x))

def DICTACONJUNTO (conj):
  for i in conj: #Mencionamos los elementos del conjunto
    if i=='0':
      sound0 = playsound(ruta+'/0.mp3')
    if i=='1':
      sound1 = playsound(ruta+'/1.mp3')
    if i=='2':
      sound2 = playsound(ruta+'/2.mp3')
    if i=='3':
      sound3 = playsound(ruta+'/3.mp3')
    if i=='4':
      sound4 = playsound(ruta+'/4.mp3')
    if i=='5':
      sound5 = playsound(ruta+'/5.mp3')
    if i=='6':
      sound6 = playsound(ruta+'/6.mp3')
    if i=='7':
      sound7 = playsound(ruta+'/7.mp3')
    if i=='8':
      sound8 = playsound(ruta+'/8.mp3')
    if i=='9':
      sound9 = playsound(ruta+'/9.mp3')

def DICTANUMERO (n):
  ConjRetro=set()
  ConjRetro.add(n)
  DICTACONJUNTO(ConjRetro)

def EXPLORACONJUNTOS (listA, listB, n):
  mov='a'
  i=0
  if (n==1):
    print('Estás en el Conjunto A')
    soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')

  elif (n==2):
    print('Estás en el Conjunto B')
    soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')

  while(mov!='-'):
    print(listA[i])
    DICTANUMERO(listA[i])
    mov=input('Movimiento:')
    if (mov=='d' or mov=='D'):
      if (i<(len(listA)-1)):
        i+=1
      else:
        i=0
    elif (mov=='a' or mov=='A'):
      if (i==0):
        i=len(listA)-1
      else:
        i-=1
    elif (mov=='s' or mov=='S'):
      if (n==1):
        EXPLORACONJUNTOS (listB, listA, 2)
        break
      else:
        EXPLORACONJUNTOS (listB, listA, 1)
        break

def EXPLORACONJUNTOS2 (listA, listB, listC, n):
  mov='a'
  i=0
  if (n==1):
    print('Estás en el Conjunto A')
    soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')

  elif (n==2):
    print('Estás en el Conjunto B')
    soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')

  elif (n==3):
    print('Estás en el Conjunto C')
    soundconjuntoC = playsound(ruta+'/ConjuntoC.mp3')

  while(mov!='-'):
    print(listA[i])
    DICTANUMERO(listA[i])
    mov=input('Movimiento:')
    if (mov=='d' or mov=='D'):
      if (i<(len(listA)-1)):
        i+=1
      else:
        i=0
    elif (mov=='a' or mov=='A'):
      if (i==0):
        i=len(listA)-1
      else:
        i-=1
    elif (mov=='s' or mov=='S'):
      if (n==1):
        EXPLORACONJUNTOS2 (listB, listC, listA, 2)
        break
      elif (n==2):
        EXPLORACONJUNTOS2 (listB, listC, listA, 3)
        break
      else:
        EXPLORACONJUNTOS2 (listB, listC, listA, 1)
        break

#Variables globales
npreg0=1
npreg1=10
npreg2=10
npreg3=10
npreg4=10


#Bienvenida al programa (Audios):
soundbienvenida = playsound(ruta+'/Bienvenida.mp3')
print('''Bienvenido al programa de actividades didácticas para la enseñanza y evaluación de
la diferencia de dos conjuntos.''')
sounddefinicionformal = playsound(ruta+'/DefinicionFormal.mp3')
print('''Definición formal: La diferencia de dos conjuntos es una operación que da como resultado
otro conjunto con los elementos del primer conjunto sin los elementos del segundo conjunto.''')
soundexplicacion = playsound(ruta+'/Explicacion.mp3')
print('''Este programa cuenta con 5 actividades diferentes, una de ellas tiene la finalidad de
enseñarte, y las otras cuatro son evaluaciones que van aumentando su dificultad e incluyendo
cada vez más conceptos.''')
soundenter = playsound(ruta+'/Enter.mp3')
print('''Para interactuar con este programa deberás teclear Enter después de cada número.''')


#Menú de actividades:
act='0'
while (act!='6'):
  soundmenu = playsound(ruta+'/Menu.mp3')
  print('''Presiona cero para la actividad de aprendizaje, uno para la evaluación en su nivel
  más básico, dos para la evaluación del nivel básico incluyendo conjuntos vacíos, tres
  para la evaluación de la conmutatividad, cuatro para la evaluación con tres conjuntos,
  cinco para realizar ejercicios de práctica, seis para salir.''')
  act=input('Opción Menú:')
  if act=='0':
    print('Actividad Didáctica')
    sounddidactica1 = playsound(ruta+'/Didactica1.mp3')
    print('''Haz ingresado a la actividad didáctica de la diferencia. En esta parte te
    enseñaremos cómo funciona, te daremos algunos ejemplos, y te guiaremos a que realices
    algunos ejercicios similares a las actividades de evaluación.''')
    soundmenudidac = playsound(ruta+'/Didactica1_1.mp3')
    print('''Si quieres estudiar la actividad didáctica de manera fluida presiona 'a' seguido
    de enter, o si prefieres poder moverte entre sus diferentes secciones presiona 'd' y enter.''')
    menudidac=input('Fluidez didáctica:')
    didac=''
    while(didac=='' or didac=='1' or didac=='2' or didac=='3' or didac=='4'):
      if (menudidac=='d'):
```

```
        print ('Avance por partes')
        soundrep1 = playsound(ruta+'/Rep1.mp3')
        print('''Presiona uno para la definición informal, dos para los ejemplos en la vida
        cotidiana, tres para las instrucciones de los ejercicios, o cuatro para los ejercicios
        prácticos. Cualquier otro número para salir.''')
        didac=input('Opción menú didáctica:')
      else:
        print ('Avance fluido')
      if(didac=='' or didac=='1'):
        print ('Definición Informal')
        sounddidactica2 = playsound(ruta+'/Didactica2.mp3')
        print('''La diferencia entre conjuntos es una operación que 'elimina' del primer
        conjunto los elementos que aparecen en el segundo conjunto.''')
      if(didac=='' or didac=='2'):
        print ('Ejemplos')
        ejemp='0'
        if (menudidac=='d'):
          soundselecejem = playsound(ruta+'/Didactica3_1.mp3')
          print('''Hay cinco ejemplos, presiona el número de ejemplo que quieres escuchar
          seguido de un enter o cero si quieres escucharlos todos.''')
          ejemp=input('Número de ejemplo:')
        if (ejemp=='0' or ejemp=='1'):
          print ('Ejemplo 1')
          sounddidactica3 = playsound(ruta+'/Didactica3.mp3')
          print('''Ejemplo 1: El conjunto A tendrá la propiedad que serán todas las personas
          a las que les gusta lo dulce, y el conjunto B tendrá la propiedad que serán todas
          las personas a las que les gusta lo salado. Como recordarás, tenemos que quitar del
          conjunto A a todos los elementos que también aparezcan en el conjunto B, es decir,
          del conjunto A que son todas las personas a las que les gusta lo dulce, vamos a quitar
          a todas las personas que también les gusta lo salado, y nuestro conjunto diferencia
          son todas las personas a las que les gusta lo dulce, pero no les gusta lo salado.''')
        if (ejemp=='0' or ejemp=='2'):
          print ('Ejemplo 2')
          sounddidactica4 = playsound(ruta+'/Didactica4.mp3')
          print('''Ejemplo 2: El conjunto A tendrá la propiedad que serán todas las personas
          a las que les gusta el rock, y el conjunto B tendrá la propiedad que serán todas
          las personas a las que les gustan los boleros. Al igual que en el ejemplo anterior,
          del conjunto A que son todas las personas a las que les gusta el rock, vamos a quitar
          a todas las personas que también les gustan los boleros, porque estos están en el
          conjunto B. Quedándonos nuestro conjunto diferencia como todas las personas que les
          gusta el rock, pero que no les gustan los boleros.''')
        if (ejemp=='0' or ejemp=='3'):
          print ('Ejemplo 3')
          sounddidactica5 = playsound(ruta+'/Didactica5.mp3')
          print('''Ejemplo 3: Ahora revisemos un caso donde la diferencia sea vacía. Es decir,
          donde todos los elementos del conjunto A también están en el conjunto B. Pongamos que
          el conjunto A tendrá la propiedad que serán todas las personas que saben Braille,
          y el conjunto B tendrá la propiedad que serán todas las personas que saben leer
          o escribir. Como podrás notar, todas las personas que saben Braille saben leer o
          escribir, por lo que, si al conjunto A que son todas las personas que saben Braille,
          le quitamos todas las personas que saben leer o escribir, la diferencia es vacía.''')
        if (ejemp=='0' or ejemp=='4'):
          print ('Ejemplo 4')
          sounddidactica6 = playsound(ruta+'/Didactica6.mp3')
          print('''Ejemplo 4: Aquí vamos a observar que la operación diferencia no es
          conmutativa, esto quiere decir que, sí importa si hacemos la operación diferencia
          del conjunto A con el conjunto B, o si la hacemos al revés, que hagamos la diferencia
          del conjunto B con el conjunto A, porque nos puede dar un resultado completamente
          diferente. Veamos: Si el conjunto A tiene la propiedad que son todas las personas
          que tienen más de 25 años, y el conjunto B tiene la propiedad que son todas las
          personas que tienen hijos, el conjunto diferencia del conjunto A con el conjunto B,
          son todas las personas que tienen más de 25 años y que no tienen hijos; pero si
          hacemos la diferencia en diferente orden, es decir, la diferencia del conjunto B con
          el conjunto A, nos daremos cuenta que no es lo mismo, ahora la respuesta son todas
          las personas que tienen hijos, pero que no tienen más de 25 años.''')
        if (ejemp=='0' or ejemp=='5'):
          print ('Ejemplo 5')
          sounddidactica7 = playsound(ruta+'/Didactica7.mp3')
          print('''Ejemplo 5: Veamos un caso en el que haremos la diferencia de tres conjuntos:
          Pongamos el conjunto A con la propiedad de que son todas las personas que trabajan, el
          conjunto B con la propiedad de que son todas las personas que estudian la universidad,
          y el conjunto C con la propiedad que son todas las personas que hacen ejercicio, y
          el resultado serán todas las personas que cumplen la primera propiedad pero no las
          otras dos, es decir, que son todas las personas que trabajan, pero que no estudian
          la universidad, y que no hacen ejercicio.''')
      if(didac=='' or didac=='3'):
        print ('Instrucciones')
        sounddidactica8 = playsound(ruta+'/Didactica8.mp3')
        print('''A continuación te guiaremos a realizar algunos ejercicios similares a las
        actividades de evaluación, dándote las respuestas, para que te vayas familiarizando. Cabe
        aclarar que para las actividades de evaluación solo utilizarás las teclas del 0 al
        9 y el Enter. Si respondes incorrectamente, se te volverá a hacer la misma pregunta
        hasta que respondas correctamente.''')
        sounddidactica9 = playsound(ruta+'/Didactica9.mp3')
        print('''Instrucciones para los ejercicios: Primero te vamos a mencionar los conjuntos A
        y B con sus elementos, porque deberás quitar los elementos del conjunto A que aparezcan
        en el conjunto B, y los que queden serán la respuesta, después de eso escucharás el
        siguiente sonido para indicarte que ya puedes responder:''')
        soundpideresp = playsound(ruta+'/PideResp.mp3')
        soundsirespuestacorrecta = playsound(ruta+'/SiRespuestaCorrecta.mp3')
        print('''Si tu respuesta es correcta se escuchará:''')
        soundrespuestacorrecta = playsound(ruta+'/RespuestaCorrecta.mp3')
        soundsirespuestaincorrecta = playsound(ruta+'/SiRespuestaIncorrecta.mp3')
        print('''Y si tu respuesta es incorrecta se escuchará:''')
        soundrespuestaincorrecta = playsound(ruta+'/RespuestaIncorrecta.mp3')
        sounddidactica10 = playsound(ruta+'/Didactica10.mp3')
        print('''Los números de tu respuesta debererás ingresarlos uno a uno seguido de la
        tecla Enter, y en los casos que la respuesta es vacía, solo presiona Enter. Te sugerimos
        ayudarte con los dedos de las manos si se te dificulta memorizar. En esta actividad por
        ser de aprendizaje te daremos la respuesta, pero en las evaluaciones ya no será así.''')
        sounddidactica11 = playsound(ruta+'/Didactica11.mp3')
        print('''Si no alzanzaste a escuchar o simplemente quieres que te volvamos a repetir el
        ejercicio antes de ingresar tu respuesta, presiona la tecla de suma y después Enter.''')
        sounddidactica12 = playsound(ruta+'/Didactica12.mp3')
        print('''Si quieres entrar a explorar los elementos de los conjuntos mientras estás
        respondiendo porque se te olvidó o simplemente no escuchaste bien, presiona la tecla
        de asterisco, lo que te llevará al primer elemento del primer conjunto. Podrás moverte
        por el conjunto con las teclas a y d, y si quieres cambiar de conjunto, presiona s. Para
        salir de la exploración y continuar con tu respuesta, presiona la tecla menos.''')
      if(didac=='' or didac=='4'):
        i=0
        while i<npreg0: #Con esta instrucción se creará un bucle dependiendo que cuantas
        preguntas se requieran en la actividad.
          conjA=set() #Creamos un primer conjunto de números aleatorios.
          conjB=set() #Creamos un segundo conjunto de números aleatorios.
          LLENARCONJUNTOS(conjA,3,4)
          LLENARCONJUNTOS(conjB,3,4)
          conjC=set()
          conjC=conjA-conjB #Creamos el conjunto Diferencia (Respuesta)
          if len(conjC)!=0:
            i+=1
            print('  EJERCICIO DE PRÁCTICA', i)
            print('Conjunto A:', conjA)
```

```
            print('Conjunto B:', conjB)
            print('Conjunto Diferencia (Respuesta):', conjC)
            incorrecto=True
            while incorrecto: #Ponemos un while para que se repita la
            ↪  pregunta hasta que sea
            contestada correctamente
              soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')
              DICTACONJUNTO(conjA)
              soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')
              DICTACONJUNTO(conjB)
              conjD=set()
              soundconjuntoresp = playsound(ruta+'/ConjuntoResp.mp3')
              print('''La respuesta es:''')
              DICTACONJUNTO(conjC)
              print('Escribe tu respuesta:')
              soundpideresp = playsound(ruta+'/PideResp.mp3')
              repetir=False
              j=0
              while (j<(len(conjC))): #Recibimos el conjunto
              ↪  respuesta del teclado.
                j+=1
                resp=input()
                if (resp=='+'):
                  repetir=True
                  break
                if (resp==''):
                  break
                if (resp=='*'):
                  listA=list(conjA)
                  listB=list(conjB)
                  EXPLORACONJUNTOS (listA, listB, 1)
                  print('Continua tu respuesta:')
                  soundpideresp = playsound(ruta+'/PideResp.mp3')
                  j-=1
                else:
                  conjD.add(resp)
              if conjC==conjD: #Revisamos si el usuario respondió
              ↪  adecuadamente.
                incorrecto=False
                print("      ¡RESPUESTA CORRECTA!\n")
                soundrespuestacorrecta =
                ↪  playsound(ruta+'/RespuestaCorrecta.mp3')
              elif repetir==True:
                soundrepetir = playsound(ruta+'/Repetir.mp3')
                print('''Repitamos el ejercicio.''')
              else:
                print("      ¡RESPUESTA INCORRECTA!\n")
                soundrespuestaincorrecta =
                ↪  playsound(ruta+'/RespuestaIncorrecta.mp3')
          if (menudidac=='a'):
            menudidac='d'
        soundfelicitacionesdidac =
        ↪  playsound(ruta+'/FelicitacionesDidac.mp3')
        print('''Felicitaciones. Haz terminado la actividad para el
        ↪  aprendizaje de la diferencia,
        y tú puedes decidir si estás listo para pasar a las actividades
↪       de evaluación o vuelves
        a repetir la actividad de aprendizaje.''')


      elif act=='1':
        Fila_Datos=[]
        Filas_Datos=[]
        timeFA=0
        timeSEA=0
        timeEA=0
        timeCEA=0
        nRA=0
        nEA=0
        nREA=0
        errA=0
        soundevaluacion1 = playsound(ruta+'/Evaluacion1.mp3')
        print('''Haz ingresado a la 'Evaluación 1'. En esta parte se
        ↪  evaluará la diferencia de
        dos conjuntos en su nivel más básico.''')
        i=0
        while i<npreg1: #Con esta instrucción se creará un bucle
        ↪  dependiendo que cuantas preguntas
        se requieran en la actividad.
          conjA=set() #Creamos un primer conjunto de números aleatorios.
          conjB=set() #Creamos un segundo conjunto de números
          ↪  aleatorios.
          LLENARCONJUNTOS(conjA,4,5)
          LLENARCONJUNTOS(conjB,4,5)
          conjC=set()
          conjC=conjA-conjB #Creamos el conjunto Diferencia (Respuesta)
          if len(conjC)!=0:
            i+=1
            print('  PREGUNTA', i)
            print('Conjunto A:', conjA)
            print('Conjunto B:', conjB)
            print('Conjunto Diferencia (Respuesta):', conjC)
            incorrecto=True
            repetir=False
            time1SE=time.time()
            timeE=0
            nR=0
            nE=0
            err=0
            while incorrecto: #Ponemos un while para que se repita la
            ↪  pregunta hasta que sea
            contestada correctamente
              time1R=time.time()
              soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')
              DICTACONJUNTO(conjA)
              soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')
              DICTACONJUNTO(conjB)
              conjD=set()
              print('Escribe tu respuesta:')
              soundpideresp = playsound(ruta+'/PideResp.mp3')
              repetir=False
              time1F=time.time()
              time2R=time.time()
              time1SE+=time2R-time1R
              j=0
              while (j<(len(conjC))): #Recibimos el conjunto respuesta
              ↪  del teclado.
                j+=1
                resp=input()
                if (resp=='+'):
                  repetir=True
                  break
                if (resp=='*'):
                  listA=list(conjA)
                  listB=list(conjB)
                  time1E=time.time()
                  EXPLORACONJUNTOS (listA, listB, 1)
                  time2E=time.time()
                  time1F+=time2E-time1E
                  time1SE+=time2E-time1E
                  timeE+=time2E-time1E
                  nE+=1
                  print('Continua tu respuesta:')
                  soundpideresp = playsound(ruta+'/PideResp.mp3')
                  j-=1
                else:
                  conjD.add(resp)
              if conjC==conjD: #Revisamos si el usuario respondió
              ↪  adecuadamente.
                timeFF=time.time()
                incorrecto=False
                print("      ¡RESPUESTA CORRECTA!\n")
                soundrespuestacorrecta =
                ↪  playsound(ruta+'/RespuestaCorrecta.mp3')
              elif repetir==True:
                soundrepetir = playsound(ruta+'/Repetir.mp3')
                print('''Repitamos el ejercicio.''')
                nR+=1
              else:
                print("      ¡RESPUESTA INCORRECTA!\n")
                soundrespuestacorrecta =
                ↪  playsound(ruta+'/RespuestaIncorrecta.mp3')
                err+=1
            timeFA+=timeFF-time1F
            timeSEA+=timeFF-time1SE
            timeEA+=timeE
            timeCEA+=(timeFF-time1SE)+timeE
            nRA+=nR
            nEA+=nE
            nREA+=nR+nE
            errA+=err
            Fila_Datos=[conjC, timeFF-time1F, timeFA, timeFF-time1SE,
            ↪  timeSEA, timeE, timeEA,
            (timeFF-time1SE)+timeE, timeCEA, nR, nRA, nE, nEA, nR+nE,
            ↪  nREA, err, errA]
            Filas_Datos.append(Fila_Datos)
        soundretroa1 = playsound(ruta+'/Retroa1.mp3')
        print('''En esta actividad se realizaron''')
        DICTANUMERO (str(npreg1))
        soundretroa2 = playsound(ruta+'/Retroa2.mp3')
        print('''preguntas, y tus errores fueron''')
        DICTANUMERO (str(errA))
        if (errA<(npreg1/2)):
          soundfelicitaciones1 = playsound(ruta+'/Felicitaciones1.mp3')
          print('''Felicitaciones. Haz concluido la 'Evaluación 1'.''')
          soundsiguienteactividad =
          ↪  playsound(ruta+'/SiguienteActividad.mp3')
          print('''Puedes pasar a la siguiente actividad.''')
        else:
          soundintentanuevamente =
          ↪  playsound(ruta+'/IntentaNuevamente.mp3')
          print('''Por la cantidad de respuestas incorrectas que tuviste
          ↪  en esta actividad te
          sugerimos volver a realizarla antes de pasar a la
↪       siguiente.''')
        df_dat1=pd.DataFrame(Filas_Datos, columns=['Conjunto respuesta',
        ↪  'Tiempo final en contestar
        correctamente', 'Tiempo final en contestar correctamente
        ↪  acumulado', 'Tiempo sin exploración
        por pregunta', 'Tiempo sin exploración acumulado', 'Tiempo de
        ↪  exploración por pregunta',
```

```
    'Tiempo de exploración acumulado', 'Tiempo con exploración por
    ↪ pregunta', 'Tiempo con
    exploración acumulado', 'Número de repeticiones por pregunta',
    ↪ 'Número de repeticiones
    acumulado', 'Número de exploraciones por pregunta', 'Número de
    ↪ exploraciones acumulado',
    'Número de repeticiones más exploraciones por pregunta', 'Número
    ↪ de repeticiones más
    exploraciones acumulado', 'Errores por pregunta', 'Errores
    ↪ acumulados'])
    df_dat1
    df_dat1.to_csv(ruta+'/CD1/dat1_D_.csv')


  elif act=='2':
    Fila_Datos=[]
    Filas_Datos=[]
    timeFA=0
    timeSEA=0
    timeEA=0
    timeCEA=0
    nRA=0
    nEA=0
    nREA=0
    errA=0
    soundevaluacion2 = playsound(ruta+'/Evaluacion2.mp3')
    print('''Haz ingresado a la 'Evaluación 2'. En esta parte se
    ↪ evaluará la diferencia de
    dos conjuntos con la posibilidad de que haya conjuntos
↪   vacíos.''')
    i=0
    while i<npreg2: #Con esta instrucción se creará un bucle
    ↪ dependiendo que cuantas preguntas
    se requieran en la actividad.
      conjA=set() #Creamos un primer conjunto de números aleatorios.
      conjB=set() #Creamos un segundo conjunto de números
      ↪ aleatorios.
      LLENARCONJUNTOS(conjA,5,6)
      LLENARCONJUNTOS(conjB,4,6)
      conjC=set()
      conjC=conjA-conjB #Creamos el conjunto diferencia (Respuesta)
      i+=1
      print('   PREGUNTA', i)
      print('Conjunto A:', conjA)
      print('Conjunto B:', conjB)
      print('Conjunto Diferencia (Respuesta):', conjC)
      incorrecto=True
      repetir=False
      time1SE=time.time()
      timeE=0
      nR=0
      nE=0
      err=0
      while incorrecto: #Ponemos un while para que se repita la
      ↪ pregunta hasta que sea contestada
      correctamente
        time1R=time.time()
        soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')
        DICTACONJUNTO(conjA)
        soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')
        DICTACONJUNTO(conjB)
        conjD=set()
        print('Escribe tu respuesta:')
        soundprideresp = playsound(ruta+'/PideResp.mp3')
        repetir=False
        time1F=time.time()
        time2R=time.time()
        time1SE+=time2R-time1R
        if (len(conjC)==0):
          resp=input()
          if (resp=='+'):
            repetir=True
          elif (resp==''):
            conjD=set()
          elif (resp=='*'):
            listA=list(conjA)
            listB=list(conjB)
            time1E=time.time()
            EXPLORACONJUNTOS (listA, listB, 1)
            time2E=time.time()
            time1F+=time2E-time1E
            time1SE+=time2E-time1E
            timeE+=time2E-time1E
            nE+=1
            print('Continua tu respuesta:')
            soundprideresp = playsound(ruta+'/PideResp.mp3')
            resp=input()
            if (resp==''):
              conjD=set()
            else:
              conjD.add(resp)
          else:
            conjD.add(resp)
        else:
          j=0
          while (j<(len(conjC))): #Recibimos el conjunto respuesta
          ↪ del teclado.
            j+=1
            resp=input()
            if (resp=='+'):
              repetir=True
              break
            if (resp=='*'):
              listA=list(conjA)
              listB=list(conjB)
              time1E=time.time()
              EXPLORACONJUNTOS (listA, listB, 1)
              time2E=time.time()
              time1F+=time2E-time1E
              time1SE+=time2E-time1E
              timeE+=time2E-time1E
              nE+=1
              print('Continua tu respuesta:')
              soundprideresp = playsound(ruta+'/PideResp.mp3')
              j-=1
            else:
              conjD.add(resp)
        if (conjC==conjD and repetir==False): #Revisamos si el
        ↪ usuario respondió adecuadamente.
          timeFF=time.time()
          incorrecto=False
          print("      ¡RESPUESTA CORRECTA!\n")
          soundrespuestacorrecta =
          ↪ playsound(ruta+'/RespuestaCorrecta.mp3')
        elif repetir==True:
          soundrepetir = playsound(ruta+'/Repetir.mp3')
          print('''Repitamos el ejercicio.''')
          nR+=1
        else:
          print("      ¡RESPUESTA INCORRECTA!\n")
          soundrespuestacorrecta =
          ↪ playsound(ruta+'/RespuestaIncorrecta.mp3')
          err+=1
      timeFA+=timeFF-time1F
      timeSEA+=timeFF-time1SE
      timeEA+=timeE
      timeCEA+=(timeFF-time1SE)+timeE
      nRA+=nR
      nEA+=nE
      nREA+=nR+nE
      errA+=err
      Fila_Datos=[conjC, timeFF-time1F, timeFA, timeFF-time1SE,
      ↪ timeSEA, timeE, timeEA,
      (timeFF-time1SE)+timeE, timeCEA, nR, nRA, nE, nEA, nR+nE, nREA,
      ↪ err, errA]
      Filas_Datos.append(Fila_Datos)
    soundretroa1 = playsound(ruta+'/Retroa1.mp3')
    print('''En esta actividad se realizaron''')
    DICTANUMERO (str(npreg2))
    soundretroa2 = playsound(ruta+'/Retroa2.mp3')
    print('''preguntas, y tus errores fueron''')
    DICTANUMERO (str(errA))
    if (errA<(npreg2/2)):
      soundfelicitaciones2 = playsound(ruta+'/Felicitaciones2.mp3')
      print('''Felicitaciones. Haz concluido la 'Evaluación 2'.''')
      soundsiguienteactividad =
      ↪ playsound(ruta+'/SiguienteActividad.mp3')
      print('''Puedes pasar a la siguiente actividad.''')
    else:
      soundintentanuevamente =
      ↪ playsound(ruta+'/IntentaNuevamente.mp3')
      print('''Por la cantidad de respuestas incorrectas que tuviste
      ↪ en esta actividad te
      sugerimos volver a realizarla antes de pasar a la
↪   siguiente.''')
    df_dat2=pd.DataFrame(Filas_Datos, columns=['Conjunto respuesta',
    ↪ 'Tiempo final en contestar
    correctamente', 'Tiempo final en contestar correctamente
    ↪ acumulado', 'Tiempo sin exploración
    por pregunta', 'Tiempo sin exploración acumulado', 'Tiempo de
    ↪ exploración por pregunta',
    'Tiempo de exploración acumulado', 'Tiempo con exploración por
    ↪ pregunta', 'Tiempo con
    exploración acumulado', 'Número de repeticiones por pregunta',
    ↪ 'Número de repeticiones
    acumulado', 'Número de exploraciones por pregunta', 'Número de
    ↪ exploraciones acumulado',
    'Número de repeticiones más exploraciones por pregunta', 'Número
    ↪ de repeticiones más
    exploraciones acumulado', 'Errores por pregunta', 'Errores
    ↪ acumulados'])
    df_dat2
    df_dat2.to_csv(ruta+'/CD2/dat2_D_.csv')


  elif act=='3':
    Fila_Datos=[]
    Filas_Datos=[]
    timeFA=0
    timeSEA=0
    timeEA=0
    timeCEA=0
```

```
    nRA=0
    nEA=0
    nREA=0
    errA=0
    soundevaluacion3 = playsound(ruta+'/Evaluacion3.mp3')
    print('''Haz ingresado a la 'Evaluación 3'. En esta parte se
    ↪  evaluará la diferencia del
    Conjunto A con el Conjunto B, y posteriormente la diferencia del
↪   Conjunto B con el Conjunto A,
    existiendo la posibilidad de que haya conjuntos vacíos. Nótese
↪   que la operación diferencia
    no es conmutativa, o sea, que no da el mismo resultado y depende
↪   del orden de los conjuntos
    al hacer la operación diferencia.''')
    i=0
    while i<npreg3: #Con esta instrucción se creará un bucle
    ↪  dependiendo que cuantas preguntas
    se requieran en la actividad.
      conjA=set() #Creamos un primer conjunto de números aleatorios.
      conjB=set() #Creamos un segundo conjunto de números
      ↪  aleatorios.
      LLENARCONJUNTOS(conjA,5,7)
      LLENARCONJUNTOS(conjB,5,7)
      conjC=set()
      conjC=conjA-conjB #Creamos el conjunto diferencia (Respuesta)
      i+=1
      print('   PREGUNTA', i, 'parte 1:')
      print('Conjunto A:', conjA)
      print('Conjunto B:', conjB)
      print('Conjunto Diferencia (Respuesta):', conjC)
      soundact3parte1 = playsound(ruta+'/Act3Parte1.mp3')
      print('''Parte 1: Diferencia del Conjunto A con el Conjunto
      ↪  B.''')
      incorrecto=True
      repetir=False
      time1SE=time.time()
      timeE=0
      nR=0
      nE=0
      err=0
      while incorrecto: #Ponemos un while para que se repita la
      ↪  pregunta hasta que sea contestada
      correctamente
        time1R=time.time()
        soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')
        DICTACONJUNTO(conjA)
        soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')
        DICTACONJUNTO(conjB)
        conjD=set()
        print('Escribe tu respuesta:')
        soundprideresp = playsound(ruta+'/PideResp.mp3')
        repetir=False
        time1F=time.time()
        time2R=time.time()
        time1SE+=time2R-time1R
        if (len(conjC)==0):
          resp=input()
          if (resp=='+'):
            repetir=True
          elif (resp==''):
            conjD=set()
          elif (resp=='*'):
            listA=list(conjA)
            listB=list(conjB)
            time1E=time.time()
            EXPLORACONJUNTOS (listA, listB, 1)
            time2E=time.time()
            time1F+=time2E-time1E
            time1SE+=time2E-time1E
            timeE+=time2E-time1E
            nE+=1
            print('Continua tu respuesta:')
            soundprideresp = playsound(ruta+'/PideResp.mp3')
            resp=input()
            if (resp==''):
              conjD=set()
            else:
              conjD.add(resp)
          else:
            conjD.add(resp)
        else:
          j=0
          while (j<(len(conjC))): #Recibimos el conjunto respuesta
          ↪  del teclado.
            j+=1
            resp=input()
            if (resp=='+'):
              repetir=True
              break
            if (resp=='*'):
              listA=list(conjA)
              listB=list(conjB)
              time1E=time.time()
              EXPLORACONJUNTOS (listA, listB, 1)
              time2E=time.time()
              time1F+=time2E-time1E
              time1SE+=time2E-time1E
              timeE+=time2E-time1E
              nE+=1
              print('Continua tu respuesta:')
              soundprideresp = playsound(ruta+'/PideResp.mp3')
              j-=1
            else:
              conjD.add(resp)
        if (conjC==conjD and repetir==False): #Revisamos si el
        ↪  usuario respondió adecuadamente.
          timeFF=time.time()
          incorrecto=False
          print("      ¡RESPUESTA CORRECTA!\n")
          soundrespuestacorrecta =
          ↪  playsound(ruta+'/RespuestaCorrecta.mp3')
        elif repetir==True:
          soundrepetir = playsound(ruta+'/Repetir.mp3')
          print('''Repitamos el ejercicio.''')
          nR+=1
        else:
          print("      ¡RESPUESTA INCORRECTA!\n")
          soundrespuestacorrecta =
          ↪  playsound(ruta+'/RespuestaIncorrecta.mp3')
          err+=1
    conjC=conjB-conjA #Creamos el conjunto diferencia (Respuesta)
    print('Parte 2:\nConjunto Diferencia (Respuesta):', conjC)
    soundact3parte2 = playsound(ruta+'/Act3Parte2.mp3')
    print('''Parte 2: Diferencia del Conjunto B con el Conjunto
    ↪  A.''')
    incorrecto=True
    repetir=False
    time1SE2=time.time()
    timeE2=0
    nR2=0
    nE2=0
    err2=0
    while incorrecto: #Ponemos un while para que se repita la
    ↪  pregunta hasta que sea contestada
    correctamente.
      time1R2=time.time()
      soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')
      DICTACONJUNTO(conjB)
      soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')
      DICTACONJUNTO(conjA)
      conjD=set()
      print('Escribe tu respuesta:')
      soundprideresp = playsound(ruta+'/PideResp.mp3')
      repetir=False
      time1F2=time.time()
      time2R2=time.time()
      time1SE2+=time2R2-time1R2
      if (len(conjC)==0):
        resp=input()
        if (resp=='+'):
          repetir=True
        elif (resp==''):
          conjD=set()
        elif (resp=='*'):
          listA=list(conjA)
          listB=list(conjB)
          time1E2=time.time()
          EXPLORACONJUNTOS (listB, listA, 2)
          time2E2=time.time()
          time1F2+=time2E2-time1E2
          time1SE2+=time2E2-time1E2
          timeE2+=time2E2-time1E2
          nE2+=1
          print('Continua tu respuesta:')
          soundprideresp = playsound(ruta+'/PideResp.mp3')
          resp=input()
          if (resp==''):
            conjD=set()
          else:
            conjD.add(resp)
        else:
          conjD.add(resp)
      else:
        j=0
        while (j<(len(conjC))): #Recibimos el conjunto respuesta
        ↪  del teclado.
          j+=1
          resp=input()
          if (resp=='+'):
            repetir=True
            break
          if (resp=='*'):
            listA=list(conjA)
            listB=list(conjB)
            time1E2=time.time()
            EXPLORACONJUNTOS (listB, listA, 2)
            time2E2=time.time()
            time1F2+=time2E2-time1E2
            time1SE2+=time2E2-time1E2
            timeE2+=time2E2-time1E2
            nE2+=1
            print('Continua tu respuesta:')
```

```
            soundpideresp = playsound(ruta+'/PideResp.mp3')
            j-=1
          else:
            conjD.add(resp)
      if (conjC==conjD and repetir==False): #Revisamos si el
      ↪  usuario respondió adecuadamente.
        timeFF2=time.time()
        incorrecto=False
        print("      ¡RESPUESTA CORRECTA!\n")
        soundrespuestacorrecta =
        ↪  playsound(ruta+'/RespuestaCorrecta.mp3')
      elif repetir==True:
        soundrepetir = playsound(ruta+'/Repetir.mp3')
        print('''Repitamos el ejercicio.''')
        nR2+=1
      else:
        print("      ¡RESPUESTA INCORRECTA!\n")
        soundrespuestacorrecta =
        ↪  playsound(ruta+'/RespuestaIncorrecta.mp3')
        err2+=1
    timeFA+=(timeFF-time1F)+(timeFF2-time1F2)
    timeSEA+=(timeFF-time1SE)+(timeFF2-time1SE2)
    timeEA+=timeE+timeE2
    timeCEA+=((timeFF-time1SE)+timeE)+((timeFF2-time1SE2)+timeE2)
    nRA+=nR+nR2
    nEA+=nE+nE2
    nREA+=(nR+nE)+(nR2+nE2)
    errA+=err+err2
    Fila_Datos=[conjC, (timeFF-time1F)+(timeFF2-time1F2), timeFA,
    (timeFF-time1SE)+(timeFF2-time1SE2), timeSEA, timeE+timeE2,
    ↪  timeEA,
    ((timeFF-time1SE)+timeE)+((timeFF2-time1SE2)+timeE2), timeCEA,
    ↪  nR+nR2, nRA, nE+nE2, nEA,
    (nR+nE)+(nR2+nE2), nREA, err+err2, errA]
    Filas_Datos.append(Fila_Datos)
  soundretroa1 = playsound(ruta+'/Retroa1.mp3')
  print('''En esta actividad se realizaron''')
  DICTANUMERO (str(npreg3))
  soundretroa2 = playsound(ruta+'/Retroa2.mp3')
  print('''preguntas, y tus errores fueron''')
  DICTANUMERO (str(errA))
  if (errA<(npreg3/2)):
    soundrfelicitaciones3 = playsound(ruta+'/Felicitaciones3.mp3')
    print('''Felicitaciones. Haz concluido la 'Evaluación 3'.''')
    soundsiguienteactividad =
    ↪  playsound(ruta+'/SiguienteActividad.mp3')
    print('''Puedes pasar a la siguiente actividad.''')
  else:
    soundintentanuevamente =
    ↪  playsound(ruta+'/IntentaNuevamente.mp3')
    print('''Por la cantidad de respuestas incorrectas que tuviste
    ↪  en esta actividad te
    sugerimos volver a realizarla antes de pasar a la
↪  siguiente.''')
  df_dat3=pd.DataFrame(Filas_Datos, columns=['Conjunto respuesta',
  ↪  'Tiempo final en contestar
  correctamente', 'Tiempo final en contestar correctamente
  ↪  acumulado', 'Tiempo sin exploración
  por pregunta', 'Tiempo sin exploración acumulado', 'Tiempo de
  ↪  exploración por pregunta',
  'Tiempo de exploración acumulado', 'Tiempo con exploración por
  ↪  pregunta', 'Tiempo con
  exploración acumulado', 'Número de repeticiones por pregunta',
  ↪  'Número de repeticiones
  acumulado', 'Número de exploraciones por pregunta', 'Número de
  ↪  exploraciones acumulado',
  'Número de repeticiones más exploraciones por pregunta', 'Número
  ↪  de repeticiones más
  exploraciones acumulado', 'Errores por pregunta', 'Errores
  ↪  acumulados'])
  df_dat3
  df_dat3.to_csv(ruta+'/CD3/dat3_D_.csv')

elif act=='4':
  Fila_Datos=[]
  Filas_Datos=[]
  timeFA=0
  timeSEA=0
  timeEA=0
  timeCEA=0
  nRA=0
  nEA=0
  nREA=0
  errA=0
  soundevaluacion4 = playsound(ruta+'/Evaluacion4.mp3')
  print('''Haz ingresado a la 'Evaluación 4'. En esta parte se
  ↪  evaluará la diferencia de
  tres conjuntos con la posibilidad de que haya conjuntos
↪  vacíos.''')
  i=0
  while i<npreg4: #Con esta instrucción se creará un bucle
  ↪  dependiendo que cuantas preguntas
  se requieran en la actividad.
    conjA=set() #Creamos un primer conjunto de números aleatorios.
    conjB=set() #Creamos un segundo conjunto de números
    ↪  aleatorios.
    conjC=set() #Creamos un tercer conjunto de números aleatorios.
    LLENARCONJUNTOS(conjA,7,8)
    LLENARCONJUNTOS(conjB,3,8)
    LLENARCONJUNTOS(conjC,3,8)
    conjD=set()
    conjD=conjA-conjB-conjC #Creamos el conjunto diferencia
    ↪  (Respuesta)
    i+=1
    print('   PREGUNTA', i)
    print('Conjunto A:', conjA)
    print('Conjunto B:', conjB)
    print('Conjunto C:', conjC)
    print('Conjunto Diferencia (Respuesta):', conjD)
    incorrecto=True
    repetir=False
    time1SE=time.time()
    timeE=0
    nR=0
    nE=0
    err=0
    while incorrecto: #Ponemos un while para que se repita la
    ↪  pregunta hasta que sea contestada
    correctamente
      time1R=time.time()
      soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')
      DICTACONJUNTO(conjA)
      soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')
      DICTACONJUNTO(conjB)
      soundconjuntoC = playsound(ruta+'/ConjuntoC.mp3')
      DICTACONJUNTO(conjC)
      conjE=set()
      print('Escribe tu respuesta:')
      time1=time.time()
      if (repetir==False):
        time1r=time.time()
      soundpideresp = playsound(ruta+'/PideResp.mp3')
      repetir=False
      time1F=time.time()
      time2R=time.time()
      time1SE+=time2R-time1R
      if (len(conjD)==0):
        resp=input()
        if (resp=='+'):
          repetir=True
        elif (resp==''):
          conjE=set()
        elif (resp=='*'):
          listA=list(conjA)
          listB=list(conjB)
          listC=list(conjC)
          time1E=time.time()
          EXPLORACONJUNTOS2 (listA, listB, listC, 1)
          time2E=time.time()
          time1F+=time2E-time1E
          time1SE+=time2E-time1E
          timeE+=time2E-time1E
          nE+=1
          print('Continua tu respuesta:')
          soundpideresp = playsound(ruta+'/PideResp.mp3')
          resp=input()
          if (resp==''):
            conjE=set()
          else:
            conjE.add(resp)
        else:
          conjE.add(resp)
      else:
        j=0
        while (j<(len(conjD))): #Recibimos el conjunto respuesta
        ↪  del teclado.
          j+=1
          resp=input()
          if (resp=='+'):
            repetir=True
            break
          if (resp=='*'):
            listA=list(conjA)
            listB=list(conjB)
            listC=list(conjC)
            time1E=time.time()
            EXPLORACONJUNTOS2 (listA, listB, listC, 1)
            time2E=time.time()
            time1F+=time2E-time1E
            time1SE+=time2E-time1E
            timeE+=time2E-time1E
            nE+=1
            print('Continua tu respuesta:')
            soundpideresp = playsound(ruta+'/PideResp.mp3')
            j-=1
          else:
            conjE.add(resp)
      if (conjD==conjE and repetir==False): #Revisamos si el
      ↪  usuario respondió adecuadamente.
        timeFF=time.time()
        incorrecto=False
        print("      ¡RESPUESTA CORRECTA!\n")
```

```
        soundrespuestacorrecta =
        ↪ playsound(ruta+'/RespuestaCorrecta.mp3')
      elif repetir==True:
        soundrepetir = playsound(ruta+'/Repetir.mp3')
        print('''Repitamos el ejercicio.''')
        nR+=1
      else:
        print("      ¡RESPUESTA INCORRECTA!\n")
        soundrespuestacorrecta =
        ↪ playsound(ruta+'/RespuestaIncorrecta.mp3')
        err+=1
    timeFA+=timeFF-time1F
    timeSEA+=timeFF-time1SE
    timeEA+=timeE
    timeCEA+=(timeFF-time1SE)+timeE
    nRA+=nR
    nEA+=nE
    nREA+=nR+nE
    errA+=err
    Fila_Datos=[conjC, timeFF-time1F, timeFA, timeFF-time1SE,
    ↪ timeSEA, timeE, timeEA,
    (timeFF-time1SE)+timeE, timeCEA, nR, nRA, nE, nEA, nR+nE, nREA,
    ↪ err, errA]
    Filas_Datos.append(Fila_Datos)
  soundretroa1 = playsound(ruta+'/Retroa1.mp3')
  print('''En esta actividad se realizaron''')
  DICTANUMERO (str(npreg4))
  soundretroa2 = playsound(ruta+'/Retroa2.mp3')
  print('''preguntas, y tus errores fueron''')
  DICTANUMERO (str(errA))
  if (errA<(npreg4/2)):
    soundrfelicitaciones4 = playsound(ruta+'/Felicitaciones4.mp3')
    print('''Felicitaciones. Haz concluido la 'Evaluación 4'.''')
    soundsiguienteactividad =
    ↪ playsound(ruta+'/SiguienteActividad.mp3')
    print('''Puedes pasar a la siguiente actividad.''')
  else:
    soundintentanuevamente =
    ↪ playsound(ruta+'/IntentaNuevamente.mp3')
    print('''Por la cantidad de respuestas incorrectas que tuviste
    ↪ en esta actividad te
    sugerimos volver a realizarla antes de pasar a la
↪ siguiente.''')
  df_dat4=pd.DataFrame(Filas_Datos, columns=['Conjunto respuesta',
  ↪ 'Tiempo final en contestar
  correctamente', 'Tiempo final en contestar correctamente
  ↪ acumulado', 'Tiempo sin exploración
  por pregunta', 'Tiempo sin exploración acumulado', 'Tiempo de
  ↪ exploración por pregunta',
  'Tiempo de exploración acumulado', 'Tiempo con exploración por
  ↪ pregunta', 'Tiempo con
  exploración acumulado', 'Número de repeticiones por pregunta',
  ↪ 'Número de repeticiones
  acumulado', 'Número de exploraciones por pregunta', 'Número de
  ↪ exploraciones acumulado',
  'Número de repeticiones más exploraciones por pregunta', 'Número
  ↪ de repeticiones más
  exploraciones acumulado', 'Errores por pregunta', 'Errores
  ↪ acumulados'])
  df_dat4
  df_dat4.to_csv(ruta+'/CD4/dat4_D_.csv')


elif act=='5':
  i=0
  while i<npreg0: #Con esta instrucción se creará un bucle
  ↪ dependiendo que cuantas preguntas
  se requieran en la actividad.
    conjA=set() #Creamos un primer conjunto de números aleatorios.
    conjB=set() #Creamos un segundo conjunto de números
    ↪ aleatorios.
    LLENARCONJUNTOS(conjA,3,4)
    LLENARCONJUNTOS(conjB,3,4)
    conjC=set()
    conjC=conjA-conjB #Creamos el conjunto intersección
    ↪ (Respuesta).
    if len(conjC)!=0:
      i+=1
      print('  EJERCICIO DE PRÁCTICA', i)
      print('Conjunto A:', conjA)
      print('Conjunto B:', conjB)
      print('Conjunto Intersección (Respuesta):', conjC)
      incorrecto=True
      while incorrecto: #Ponemos un while para que se repita la
      ↪ pregunta hasta que sea
      contestada correctamente.
        soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')
        DICTACONJUNTO(conjA)
        soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')
        DICTACONJUNTO(conjB)
        conjD=set()
        soundconjuntoresp = playsound(ruta+'/ConjuntoResp.mp3')
        print('''La respuesta es:''')
        DICTACONJUNTO(conjC)
        print('Escribe tu respuesta:')
        soundprideresp = playsound(ruta+'/PideResp.mp3')
        repetir=False
        j=0
        while (j<(len(conjC))): #Recibimos el conjunto respuesta
        ↪ del teclado.
          j+=1
          resp=input()
          if (resp=='+'):
            repetir=True
            break
          if (resp==''):
            break
          if (resp=='*'):
            listA=list(conjA)
            listB=list(conjB)
            EXPLORACONJUNTOS (listA, listB, 1)
            print('Continua tu respuesta:')
            soundprideresp = playsound(ruta+'/PideResp.mp3')
            j-=1
          else:
            conjD.add(resp)
        if conjC==conjD: #Revisamos si el usuario respondió
        ↪ adecuadamente.
          incorrecto=False
          print("      ¡RESPUESTA CORRECTA!\n")
          soundrespuestacorrecta =
          ↪ playsound(ruta+'/RespuestaCorrecta.mp3')
        elif repetir==True:
          soundrepetir = playsound(ruta+'/Repetir.mp3')
          print('''Repitamos el ejercicio.''')
        else:
          print("      ¡RESPUESTA INCORRECTA!\n")
          soundrespuestaincorrecta =
          ↪ playsound(ruta+'/RespuestaIncorrecta.mp3')
      soundmas = playsound(ruta+'/Mas.mp3')
      print('''Si deseas otro ejercicio de práctica presiona la
      ↪ tecla de '+' seguido de un
      enter, sino solo enter.''')
      mas=input('¿Practicar de nuevo?:')
      if (mas=='+'):
        i-=1
  elif act=='6':
    print("Estás saliendo.")
  else:
    print("Opción erronea")
    soundrespuestaincorrecta =
    ↪ playsound(ruta+'/RespuestaIncorrecta.mp3')
soundsalida = playsound(ruta+'/Salida.mp3')
print('''Estás saliendo del programa para la enseñanza y evaluación
↪ de la diferencia. Muchas
gracias por participar.''')
```

A.4. Complemento

```
#bibliotecas
import time
from datetime import datetime
import pandas as pd
from playsound import playsound
from tkinter import Tk, Label, Button
import random as rd
import math
import os

#definiciones
#función para el audio
direc=os.getcwd()
def elemetosconj(conj):
    for r in conj:
        if r=="0":
            audin1=playsound(direc + '/0.mp3')
        elif r=="1":
            audin1=playsound(direc + '/1.mp3')
        elif r=="2":
            audin2=playsound(direc + '/2.mp3')
        elif r=="3":
            audin3=playsound(direc + '/3.mp3')
        elif r=="4":
            audin4=playsound(direc + '/4.mp3')
        elif r=="5":
            audin5=playsound(direc + '/5.mp3')
        elif r=="6":
            audin5=playsound(direc + '/6.mp3')
        elif r=="7":
            audin5=playsound(direc + '/7.mp3')
        elif r=="8":
            audin5=playsound(direc + '/8.mp3')
        elif r=="9":
            audin5=playsound(direc + '/9.mp3')
    return(conj)
#función para crear conjunto con números aleatorios
def rellenacon(conj,a,b,n):
    for i in range(n):
        agre=rd.randint(a,b)
        agre=str(agre)
        conj.add(agre)
#Función para crear conjuntos disjuntos
def rellenacondife(conj,a,b,n):
    for i in range(n):
        t=rd.randint(a,b)
        t=str(t)
        if t not in conj:
            conj.add(t)
        else:
            while t not in conj:
                t=rd.randint(a,b)
                t=str(t)
#Crea números diferentes
def creanumdif(a,b):
    x=rd.randint(a,b)
    y=rd.randint(a,b)
    while y==x:
        x=rd.randint(a,b)
        y=rd.randint(a,b)
    x=str(x)
    y=str(y)
    return(x,y)
#def crea conjuntos disjuntos
def creadosconjdis1(conj1,conj2,a,b,n):
    for i in range(n):
        r,w=creanumdif(a,b)
        conj3=conj1 & conj2
        if len(conj3)==0:
            conj1.add(r)
            conj2.add(w)
    return(conj1,conj2)
#Función para crear conjuntos disjuntos
def creadosconjdis(conj1,conj2,a,b,n):
    a=math.floor(a/2)
    b=math.floor(b/2)
    for i in range(n):
        a1=2*rd.randint(a,b)
        b1=1+(2*rd.randint(a,b))
        a1=str(a1)
        b1=str(b1)
        conj1.add(a1)
        conj2.add(b1)
    return(conj1,conj2)
def compleleA(conj1,A):
    if A.issubset(conj1):
        conj1=conj1-A
        #print(conj1)
        if conj1==set():
            print('''el conjunto es vació''')
    else:
        print('''el complemento del vació es el universo''')
        #print(conj1)
    return(conj1)
#función de explorar conjuntos
def explorar(a,b,repe):
    conjAlist=list(a)
    conjBlist=list(b)
    cambiarconj="s"
    tamA=len(conjAlist)
    tamB=len(conjBlist)
    muevedere="d"
    mueveizqu="a"
    posicionA=0
    posicionB=0
    print(conjAlist[0])
    while repe=='*':
        print('''Estás en el primer conjunto''')
        audieje5au=playsound(direc + '/estas1.mp3')
        audieje5au=playsound(direc + '/mover.mp3')
        while cambiarconj=="s":
            usumover=input("¿A dónde te quiere mover?")
            if (usumover=="d" or usumover=="D"):
                posicionA=(posicionA+1) % tamA
                print(conjAlist[posicionA])
                elemetosconj(set(conjAlist[posicionA]))
            elif (usumover=="a" or usumover=="A"):
                posicionA=(posicionA-1) % tamA
                print(conjAlist[posicionA])
                elemetosconj(set(conjAlist[posicionA]))
            elif (usumover=="s" or usumover=="S"):
                    cambiarconj="ss"
            elif usumover=="-":
                print('''Estás saliendo de la exploración''')
                audieje5au=playsound(direc + '/salirexplo.mp3')
                break
            elif usumover is not ('d' or 'a' or 'ss' or 'D' or 'A' or
            ↪  'SS'):
                print('''Tu opción no es valida''')
                audieje5au=playsound(direc + '/novalida.mp3')
        #se sale de while
            #print(conjBlist[0])
        if usumover=='-':
            break
        print('''Estás en el segundo conjunto''')
        audieje5au=playsound(direc + '/estas2.mp3')
        audieje5au=playsound(direc + '/mover.mp3')
        while cambiarconj=="ss":
            usumover=input("¿A dónde te quiere mover?")
        #mover a la derecha
            if (usumover=="d" or usumover=="D"):
                posicionB=(posicionB+1) % tamB
                print(conjBlist[posicionB])
                elemetosconj(set(conjBlist[posicionB]))
        #mover a la izquierdaconjAlist[posicionA]
            elif (usumover=="a" or usumover=="A"):
                posicionB=(posicionB-1) % tamB
                print(conjBlist[posicionB])
                elemetosconj(set(conjBlist[posicionB]))
            elif (usumover=="s" or usumover=="S"):
                cambiarconj="s"
            elif usumover=="-":
                print('''Estás saliendo de la exploración''')
                audieje5au=playsound(direc + '/salirexplo.mp3')
                break
            elif usumover is not ('d' or 'a' or 's' or 'D' or 'A' or
            ↪  'S'):
                print('''Tu opción no es valida''')
                audieje5au=playsound(direc + '/novalida.mp3')
        if repe != '-':
            repe='*'
        else:
             audieje5au=playsound(direc + '/salirexplo.mp3')
             repe='-'
             break
        if usumover =='-':
            break
    return(a,b)

def explorar3(a,b,c,repe):
    conjAlist=list(a)
    conjBlist=list(b)
    conjClist=list(c)
    cambiarconj="s"
    tamA=len(conjAlist)
    tamB=len(conjBlist)
    tamC=len(conjClist)
    muevedere="d"
    mueveizqu="a"
    posicionA=0
```

```
    posicionB=0
    posicionC=0
    print(conjAlist[0])
    print('''Estás en el primer conjunto''')
    while repe=='*':
        audieje5au=playsound(direc + '/estas1.mp3')
        audieje5au=playsound(direc + '/mover.mp3')
        while cambiarconj=="s":

            usumover=input("¿A dónde te quiere mover?")
            if (usumover=="d" or usumover=="D"):
                posicionA=(posicionA+1) % tamA
                elemetosconj(set(conjAlist[posicionA]))
                print(conjAlist[posicionA])
            elif (usumover=="a" or usumover=="A"):
                posicionA=(posicionA-1) % tamA
                elemetosconj(set(conjAlist[posicionA]))
                print(conjAlist[posicionA])
            elif (usumover=="s" or usumover=="S"):
                cambiarconj="ss"
            elif usumover=="-":
                print('''Estás saliendo de la exploración''')
                audieje5au=playsound(direc + '/salirexplo.mp3')
                break
            elif usumover is not ('d' or 'a' or 'ss' or 'D' or 'A' or
            ↪  'SS'):
                print('''Tu opción no es valida''')
                audieje5au=playsound(direc + '/novalida.mp3')
    #se sale de while
        #print(conjBlist[0])
        if usumover=='-':
            break
        print('''Estás en el segundo conjunto''')
        audieje5au=playsound(direc + '/estas2.mp3')
        audieje5au=playsound(direc + '/mover.mp3')
        while cambiarconj=="ss":
            usumover=input("¿A dónde te quiere mover?")
    #mover a la derecha
            if (usumover=="d" or usumover=="D"):
                posicionB=(posicionB+1) % tamB
                elemetosconj(set(conjBlist[posicionB]))
                print(conjBlist[posicionB])

    #mover a la izquierda
            elif (usumover=="a" or usumover=="A"):
                posicionB=(posicionB-1) % tamB
                elemetosconj(set(conjBlist[posicionB]))
                print(conjBlist[posicionB])

            elif (usumover=="s" or usumover=="S"):
                cambiarconj="sss"
            elif usumover=="-":
                print('''Estás saliendo de la exploración''')
                audieje5au=playsound(direc + '/salirexplo.mp3')
                break
            elif usumover is not ('d' or 'a' or 's' or 'D' or 'A' or
            ↪  'S'):
                audieje5au=playsound(direc + '/novalida.mp3')
                print('''Tu opción no es valida''')
        if usumover=='-':
            break
        print('''Estás en el tercer conjunto''')
        audieje5au=playsound(direc + '/estas3.mp3')
        audieje5au=playsound(direc + '/mover.mp3')
        while cambiarconj=="sss":
            usumover=input("¿A dónde te quiere mover?")
    #mover a la derecha
            if (usumover=="d" or usumover=="D"):
                posicionC=(posicionC+1) % tamC
                elemetosconj(set(conjClist[posicionC]))
                print(conjClist[posicionC])
    #mover a la izquierda
            elif (usumover=="a" or usumover=="A"):
                posicionC=(posicionC-1) % tamC
                elemetosconj(set(conjClist[posicionC]))
                print(conjClist[posicionC])
            elif (usumover=="s" or usumover=="S"):
                cambiarconj="s"
            elif usumover=="-":
                print('''Estás saliendo de la exploración''')
                audieje5au=playsound(direc + '/salirexplo.mp3')
                break
            elif usumover is not ('d' or 'a' or 's' or 'D' or 'A' or
            ↪  'S'):
                print('''Tu opción no es valida''')
                audieje5au=playsound(direc + '/novalida.mp3')
        if repe != '-':
            repe='*'
        else:
             audieje5au=playsound(direc + '/salirexplo.mp3')
             repe='-'
             break
        if usumover =='-':
            break

    return(a,b,c)
```

```
    #llenas un conjunto por el usuario
    def respconj(conj, n,a,b, conjresp):
        numr=0
        tiem=0
        nt=0
        while len(conj)<n:
            if conjresp==set():
                print('''El complemento es vació''')
                break

            print('''Ingresa los elementos uno por uno con un enter''')
            uu=input("")
            if uu=='+':
                print('''Se volverá a repetir el ejercicio''')
                audis11=playsound(direc + '/repetirejercicio.mp3')
                break
            if uu=='*':
                numr=numr+1
                ta=time.time()
                a,b=explorar(a,b,'*')
                tb=time.time()
                nt=tb-ta
                audis11=playsound(direc + '/pide.mp3')
            else:
                conj.add(uu)
        return(conj,nt,numr)

    def respconj3(conj, n,a,b,c, conjresp):
        numr=0
        tiem=0
        nt=0
        while len(conj)<n:
            print('''Ingresa los elementos uno por uno con un enter''')
            uu=input("")

            if uu=='+':
                print('''Se volverá a repetir el ejercicio''')
                audis11=playsound(direc + '/repetirejercicio.mp3')
                break
            if uu=='*':
                numr=numr+1
                ta=time.time()
                a,b,c=explorar3(a,b,c,'*')
                tb=time.time()
                tiem=tb-ta
                audis11=playsound(direc + '/pide.mp3')

            if conjresp==set():
                print('''el complemento es vacío''')
                break

            else:
                conj.add(uu)
            nt=tiem+nt

        return(conj,nt, numr)

    #numero de preguntas

    audis11=playsound(direc + '/bienvenidoc.mp3')
    repetir="+"
    print('''menu''')
    #numpreg=(input("ingrese el numero de preguntas"))
    numpreg=10
    print('''Bienvenido a las actividades del complemento''')

    print('''En este programa realizaras diversas actividades que te
    ↪  guíen en el aprendizaje
    del  conjunto complemento, la dificultad de los niveles agrega
    ↪  matices conceptuales. Se
    recomienda las hagas en orden. Para ingresar tus respuestas, deberás
    ↪  esperar a que
    acabe el audio y posteriormente agregar cada elemento del conjunto
    ↪  con un enter. Solo
    se utilizaran los números del cero al nueve. Si deseas repetir
    ↪  cualquier instrucción
    aprieta el signo de +''')
    ##agregar que debe apretar enter para continuar. cada que se detenga
    ↪  el audio significa
    que debes apretar enter o signo de mas para continuar
    audis11=playsound(direc + '/instru1.mp3')
    audis11=playsound(direc + '/enter.mp3')
    menu="0"
    print('''Aprieta el numero del nivel que deseas cursas \
        has ingresado al menú: presiona 0 para actividad de aprendizaje,
    ↪  1 calcular
        el complemento de un elemento, 2 para calcular el complemento de
    ↪  un conjunto :
        3 calcular el complemento de un  complemento,  4 para calcular el
    ↪  complemento
        cuando el conjunto universo tiene mas de 3 conjuntos,  y en el 5
    ↪  aprenderás a
        ingresar respuestas, digita cualquier otro numero para salir\
      ''')
    audidefiau=playsound(direc + '/menu.mp3')
    while (menu=="0" or menu=="1" or menu=="2" or menu=="3" or menu=="4"\
```

```
       or menu=='5' or menu=="6" or menu=="7"):
    audis14=playsound(direc + '/menu2.mp3')
    print('''Estas en el menu, igresa tu opción''')
    menu=input("ingresa tu opción ")


    if menu=="0":
        print('''Actividad didactica del complemento''')

        audiresc=playsound(direc + '/dida1.mp3')
########---Actividad didatica---#####################################
        eleccion='''¿Cómo quieres estudiar de forma ordenada, es decir, primero
        la definición, posteriormente ejemplos. o de quieres entrar a una  parte
        especifica de la didáctica donde tu elegirás tu orden. presiona a para la
        primera opción y d para la segunda '''
        audieje5au=playsound(direc + '/eleccion.mp3')
        bande=True
        while bande==True:
            elec=input("ingresa tu opcion ")
            if (elec=='a' or elec=='d'):
                bande=False
            else:
                print('''opcion no valida''')
                audieje5au=playsound(direc + '/novalida.mp3')
        if elec=='a':
            audieje5au=playsound(direc + '/opcionA.mp3')
            definicion='''El conjunto complemento de A esta compuesto por todos los
            conjuntos o elementos que no pertenecen al conjunto A. En otras palabras,
            el complemento es todo lo que no es A.'''
            audidefiau=playsound(direc + '/defi1.mp3')
            bande1=True
            while bande1==True:

                audidefiau=playsound(direc + '/repemasenter.mp3')
                respdida=input("si quieres repetir la definición ingresa el signo +,
                de lo contrario da un enter para continuar")
                if respdida=="+":
                    print(definicion)
                    audidefiau=playsound(direc + '/defi1.mp3')
                elif respdida != "+":
                    bande1=False
                    print('''Sin repetición''')
            Ejemplo1='''Existe una manera más sencilla de calcular el conjunto
            complemento, lo podemos ver como una diferencia del conjunto universo y el
            conjunto del que quieres calcular el complemento. Imaginemos que tenemos
            una cubeta llena de canicas de distintos tamaños y queremos sacar el
            complemento de las más pequeñas. La cubeta es el conjunto universo que
            engloba todas las canicas, para hacer la diferencia o resta, le quitamos
            todas las canicas pequeñas y lo que nos queda es el complemento de las
            canicas pequeñas. observemos que el complemento tiene distintos elementos:
            canicas grandes, extra grandes, etc, excepto las canicas pequeñas.'''
            audieje1au=playsound(direc + '/defi2.mp3')
            bande1=True
            while bande1==True:

                audidefiau=playsound(direc + '/repemasenter.mp3')
                respdida=input("si quieres repetir el ejemplo 1 ingresa el signo +")
                if respdida=="+":
                    print(definicion)
                    audidefiau=playsound(direc + '/defi2.mp3')
                elif respdida != "+":
                    bande1=False
            Ejemplo2=''' Es importante notar que todo conjunto tiene conjunto complemento,
            de ahí su importancia.  Inclusive el complemento del complemento de un
            conjunto existe, a esto se le conoce como involución. Este ultimo es el
            conjunto en sí mismo, pues supongamos que queremos el complemento de A
            que por definición es un resta del conjunto universo  con el conjunto A,
            volviendo a calcular su complemento volvemos a hacer la resta pero ahora
            del universo con A complemento de lo cual resulta A. Es como una doble
            negación, por ejemplo, cuando te dicen no es el caso que no sea posible,
            quiere decir, es posible .'''
            audieje2au=playsound(direc + '/defi3.mp3')
            bande1=True
            while bande1==True:
                audidefiau=playsound(direc + '/repemasenter.mp3')
                respdida=input("si quieres repetir el ejemplo 2 ingresa el signo +")
                if respdida=="+":
                    print(definicion)
                    audidefiau=playsound(direc + '/defi3.mp3')
                elif respdida != "+":
                    bande1=False
            Ejemplo3='''Ejemplo 1. Regresemos al caso de los seres humanos y se quiere
            el complemento de las mujeres. Mi conjunto universo son todos los seres
            humanos,  tomo el conjunto universo (hombre-mujeres) y después le resto
            todas las mujeres, de donde solo te quedan los hombres. Así, el complemento
            de las mujeres son los hombres'''
            audieje3au=playsound(direc + '/eje1.mp3')
            bande1=True
            while bande1==True:
                audidefiau=playsound(direc + '/repemasenter.mp3')
                respdida=input("si quieres repetir el ejemplo3 ingresa el signo +")
                if respdida=="+":
                    print(definicion)
                    audidefiau=playsound(direc + '/eje1.mp3')
                elif respdida != "+":
                    bande1=False
            Ejemplo4='''Ejemplo2. ¿Entonces, cual es el complemento del conjunto
            universo? Para saber hacemos la resta del universo menos el universo de
            lo que resulta el vacío.'''
            audieje4au=playsound(direc + '/ejeuniconj.mp3')
            bande1=True
            while bande1==True:
                audidefiau=playsound(direc + '/repemasenter.mp3')
                respdida=input("si quieres repetir el ejemplo 4 ingresa el signo +")
                if respdida=="+":
                    print(definicion)
                    audidefiau=playsound(direc + '/ejeuniconj.mp3')
                elif respdida != "+":
                    bande1=False
            Ejemplo5=''' Ejemplo 3 con números. Imaginemos que tienes el conjunto de los
            diez primeros números 1,2,3, hasta el 10, es decir, ese es tu conjunto
            universo, ahora queremos el complemento del numero 2, el cual queda como
            1,3,4,5, hasta el 10, sin el dos. pongamos un poco mas de dificultad en
            el mismo conjunto. Queremos calcular el complemento de los los números
            pares que son 2,4,6,8,10,  haciendo el complemento de los números pares
            me queda el conjunto 1,3,5,7,9. '''
            audieje5au=playsound(direc + '/eje2.mp3')
            bande1=True
            while bande1==True:
                audidefiau=playsound(direc + '/repemasenter.mp3')
                respdida=input("si quieres repetirel ejemplo 5 ingresa el signo +")
                if respdida=="+":
                    print(definicion)
                    audidefiau=playsound(direc + '/eje2.mp3')
                elif respdida != "+":
                    bande1=False
            Ejemplo6=''' Ejemplo 4 con números. Nuestro conjunto universo esta
            compuesto por varios conjuntos, a saber: el conjunto de los múltiplos de
            2=2,4,6,8,10,12..., y de los múltiplos de 3=3,6,9,12,15,18..., es decir,
            tenemos una infinidad de números, pero no todos los números, por ejemplo
            no tenemos el 5 o el 7 entre otra infinidad de números. Supongamos que
            queremos el complemento de los múltiplos del 6. Entonces debemos hacer
            un diferencia para decir cuales son los elementos del complemento de los
            múltiplos de 6. Nuestro universo esta conformado por los dos  números
            divisibles entre 2 y 3, quitando los los divisibles entre 6, los primeros
            elementos del complemento serian son: 2,3,4,8,9,10,14... etc'''
            audieje6au=playsound(direc + '/eje3.mp3')
            bande1=True
            while bande1==True:

                audidefiau=playsound(direc + '/repemasenter.mp3')
```

```
                respdida=input("si quieres repetir el ejemplo 6 ingresa el signo +")
                if respdida=="+":
                    print(definicion)
                    audidefiau=playsound(direc + '/eje3.mp3')
                elif respdida != "+":
                    bande1=False
        banderae=True
        while banderae==True:
            if elec=='d':
                print('''Has elegido tu orden, las opciones van del 1 al 6, del 1 al
                3 corresponden a la definición y del 4 al 6 a ejemplos, preciosa el
                numero que deseas escuchar. para salir cualquier otro numero''')
                audieje5au=playsound(direc + '/elecciond.mp3')
                audidefiau=playsound(direc + '/didarepetodo.mp3')
            if elec=='a':
                repe='''Si deseas volver a escuchar la definición formal aprieta 0,o
                del 1 al 6 para volver a escuchar algún ejemplo en concreto, para
                ilustrar, si gustas escuchar el ejemplo 3 aprieta el numero 3. para
                continuar sin repetición presiona  cualquier otro numero '''
                audidefiau=playsound(direc + '/didarepetodo.mp3')
            audidefiau=playsound(direc + '/opcioningre.mp3')
            respdida=input("ingresa tu opcion de ejemplos")
            if respdida=="0":
                audidefiau=playsound(direc + '/defi1.mp3')
            elif respdida=="1":
                audieje1au=playsound(direc + '/defi2.mp3')
            elif respdida=="2":
                audieje2au=playsound(direc + '/defi3.mp3')
            elif respdida=="3":
                audieje3au=playsound(direc + '/eje1.mp3')
            elif respdida=="4":
                audieje4au=playsound(direc + '/ejeuniconj.mp3')
            elif respdida=="5":
                audieje5au=playsound(direc + '/eje2.mp3')
            elif respdida=="6":
                eje5au=playsound(direc + '/eje3.mp3')
            else:
                #print('''presiona enter")
                banderae=False
        ejemplo_instrucciones='''A continuación haremos un ejercicio para que te
        familiarices con el programa, para ello calcularemos el complemento de un
        conjunto  '''
        audiresc=playsound(direc + '/ejeminstru2.mp3')
        audiresc=playsound(direc + '/didaintrudos.mp3')
        audieje5au=playsound(direc + '/explorar.mp3')
        audieje5au=playsound(direc + '/explirepe.mp3')
        conjeje1=set()
        conjeje=set()
        #llenar conjuntos
        print('''podrás ingresar tus respuestas después de este sonido___cada que
        ingreses un elemento da un enter, es decir, si el conjunto respuesta conta de los
        elementos 1,2,3, entonces agrega el 1, enter, 2 enter, 3 y enter. si la respuesta
        es correcta se escuchara este sonido___ y en caso contrario este otro sonido ''')
        print('''para explorar los elementos de los conjuntos presiona *, la d para
        la derecha y la a para la izquierda, s para cambiar de conjunto, para salir -''')
        rellenacon(conjeje,0,9,4)
        e=rd.choice(list(conjeje))
        conjeje1.add(e)
        conjeje2=compleleA(conjeje,conjeje1)
        bandera=True
        while bandera==True:
            print(conjeje)
            audiresc=playsound(direc + '/conjA.mp3')
            elemetosconj(conjeje)
            print(conjeje1)
            audiresc=playsound(direc + '/conjB.mp3')
            elemetosconj(conjeje1)
            audis11=playsound(direc + '/pide.mp3')
            conjres=set()
            respconj(conjres,
            len(conjeje2),conjeje,conjeje1,conjeje2)
            if conjeje2==conjres:
                bandera=False
                audis11=playsound(direc + '/rescorre.mp3')
                print('''Es correcta tu respuesta''')
            else:
                if len(conjres)==len(conjeje2):
                    audis11=playsound(direc + '/resincorre.mp3')
                    print('''vuelve a intentarlo''')
                conjres.clear()
        print('''Estas listo para iniciar las actividades de evaluación para la
        unión. Están divididas en 4 niveles escoge el nivel que te gustaría practicar
        las veces que sean necesarias. ''')
        audis11=playsound(direc + '/findida.mp3')
########---Nivel uno---#####################################
    elif menu=="1":
        print('''Nivel 1''')
        a1=time.time() #paratomar el tiempo completo

        print('''Bienvenido al nivel 1. En este nivel deberás calcular el complemento
        de un elemento del conjunto A. Debes suponer que A es el conjunto universo ''')
        audidefiau=playsound(direc + '/nivel1intru.mp3')
        ##################listas de las variables
        conjuntorespuesta=[]
        #1 conjunto respuesta
        tiempoxpreguntasin=[]
        #2 tiempo por pregunta sin repetición ni acumulado
        tiemacumupoxpreguntasin=[]
        #3 tiempo acumulado sin repetición ni acumulado
        tiemporepexpreguntacon=[]
        #4 tiempo por repeticiones por pregunta
        tiempoacumurepexpreguntacon=[]
        #5 tiempo por repeticiones acumulado
        tiempoexploestrella=[]
        #6 tiempo con exploración por pregunta
        tiempoexploestrellaacumulado=[]
        #7 tiempo acumulado con exploración
        repesumaestrella=[]
        #8 tiempo con repetición y exploración por pregunta
        repesumaestrellaacumulado=[]
        #9 tiempo acumulado con repetición y exploración
        numrepesuma=[]
        #10 número de repeticiones por pregunta
        numrepesumaacumu=[]
        #11 número de repeticiones acumulado
        numexploxpreg=[]
        #12 numero de exploraciones por pregunta
        numexploacumu=[]
        #13 numero de exploraciones acumulada
        numsumaexploxpregunta=[]
        #14 numero de suma de repetición y exploración por pregunta
        numsumaexploacumu=[]
        #15 numero de suma de repetición y exploración acumulado
        incorrestas1=[]
        #16 errores por pregunta
        incoacumuladas=[]
        #17 errores por pregunta acumulada
        mal=0
        acuti=0
        acutire=0
        repemunacu=0
        ntacumu=0
        tsumarepeacumu=0
        ndexacumu=0
        ndexyrepeacumu=0
        for i in range(numpreg):
            #crear dos conjuntos aleatorios de 1 elementos
            conj1=set()
            rellenacondife(conj1,0,9,3)
            #enunciar los elementos del conjunto por separado
            e=rd.choice(list(conj1))
            conj2=set()
            conj2.add(str(e))
            conj3=compleleA(conj1,conj2)
            c=conj3
            #hacer la pregunta
            bandera=True
            malacumu=0
            timecal=0
            repenum=0
            ntxp=0
            tsumarepe=0
            ndexp=0
            ndexyrepe=0
            conj4=set()
            while bandera==True:
                print('''Se tiene tiene el conjunto A''')
                audiresc=playsound(direc + '/conjA.mp3')
                print(conj1)
                elemetosconj(conj1)
                print('''saca el complemento de elemento''')
                audiresc=playsound(direc + '/conjB.mp3')
                print(conj2)
                elemetosconj(conj2)
                print('''sonido de respuesta''')
                audis11=playsound(direc + '/pide.mp3')
                a=time.time()
                conj4, nt,numr=respconj(conj4,len(conj3),conj1, conj2, conj3)
                b=time.time()
                cal=b-a-nt
                calre=0
```

```
                ntxp=ntxp+nt
                ntacumu=ntacumu+nt
                ndexp=ndexp+numr
                if conj3==conj4:
                    bandera=False
                    print('''respuesta correcta''')
                    audiresc=playsound(direc + '/rescorre.mp3')
                    acutire=acutire+timecal+cal
                    acuti=acuti+cal
                    calre=cal+timecal
                    tsumarepe=ntxp+calre
                    tsumarepeacumu=tsumarepeacumu+tsumarepe
                    ndexacumu=ndexacumu+ndexp
                    ndexyrepe=repenum+ndexp
                    ndexyrepeacumu=ndexyrepeacumu+ndexyrepe
                else:
                    if len(conj3)==len(conj4):
                        print('''vuelve a intentar''')
                        audiresin=playsound(direc +
                        ↪  '/resincorre.mp3')
                        mal=mal+1
                        malacumu=malacumu+1
                    timecal=timecal+cal
                    conj4.clear()
                    repenum=repenum+1
            nt=ntxp
            repemunacu=repemunacu+repenum
            conjuntorespuesta.append(c)
            tiempoxpreguntasin.append(cal)
            tiemporepexpreguntacon.append(calre)
            tiemacumupoxpreguntasin.append(acuti)
            tiempoacumurepexpreguntacon.append(acutire)
            tiempoexploestrella.append(nt)
            tiempoexploestrellaacumulado.append(ntacumu)
            repesumaestrella.append(tsumarepe)
            repesumaestrellaacumulado.append(tsumarepeacumu)
            numrepesuma.append(repenum)
            numrepesumaacumu.append(repemunacu)
            numexploxpreg.append(ndexp)
            numexploacumu.append(ndexacumu)
            numsumaexploxpregunta.append(ndexyrepe)
            numsumaexploacumu.append(ndexyrepeacumu)
            incorrestas1.append(mal)
            incoacumuladas.append(malacumu)
        print('''Notamos que aunque calculemos el complemento de
        ↪  varios conjuntos,
        siempre debemos usar la misma regla para poder
↪   calcularlos''')
        audiresin=playsound(direc + '/retro1.mp3')
        promedio=numpreg/2
        if mal==0:
            print('''Increible no te has equivocado''')
            audiresin=playsound(direc + '/increible.mp3')
        elif mal<=promedio:
            print('''Buen trabajo''')
        elif promedio>mal>=promedio*3:
            print('''Te sugerimos volver a realizar el nivel''')
            audiresin=playsound(direc + '/repenivel.mp3')
        else:
            print('''Te sugerimos volver a estudiar la activdad
            ↪  didactica por el numero
            de respuestas incorrectas''')
            audiresin=playsound(direc + '/repeincorre.mp3')

        names=['Conjunto respuesta',
               'Tiempo final en contestar correctamente',
               'Tiempo final en contestar correctamente acumulado',
               'Tiempo sin exploración por pregunta',
               'Tiempo sin exploración acumulado',
               'Tiempo de exploración por pregunta',
               'Tiempo de exploración acumulado',
               'Tiempo con exploración por pregunta',
               'Tiempo con exploración acumulado',
               'Número de repeticiones por pregunta',
               'Número de repeticiones acumulado',
               'Número de exploraciones por pregunta',
               'Número de exploraciones acumulado',
               'Número de repeticiones más exploraciones por
               ↪  pregunta',
               'Número de repeticiones más exploraciones acumulado',
               'Errores por pregunta', 'Errores acumulados']
        df=pd.DataFrame(list(zip(conjuntorespuesta,
        tiempoxpreguntasin, tiemacumupoxpreguntasin,
        ↪  tiemporepexpreguntacon,
        tiempoacumurepexpreguntacon, tiempoexploestrella,
        ↪  tiempoexploestrellaacumulado,
        repesumaestrella,repesumaestrellaacumulado, numrepesuma,
        ↪  numrepesumaacumu,
        numexploxpreg, numexploacumu, numsumaexploxpregunta,
        ↪  numsumaexploacumu,
        incorrestas1, incoacumuladas)), columns=names)
        print(df)
        df.to_csv(direc + '/CC1/dat1_C_.csv')
########---Nivel
↪   dos---##############################################
    elif menu=="2":
        print('''Nivel 2''')
        audis14=playsound(direc + '/nivel2intru.mp3')
        print('''Deberás mencionar el conjunto complemento cuando el
        ↪  conjunto universo
        esta conformado por 2 conjuntos. ''')
        ###################listas de las variables
        conjuntorespuesta=[]
        #1 conjunto respuesta
        tiempoxpreguntasin=[]
        #2 tiempo por pregunta sin repetición ni acumulado
        tiemacumupoxpreguntasin=[]
        #3 tiempo acumulado sin repetición ni acumulado
        tiemporepexpreguntacon=[]
        #4 tiempo por repeticiones por pregunta
        tiempoacumurepexpreguntacon=[]
        #5 tiempo por repeticiones acumulado
        tiempoexploestrella=[]
        #6 tiempo con exploración por pregunta
        tiempoexploestrellaacumulado=[]
        #7 tiempo acumulado con exploración
        repesumaestrella=[]
        #8 tiempo con repetición y exploración por pregunta
        repesumaestrellaacumulado=[]
        #9 tiempo acumulado con repetición y exploración
        numrepesuma=[]
        #10 número de repeticiones por pregunta
        numrepesumaacumu=[]
        #11 número de repeticiones acumulado
        numexploxpreg=[]
        #12 numero de exploraciones por pregunta
        numexploacumu=[]
        #13 numero de exploraciones acumulada
        numsumaexploxpregunta=[]
        #14 numero de suma de repetición y exploración por pregunta
        numsumaexploacumu=[]
        #15 numero de suma de repetición y exploración acumulado
        incorrestas1=[]
        #16 errores por pregunta
        incoacumuladas=[]
        #17 errores por pregunta acumulada
        mal=0
        acuti=0
        acutire=0
        repemunacu=0
        ntacumu=0
        tsumarepeacumu=0
        ndexacumu=0
        ndexyrepeacumu=0
        for i in range(numpreg):
            #print('''Une los siguientes conjuntos''')
            conj1=set()
            print('''conjunto A''')
            rellenacon(conj1,0,9,3)
            conj2=set()
            rellenacon(conj2,0,9,3)
            conj3=conj1 | conj2 #tal vez sea add
            conj4=set()#respuesta ususario
            malacumu=0
            timecal=0
            repenum=0
            ntxp=0
            tsumarepe=0
            ndexp=0
            ndexyrepe=0
            c=conj3#respuesta correcta
            bandera=True
            conj5=set()
            ab=rd.randint(1,2)
            while bandera==True:
                print('''El conjunto A''')
                audis14=playsound(direc + '/conjA2.mp3')
                print(conj1)
                elemetosconj(conj1)
                audis14=playsound(direc + '/conjb2.mp3')
                print(conj2)
                print('''y el conjunto B forman el conjunto universo
                ↪  ''')
                elemetosconj(conj2)
                #suena el tercer conjunto
                if ab==1:
                    print('''calcula el complemento del conjunto
                    ↪  A''')
                    audis14=playsound(direc + '/conjA22.mp3')
                    audis11=playsound(direc + '/pide.mp3')
                    conj5=compleleA(conj3,conj1)
                    a=time.time()
                    conj4, nt, numr=respconj(conj4,len(conj5),
                    ↪  conj2,conj3,conj5)
                    b=time.time()
                elif ab==2:
                    print('''calcula el complemento del conjunto
                    ↪  B''')
                    audis14=playsound(direc + '/conjB22.mp3')
                    audis11=playsound(direc + '/pide.mp3')
                    conj5=compleleA(conj3,conj2)
                    a=time.time()
```

```
                    conj4,nt,numr=
                    ↪  respconj(conj4,len(conj5),conj2,conj3,conj5)
                    b=time.time()
                print('''conjunto respuesta''', conj5)
                cal=b-a-nt
                calre=0
                ntxp=ntxp+nt
                ntacumu=ntacumu+nt
                ndexp=ndexp+numr
                if conj5==conj4:
                    bandera=False
                    audis14=playsound(direc + '/rescorre.mp3')
                    #sonido de correcto
                    acutire=acutire+timecal+cal
                    acuti=acuti+cal
                    calre=cal+timecal
                    tsumarepe=ntxp+calre
                    tsumarepeacumu=tsumarepeacumu+tsumarepe
                    ndexacumu=ndexacumu+ndexp
                    ndexyrepe=repenum+ndexp
                    ndexyrepeacumu=ndexyrepeacumu+ndexyrepe
                else:
                    if len(conj5)==len(conj4):
                        print('''vuelve a intentar''')
                        audiresin=playsound(direc +
                        ↪  '/resincorre.mp3')
                        mal=mal+1
                        malacumu=malacumu+1
                    conj4.clear()
                    timecal=timecal+cal
                    repenum=repenum+1
                    print('''se repetirá el ejercicio''')
            nt=ntxp
            repemunacu=repemunacu+repenum
            conjuntorespuesta.append(c)
            tiempoxpreguntasin.append(cal)
            tiemporepexpreguntacon.append(calre)
            tiemacumupoxpreguntasin.append(acuti)
            tiempoacumurepexpreguntacon.append(acutire)
            tiempoexploestrella.append(nt)
            tiempoexploestrellaacumulado.append(ntacumu)
            repesumaestrella.append(tsumarepe)
            repesumaestrellaacumulado.append(tsumarepeacumu)
            numrepesuma.append(repenum)
            numrepesumaacumu.append(repemunacu)
            numexploxpreg.append(ndexp)
            numexploacumu.append(ndexacumu)
            numsumaexploxpregunta.append(ndexyrepe)
            numsumaexploacumu.append(ndexyrepeacumu)
            incorrestas1.append(mal)
            incoacumuladas.append(malacumu)
        names=['Conjunto respuesta',
               'Tiempo final en contestar correctamente',
               'Tiempo final en contestar correctamente acumulado',
               'Tiempo sin exploración por pregunta',
               'Tiempo sin exploración acumulado',
               'Tiempo de exploración por pregunta',
               'Tiempo de exploración acumulado',
               'Tiempo con exploración por pregunta',
               'Tiempo con exploración acumulado',
               'Número de repeticiones por pregunta',
               'Número de repeticiones acumulado',
               'Número de exploraciones por pregunta',
               'Número de exploraciones acumulado',
               'Número de repeticiones más exploraciones por
               ↪  pregunta',
               'Número de repeticiones más exploraciones acumulado',
               'Errores por pregunta', 'Errores acumulados']
        df=pd.DataFrame(list(zip(conjuntorespuesta,
        tiempoxpreguntasin, tiemacumupoxpreguntasin,
        ↪  tiemporepexpreguntacon,
        tiempoacumurepexpreguntacon, tiempoexploestrella,
        ↪  tiempoexploestrellaacumulado,
        repesumaestrella,repesumaestrellaacumulado, numrepesuma,
        ↪  numrepesumaacumu,
        numexploxpreg, numexploacumu, numsumaexploxpregunta,
        ↪  numsumaexploacumu,
        incorrestas1, incoacumuladas)), columns=names)
        df.to_csv(direc + '/CC2/dat2_C_.csv')
        print('''Notamos que aunque calculemos el complemento de
        ↪  varios conjuntos,
        siempre debemos usar la misma regla para poder
↪  calcularlos''')
        audiresin=playsound(direc + '/retro2.mp3')
        promedio=numpreg/2
        if mal==0:
            print('''increible no te has equivocado''')
            audiresin=playsound(direc + '/increible.mp3')
        elif mal<=promedio:
            print('''Buen trabajo''')
        elif promedio>mal>=promedio*3:
            print('''Te sugerimos volver a realizar el nivel''')
            audiresin=playsound(direc + '/repenivel.mp3')
        else:
            print('''Te sugerimos volver a estudiar la activdad
            ↪  didactica por el numero
                de respuestas incorrectas''')
            audiresin=playsound(direc + '/repeincorre.mp3')
        print(df)
########---Nivel
↪  tres---##############################################
    elif menu=="3":
        print('''Nivel 3''')
        print('''Veremos la propiedad involutiva de complemento, es
        ↪  decir, veamos el
        complemento del complemento\
        aunque parezca repetitivo el ejercicio, es importante notar
↪  esta
        propiedad. Recuerda que es como una doble negación''')
        audis14=playsound(direc + '/nivel3instru.mp3')
        ###################listas de las variables
        conjuntorespuesta=[]
        #1 conjunto respuesta
        tiempoxpreguntasin=[]
        #2 tiempo por pregunta sin repetición ni acumulado
        tiemacumupoxpreguntasin=[]
        #3 tiempo acumulado sin repetición ni acumulado
        tiemporepexpreguntacon=[]
        #4 tiempo por repeticiones por pregunta
        tiempoacumurepexpreguntacon=[]
        #5 tiempo por repeticiones acumulado
        tiempoexploestrella=[]
        #6 tiempo con exploración por pregunta
        tiempoexploestrellaacumulado=[]
        #7 tiempo acumulado con exploración
        repesumaestrella=[]
        #8 tiempo con repetición y exploración por pregunta
        repesumaestrellaacumulado=[]
        #9 tiempo acumulado con repetición y exploración
        numrepesuma=[]
        #10 número de repeticiones por pregunta
        numrepesumaacumu=[]
        #11 número de repeticiones acumulado
        numexploxpreg=[]
        #12 numero de exploraciones por pregunta
        numexploacumu=[]
        #13 numero de exploraciones acumulada
        numsumaexploxpregunta=[]
        #14 numero de suma de repetición y exploración por pregunta
        numsumaexploacumu=[]
        #15 numero de suma de repetición y exploración acumulado
        incorrestas1=[]
        #16 errores por pregunta
        incoacumuladas=[]
        #17 errores por pregunta acumulada
        mal=0
        acuti=0
        acutire=0
        repemunacu=0
        ntacumu=0
        tsumarepeacumu=0
        ndexacumu=0
        ndexyrepeacumu=0
        for k in range(numpreg):
            conj1=set()
            rellenacon(conj1,0,7,3)
            e=rd.choice(list(conj1))
            conj2=set()
            conj2.add(e)
            f=rd.choice(list(conj1))
            conj2.add(f)
            conj3=compleleA(conj1,conj2)
            conj4=set()
            c=conj3
            bandera=True
            malacumu=0
            timecal=0
            repenum=0
            ntxp=0
            tsumarepe=0
            ndexp=0
            ndexyrepe=0
            r=rd.randint(1,2)
            while bandera==True:
                print('''Se tiene el conjunto A ''')
                audis14=playsound(direc + '/conjA2.mp3')
                print(conj1)
                elemetosconj(conj1)
                audis14=playsound(direc + '/conjb2.mp3')
                print('''El conjunto B''')
                print(conj2)
                elemetosconj(conj2)
                conj3=set()
                if r==1:
                    print('''calcula el complemento del complemento
                    ↪  del conjunto a''')
                    audis14=playsound(direc + '/conjA333.mp3')
                    conj3=conj1
                if r==2:
                    print('''calcula el complemento del complemento
                    ↪  del conjunto b''')
                    audis14=playsound(direc + '/conjB333.mp3')
```

```
                conj3=conj2
            audis11=playsound(direc + '/pide.mp3')
            a=time.time()

            conj4,nt,numr=respconj(conj4,len(conj3),conj1,conj2,conj3)
            b=time.time()
            cal=b-a-nt
            calre=0
            ntxp=ntxp+nt
            ntacumu=ntacumu+nt
            ndexp=ndexp+numr
            if conj4==conj3:
                bandera=False
                print('''correcto''')
                audiresc=playsound(direc + '/rescorre.mp3')
                acutire=acutire+timecal+cal
                acuti=acuti+cal
                calre=cal+timecal
                tsumarepe=ntxp+calre
                tsumarepeacumu=tsumarepeacumu+tsumarepe
                ndexacumu=ndexacumu+ndexp
                ndexyrepe=repenum+ndexp
                ndexyrepeacumu=ndexyrepeacumu+ndexyrepe
            else:
                if len(conj4)==len(conj3):
                    print('''vuelve a intentar''')
                    audiresin=playsound(direc +
                    '/resincorre.mp3')
                    malacumu=malacumu+1
                    mal=mal+1
                conj4.clear()
                timecal=timecal+cal
                repenum=repenum+1
                print('''vuelve a intentar''')
        nt=ntxp
        repemunacu=repemunacu+repenum
        conjuntorespuesta.append(c)
        tiempoxpreguntasin.append(cal)
        tiemporepexpreguntacon.append(calre)
        tiemacumupoxpreguntasin.append(acuti)
        tiempoacumurepexpreguntacon.append(acutire)
        tiempoexploestrella.append(nt)
        tiempoexploestrellaacumulado.append(ntacumu)
        repesumaestrella.append(tsumarepe)
        repesumaestrellaacumulado.append(tsumarepeacumu)
        numrepesuma.append(repenum)
        numrepesumaacumu.append(repemunacu)
        numexploxpreg.append(ndexp)
        numexploacumu.append(ndexacumu)
        numsumaexploxpregunta.append(ndexyrepe)
        numsumaexploacumu.append(ndexyrepeacumu)
        incorrestas1.append(mal)
        incoacumuladas.append(malacumu)
    names=['Conjunto respuesta',
           'Tiempo final en contestar correctamente',
           'Tiempo final en contestar correctamente acumulado',
           'Tiempo sin exploración por pregunta',
           'Tiempo sin exploración acumulado',
           'Tiempo de exploración por pregunta',
           'Tiempo de exploración acumulado',
           'Tiempo con exploración por pregunta',
           'Tiempo con exploración acumulado',
           'Número de repeticiones por pregunta',
           'Número de repeticiones acumulado',
           'Número de exploraciones por pregunta',
           'Número de exploraciones acumulado',
           'Número de repeticiones más exploraciones por
           pregunta',
           'Número de repeticiones más exploraciones acumulado',
           'Errores por pregunta', 'Errores acumulados']
    df=pd.DataFrame(list(zip(conjuntorespuesta,
    tiempoxpreguntasin, tiemacumupoxpreguntasin,
    tiemporepexpreguntacon,
    tiempoacumurepexpreguntacon, tiempoexploestrella,
    tiempoexploestrellaacumulado,
    repesumaestrella,repesumaestrellaacumulado, numrepesuma,
    numrepesumaacumu,
    numexploxpreg, numexploacumu, numsumaexploxpregunta,
    numsumaexploacumu,
    incorrestas1, incoacumuladas)), columns=names)
    df.to_csv(direc + '/CC3/dat3_C_.csv')
    print('''Notamos que aunque calculemos el complemento de
    varios conjuntos,
    siempre debemos usar la misma regla para poder
calcularlos''')
    audiresin=playsound(direc + '/retro3.mp3')
    promedio=numpreg/2
    if mal==0:
        print('''increible no te has equivocado''')
        audiresin=playsound(direc + '/increible.mp3')
    elif mal<=promedio:
        print('''Buen trabajo''')
    elif promedio>mal>=promedio*3:
        print('''Te sugerimos volver a realizar el nivel''')
        audiresin=playsound(direc + '/repenivel.mp3')
    else:
        print('''Te sugerimos volver a estudiar la actividad
        didactica por el numero
        de respuestas incorrectas''')
        audiresin=playsound(direc + '/repeincorre.mp3')
    print(df)
########---Nivel
cuatro---##################################################
elif menu=="4":
    print('''nivel 4''')

    #nivel 4 conjuntos de 3 elementos
    audis15=playsound(direc + '/nivel4instru.mp3')
    print('''nivel 4''')
    print('''En este nivel sacaras el complemento de 2 conjuntos,
    cuando el conjunto
    universo tiene 3 conjuntos. Sin embargo, antes deberás sacar
la intersección o
    la unión. ya sea el caso. Por ejemplo. Si se te pregunta saca
el complemento del
    conjunto A intersección B, primero deberás hacer la
intersección y después
    calcular el complemento tomando en cuenta el tercer conjunto,
es decir,  el
    conjunto universo. En cambio, si se te pregunta el
complemento del conjunto A
    unión B, primero deberás sacar la unión y después el
complemento, tomando
    en cuanta el tercer conjunto. ''')
    ##################listas de las variables
    conjuntorespuesta=[]
    #1 conjunto respuesta
    tiempoxpreguntasin=[]
    #2 tiempo por pregunta sin repetición ni acumulado
    tiemacumupoxpreguntasin=[]
    #3 tiempo acumulado sin repetición ni acumulado
    tiemporepexpreguntacon=[]
    #4 tiempo por repeticiones por pregunta
    tiempoacumurepexpreguntacon=[]
    #5 tiempo por repeticiones acumulado
    tiempoexploestrella=[]
    #6 tiempo con exploración por pregunta
    tiempoexploestrellaacumulado=[]
    #7 tiempo acumulado con exploración
    repesumaestrella=[]
    #8 tiempo con repetición y exploración por pregunta
    repesumaestrellaacumulado=[]
    #9 tiempo acumulado con repetición y exploración
    numrepesuma=[]
    #10 número de repeticiones por pregunta
    numrepesumaacumu=[]
    #11 número de repeticiones acumulado
    numexploxpreg=[]
    #12 numero de exploraciones por pregunta
    numexploacumu=[]
    #13 numero de exploraciones acumulada
    numsumaexploxpregunta=[]
    #14 numero de suma de repetición y exploración por pregunta
    numsumaexploacumu=[]
    #15 numero de suma de repetición y exploración acumulado
    incorrestas1=[]
    #16 errores por pregunta
    incoacumuladas=[]
    #17 errores por pregunta acumulada
    mal=0
    acuti=0
    acutire=0
    repemunacu=0
    ntacumu=0
    tsumarepeacumu=0
    ndexacumu=0
    ndexyrepeacumu=0
    for k in range(numpreg):
        conj1=set()
        conj2=set()
        conj3=set()
        rellenacon(conj1,0,7,2)
        rellenacon(conj2,0,7,3)
        rellenacon(conj3,0,7,2)
        conj4=set()
        conj5=conj1|conj2|conj3
        bandera=True
        malacumu=0
        timecal=0
        repenum=0
        ntxp=0
        tsumarepe=0
        ndexp=0
        ndexyrepe=0
        abc=rd.randint(1,2)
        while bandera==True:
            print('''El conjunto universo esta compuesto por los
            conjuntos A,B,C,
            ''')
            if abc==1:
                print('''Calcula el complemento del conjunto A
                intersección B''')
```

```
                audis14=playsound(direc + '/zzz.mp3')
            if abc==2:
                print('''Calcula el complemento del conjunto A
                ↪  unión B''')
                audis14=playsound(direc + '/qqq.mp3')
            audis14=playsound(direc + '/conjA33.mp3')
            print('''conjunto A''',conj1)
            elemetosconj(conj1)
            print('''el conjunto b''')
            audis14=playsound(direc + '/conjB33.mp3')
            print(conj2)
            elemetosconj(conj2)
            print('''y el conjunto c''')
            audis14=playsound(direc + '/conjC33.mp3')
            print(conj3)
            elemetosconj(conj3)
            if abc==1:
                print('''Calcula el complemento de la
                ↪  intersección A con B''')

                conj4=conj1 & conj2
                conj7=compleleA(conj5,conj4)
            if abc==2:
                print('''Calcula el complemento de la union A con
                ↪  unión B''')

                conj4=conj1|conj2
                conj7=compleleA(conj5,conj4)

            audis11=playsound(direc + '/pide.mp3')
            conj6=set()
            a=time.time()
            c=conj7
            conj6,nt, numr =
            ↪  respconj3(conj6,len(conj7),conj1,conj2,conj3,conj7)
            b=time.time()
            cal=b-a-nt
            calre=0
            ntxp=ntxp+nt
            ntacumu=ntacumu+nt
            ndexp=ndexp+numr
            print('''respuest''', conj7)
            if conj6==conj7:
                bandera=False
                print('''correct''')
                audiresc=playsound(direc + '/rescorre.mp3')

                acutire=acutire+timecal+cal
                acuti=acuti+cal
                calre=cal+timecal
                tsumarepe=ntxp+calre
                tsumarepeacumu=tsumarepeacumu+tsumarepe
                ndexacumu=ndexacumu+ndexp
                ndexyrepe=repenum+ndexp
                ndexyrepeacumu=ndexyrepeacumu+ndexyrepe
            else:
                if len(conj6)==len(conj7):
                    print('''vuelve a intentar''')
                    audiresin=playsound(direc +
                    ↪  '/resincorre.mp3')
                    mal=mal+1
                    malacumu=malacumu+1
                conj6.clear()
                timecal=timecal+cal
                repenum=repenum+1

                print('''Se volvio a repetir''')

        nt=ntxp
        repemunacu=repemunacu+repenum
        conjuntorespuesta.append(c)
        tiempoxpreguntasin.append(cal)
        tiemporepexpreguntacon.append(calre)
        tiemacumupoxpreguntasin.append(acuti)
        tiempoacumurepexpreguntacon.append(acutire)
        tiempoexploestrella.append(nt)
        tiempoexploestrellaacumulado.append(ntacumu)
        repesumaestrella.append(tsumarepe)
        repesumaestrellaacumulado.append(tsumarepeacumu)
        numrepesuma.append(repenum)
        numrepesumaacumu.append(repemunacu)
        numexploxpreg.append(ndexp)
        numexploacumu.append(ndexacumu)
        numsumaexploxpregunta.append(ndexyrepe)
        numsumaexploacumu.append(ndexyrepeacumu)
        incorrestas1.append(mal)
        incoacumuladas.append(malacumu)

    names=['Conjunto respuesta',
           'Tiempo final en contestar correctamente',
           'Tiempo final en contestar correctamente acumulado',
           'Tiempo sin exploración por pregunta',
           'Tiempo sin exploración acumulado',
           'Tiempo de exploración por pregunta',
           'Tiempo de exploración acumulado',
           'Tiempo con exploración por pregunta',
           'Tiempo con exploración acumulado',
           'Número de repeticiones por pregunta',
           'Número de repeticiones acumulado',
           'Número de exploraciones por pregunta',
           'Número de exploraciones acumulado',
           'Número de repeticiones más exploraciones por
           ↪  pregunta',
           'Número de repeticiones más exploraciones acumulado',
           'Errores por pregunta', 'Errores acumulados']
    df=pd.DataFrame(list(zip(conjuntorespuesta,
    tiempoxpreguntasin, tiemacumupoxpreguntasin,
    ↪  tiemporepexpreguntacon,
    tiempoacumurepexpreguntacon, tiempoexploestrella,
    ↪  tiempoexploestrellaacumulado,
    repesumaestrella,repesumaestrellaacumulado, numrepesuma,
    ↪  numrepesumaacumu,
    numexploxpreg, numexploacumu, numsumaexploxpregunta,
    ↪  numsumaexploacumu,
    incorrestas1, incoacumuladas)), columns=names)
    df.to_csv(direc + '/CC4/dat4_C_.csv')
    print('''Notamos que aunque calculemos el complemento de
    ↪  varios conjuntos,
    siempre debemos usar la misma regla para poder
↪   calcularlos''')
    audiresin=playsound(direc + '/retro4.mp3')
    promedio=numpreg/2
    if mal==0:
        print('''increible no te has equivocado''')
        audiresin=playsound(direc + '/increible.mp3')
    elif mal<=promedio:
        print('''Buen trabajo''')
    elif promedio>mal>=promedio*3:
        print('''Te sugerimos volver a realizar el nivel''')
        audiresin=playsound(direc + '/repenivel.mp3')
    else:
        print('''Te sugerimos volver a estudiar la actividad
        ↪  didactica por el numero
        de respuestas incorrectas''')
        audiresin=playsound(direc + '/repeincorre.mp3')
    print(df)
########---Práctica---##############################################
elif menu=='5':
    ejemplo_instrucciones="A continuación haremos un ejercicio
    ↪  para que te
    familiarices con el programa, para ello calcularemos el
    ↪  complemento de un
    conjunto  "
    audiresc=playsound(direc + '/ejeminstru2.mp3')
    audiresc=playsound(direc + '/didaintrudos.mp3')
    audieje5au=playsound(direc + '/explorar.mp3')
    audieje5au=playsound(direc + '/explirepe.mp3')
    conjeje1=set()
    conjeje=set()
    #llenar conjuntos
    print('''Podrás ingresar tus respuestas después de este
    ↪  sonido, cada que
    ingreses un elemento da un enter, es decir, si el conjunto
↪   respuesta consta
    de los elementos 1,2,3, entonces agrega el 1, enter, 2 enter,
↪   3 y enter. si
    la respuesta es correcta se escuchara este sonido, y en caso
↪   contrario este
    otro sonido. ''')
    print('''para explorar los elementos de los conjuntos
    ↪  presiona *, la d para
    la derecha y la a para la izquierda, s para cambiar de
↪   conjunto, para salir -''')
    rellenacon(conjeje,0,9,4)
    e=rd.choice(list(conjeje))
    conjeje1.add(e)
    conjeje2=compleleA(conjeje,conjeje1)
    bandera=True
    while bandera==True:
        print(conjeje)
        # conpkemento con unión
        audiresc=playsound(direc + '/conjA.mp3')
        elemetosconj(conjeje)
        print(conjeje1)
        audiresc=playsound(direc + '/conjB.mp3')
        elemetosconj(conjeje1)
        audis11=playsound(direc + '/pide.mp3')
        conjres=set()
        conjres, nt, numr= respconj(conjres,
        ↪  len(conjeje2),conjeje,conjeje1,conjeje2)
        if conjeje2==conjres:
            bandera=False
            audis11=playsound(direc + '/rescorre.mp3')
            print('''Es correcta tu respuesta''')
        else:
            if len(conjres)==len(conjeje2):
                audis11=playsound(direc + '/resincorre.mp3')
                print('''vuelve a intentarlo''')

            conjres.clear()
```

```
########---Salida de programa---#####################################
    elif menu=="6":
        audis14=playsound(direc + '/adios.mp3')
        print('''Has concluido con éxito las actividades, recuerda que el
        conjunto complemento de A es todo lo que no es A dentro del conjunto
        universo. Felicidades''')
        break


########---Opción no valida---########################################
    else:
        print('''Tu opción no es valida, vuelve a ingresar la opción del menú,
        cero para la actividad didáctica, ...''')
        audieje5au=playsound(direc + '/novalida.mp3')
        menu="7"
########---Fin del programa del complemento ---#######################
```

Anexo B

Códigos para analizar las bases de datos

Leer y ordenar datos:

```
#librerias
library(sos)
library(readr)
options(readr.show_col_types = FALSE)
library(dplyr)
library(plyr)
library(tables)
library(ggplot2)
library(patchwork)
library(gridExtra)
library(data.table)
library(reshape)
library(multcomp)
library(pastecs)
library(cowplot)


cvss<-dir()
diredatos<-getwd()
directorio<-getwd()

tmp <- list()
for(i in 1:4){
  Namefiles = list.files(directorio, pattern =
  ↪  paste("dat",i,"_U_",sep = ''))
 for (j in 1:length(Namefiles)) tmp[[j+length(Namefiles)*(i-1)]] <-
 ↪  fread(Namefiles[j])
  datosu <- rbindlist(tmp)
}
colnames(datosu) <- c('a','b','c','d','e','f','g','h','ii','j',
'k','l','m','n','o','p','r','q')
op<-rep("union", length(datosu$a));
↪  datosu<-cbind.data.frame(datosu,op);
nivel<-c(rep("N1", 180),rep("N2", 180), rep("N3", 180), rep("N4",
↪  180))
datosu<-cbind.data.frame(datosu,nivel)
write.csv(datosu,file = "uniondatosniveles.csv")


datosun1<-filter(datosu,nivel %in% c("N1"));
↪  write.csv(datosun1,file = "uniondatos1niveles.csv");
datosun2<-filter(datosu,nivel %in% c("N2"));  write.csv(datosun2,file
↪  = "uniondatos2niveles.csv");
datosun3<-filter(datosu,nivel %in% c("N3"));  write.csv(datosun3,file
↪  = "uniondatos3niveles.csv");
datosun4<-filter(datosu,nivel %in% c("N4"));  write.csv(datosun4,file
↪  = "uniondatos4niveles.csv");

tmp <- list()
for(i in 1:4){
  Namefiles = list.files(directorio, pattern =
  ↪  paste("dat",i,"_I_",sep = ''))
  for (j in 1:length(Namefiles)) tmp[[j+length(Namefiles)*(i-1)]] <-
  ↪  fread(Namefiles[j])
  datosi <- rbindlist(tmp)
}
colnames(datosi) <- c('a','b','c','d','e','f','g','h','ii','j','k',
'l','m','n','o','p','r','q')
op<-rep("interseccion", length(datosi$a));
↪  datosi<-cbind.data.frame(datosi,op);
nivel<-c(rep("N1", 180),rep("N2", 180), rep("N3", 180), rep("N4",
↪  180))
datosi<-cbind.data.frame(datosi,nivel)
write.csv(datosi,file = "interseccion datosniveles.csv")

datosin1<-filter(datosi,nivel %in% c("N1"));
↪  write.csv(datosun1,file = "interdatos1niveles.csv");
datosin2<-filter(datosi,nivel %in% c("N2")); write.csv(datosun2,file
↪  = "interdatos2niveles.csv");
datosin3<-filter(datosi,nivel %in% c("N3")); write.csv(datosun3,file
↪  = "interdatos3niveles.csv");
datosin4<-filter(datosi,nivel %in% c("N4")); write.csv(datosun4,file
↪  = "interdatos4niveles.csv");
```

Linea de tiempo

```
datostodosm7<-datostodos[-c(1931,2453,2078,2451,2,93,2709,2461),]
p<-ggplot(datostodosm7, aes(ii,numero, colour=nivel))+geom_point()+
geom_hline(yintercept = 2,
linetype = "longdash")+geom_hline(yintercept = 3,linetype =
↪  "longdash")+
geom_hline(yintercept = 4,
linetype = "longdash")
p1<-p+geom_segment(x=mean(datostodosm7$ii[1:180]),y=1,
xend=mean(datostodosm7$ii[1:180]),yend=2,
linetype = "dashed",color="red")
+geom_segment(x=mean(datostodosm7$ii[181:360]),y=1,
xend=mean(datostodosm7$ii[181:360]),yend=2,
linetype = "dashed",color="forestgreen")
p2<-p1+geom_segment(x=mean(datostodosm7$ii[361:540]),
y=1,xend=mean(datostodosm7$ii[361:540]),
yend=2,linetype = "dashed",color="turquoise2")
+geom_segment(x=mean(datostodosm7$ii[541:720]),y=1,
xend=mean(datostodosm7$ii[541:720]),yend=2,
linetype = "dashed",color="purple4")
p3<-p2+geom_segment(x=mean(datostodosm7$ii[721:900]),
y=2,xend=mean(datostodosm7$ii[721:900]),
yend=3,linetype = "dashed",color="red")
+geom_segment(x=mean(datostodosm7$ii[901:1080]),y=2,
xend=mean(datostodosm7$ii[901:1080]),
yend=3,linetype = "dashed",color="forestgreen")
p4<-p3+geom_segment(x=mean(datostodosm7$ii[1081:1260]),
y=2,xend=mean(datostodosm7$ii[1081:1260]),
yend=3,linetype = "dashed",color="turquoise2")
+geom_segment(x=mean(datostodosm7$ii[1261:1440]),
y=2,xend=mean(datostodosm7$ii[1261:1440]),yend=3,
linetype = "dashed",color="purple4")
p5<-p4+geom_segment(x=mean(datostodosm7$ii[1441:1620]),
y=3,xend=mean(datostodosm7$ii[1441:1620]),
yend=4,linetype = "dashed",color="red")
+geom_segment(x=mean(datostodosm7$ii[1621:1800]),
y=3,xend=mean(datostodosm7$ii[1621:1800]),yend=4,
linetype = "dashed",color="forestgreen")
p6<-p5+geom_segment(x=mean(datostodosm7$ii[1801:1980]),
y=3,xend=mean(datostodosm7$ii[1801:1980]),yend=4,
linetype = "dashed",color="turquoise2")
+geom_segment(x=mean(datostodosm7$ii[1981:2160]),y=3,
xend=mean(datostodosm7$ii[1981:2160]),yend=4,
linetype = "dashed",color="purple4")
p7<-p6+geom_segment(x=mean(datostodosm7$ii[2161:2340]),
y=4,xend=mean(datostodosm7$ii[2161:2340]),
yend=5,linetype = "dashed",color="red")
+geom_segment(x=mean(datostodosm7$ii[2341:2520]),y=4,
xend=mean(datostodosm7$ii[2341:2520]),yend=5,
linetype = "dashed",color="forestgreen")
p8<-p7+geom_segment(x=mean(datostodosm7$ii[2521:2700]),
y=4,xend=mean(datostodosm7$ii[2521:2700]),yend=5,
linetype = "dashed",color="turquoise2")
```

```
+geom_segment(x=mean(datostodosm7$ii[2701:2872]),
y=4,xend=mean(datostodosm7$ii[2701:2872]),yend=5,
linetype = "dashed",color="purple4")
p9<- p8+labs(title="Linea del tiempo de cada
operación,\n  con sus respectivos promedios",
x ="tiempo en segundos", y = "Operaciones")
```

```
p10<-p9+annotate('text',label="Union", x=800, y=1.5)
 +annotate('text',label="Intersección",
  x=800, y=2.5)+annotate('text',label="Diferencia",
x=800, y=3.5)+annotate('text',label="Complemento",
x=800, y=4.5)
```

Graficos de caja

```
eruu<-ggplot(filter(datosu, q>0), aes(x=nivel, colour=q  ))
↪  +geom_boxplot(aes(x=nivel, y=q,colour=nivel), lwd=1,
↪  outlier.colour="red",
outlier.fill="red",notch = TRUE) +labs(title="Errores de la unión por
↪  niveles", x ="  ", y = "Número de errores por pregunta")+
 scale_y_continuous(breaks = seq(0,40,5))
eru0u<-eruu+geom_jitter(aes(x=nivel, y=q),alpha = .5, colour="blue",
↪  cex=.4)+geom_vline(xintercept=1.5:3.5,linetype="twodash")+
geom_segment(x=0, y=mean(filter(datosu, q>0, nivel=="N1")$q),
↪  xend=1.5, yend=mean(filter(datosu, q>0, nivel=="N1")$q),
↪  colour="red")+
geom_segment(x=1.5, y=mean(filter(datosu, q>0, nivel=="N2")$q),
↪  xend=2.5, yend=mean(filter(datosu, q>0, nivel=="N2")$q),
↪  colour="red")
```

```
eru1u<-eru0u+geom_segment(x=2.5, y=mean(filter(datosu, q>0,
↪  nivel=="N3")$q), xend=3.5, yend=mean(filter(datosu, q>0,
↪  nivel=="N3")$q),
 colour="red")+geom_segment(x=3.5, y=mean(filter(datosu, q>0,
↪  nivel=="N4")$q), xend=4.5, yend=mean(filter(datosu, q>0,
↪  nivel=="N4")$q),
 colour="red")
eru2u<-eru1u+ theme(legend.key.size = unit(1.5, units =
↪  "cm"),axis.text.x=element_text(size = 15),
axis.title.x = element_text(size = 20),axis.title.y =
↪  element_text(size = 15),title = element_text(size = 17),
↪  legend.position='none' )
#+ scale_color_discrete('Niveles')
ggsave("eru2u.pdf", width = 30,  height = 20,  units = "cm")
```

Herramienta para hacer el análisis de varianza

```
#Este código es una actividad didáctica para la enseñanza y la
↪  evaluación de la diferencia
entre dos conjuntos.
import time
from datetime import datetime
import pandas as pd
import numpy as np
from playsound import playsound
from tkinter import Tk, Label, Button
import random as rd
import os
ruta = os.getcwd()


#Funciones:
def LLENARCONJUNTOS(conj, a, b): #Función para llenar los conjuntos.
  for m in range(a):
    x=rd.randint(0,b)
    conj.add(str(x))

def DICTACONJUNTO (conj):
  for i in conj: #Mencionamos los elementos del conjunto
    if i=='0':
      sound0 = playsound(ruta+'/0.mp3')
    if i=='1':
      sound1 = playsound(ruta+'/1.mp3')
    if i=='2':
      sound2 = playsound(ruta+'/2.mp3')
    if i=='3':
      sound3 = playsound(ruta+'/3.mp3')
    if i=='4':
      sound4 = playsound(ruta+'/4.mp3')
    if i=='5':
      sound5 = playsound(ruta+'/5.mp3')
    if i=='6':
      sound6 = playsound(ruta+'/6.mp3')
    if i=='7':
      sound7 = playsound(ruta+'/7.mp3')
    if i=='8':
      sound8 = playsound(ruta+'/8.mp3')
    if i=='9':
      sound9 = playsound(ruta+'/9.mp3')

def DICTANUMERO (n):
  ConjRetro=set()
  ConjRetro.add(n)
  DICTACONJUNTO(ConjRetro)

def EXPLORACONJUNTOS (listA, listB, n):
  mov='a'
  i=0
  if (n==1):
    print('Estás en el Conjunto A')
    soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')

  elif (n==2):
    print('Estás en el Conjunto B')
    soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')

  while(mov!='-'):
    print(listA[i])
    DICTANUMERO(listA[i])
    mov=input('Movimiento:')
    if (mov=='d' or mov=='D'):
      if (i<(len(listA)-1)):
        i+=1
      else:
        i=0
    elif (mov=='a' or mov=='A'):
      if (i==0):
        i=len(listA)-1
      else:
        i-=1
    elif (mov=='s' or mov=='S'):
      if (n==1):
        EXPLORACONJUNTOS (listB, listA, 2)
        break
      else:
        EXPLORACONJUNTOS (listB, listA, 1)
        break

def EXPLORACONJUNTOS2 (listA, listB, listC, n):
  mov='a'
  i=0
  if (n==1):
    print('Estás en el Conjunto A')
    soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')

  elif (n==2):
    print('Estás en el Conjunto B')
    soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')

  elif (n==3):
    print('Estás en el Conjunto C')
    soundconjuntoC = playsound(ruta+'/ConjuntoC.mp3')

  while(mov!='-'):
    print(listA[i])
    DICTANUMERO(listA[i])
    mov=input('Movimiento:')
    if (mov=='d' or mov=='D'):
      if (i<(len(listA)-1)):
        i+=1
      else:
        i=0
    elif (mov=='a' or mov=='A'):
      if (i==0):
        i=len(listA)-1
      else:
        i-=1
    elif (mov=='s' or mov=='S'):
      if (n==1):
        EXPLORACONJUNTOS2 (listB, listC, listA, 2)
        break
      elif (n==2):
        EXPLORACONJUNTOS2 (listB, listC, listA, 3)
        break
      else:
        EXPLORACONJUNTOS2 (listB, listC, listA, 1)
        break

#Variables globales
```

```
npreg0=1
npreg1=10
npreg2=10
npreg3=10
npreg4=10


#Bienvenida al programa (Audios):
soundbienvenida = playsound(ruta+'/Bienvenida.mp3')
print('''Bienvenido al programa de actividades didácticas para la
↪  enseñanza y evaluación de
la diferencia de dos conjuntos.''')
sounddefinicionformal = playsound(ruta+'/DefinicionFormal.mp3')
print('''Definición formal: La diferencia de dos conjuntos es una
↪  operación que da como resultado
otro conjunto con los elementos del primer conjunto sin los elementos
↪  del segundo conjunto.''')
soundexplicacion = playsound(ruta+'/Explicacion.mp3')
print('''Este programa cuenta con 5 actividades diferentes, una de
↪  ellas tiene la finalidad de
enseñarte, y las otras cuatro son evaluaciones que van aumentando su
↪  dificultad e incluyendo
cada vez más conceptos.''')
soundenter = playsound(ruta+'/Enter.mp3')
print('''Para interactuar con este programa deberás teclear Enter
↪  después de cada número.''')


#Menú de actividades:
act='0'
while (act!='6'):
  soundmenu = playsound(ruta+'/Menu.mp3')
  print('''Presiona cero para la actividad de aprendizaje, uno para
  ↪  la evaluación en su nivel
  más básico, dos para la evaluación del nivel básico incluyendo
↪  conjuntos vacíos, tres
  para la evaluación de la conmutatividad, cuatro para la evaluación
↪  con tres conjuntos,
  cinco para realizar ejercicios de práctica, seis para salir.''')
  act=input('Opción Menú:')
  if act=='0':
    print('Actividad Didáctica')
    sounddidactica1 = playsound(ruta+'/Didactica1.mp3')
    print('''Haz ingresado a la actividad didáctica de la diferencia.
    ↪  En esta parte te
    enseñaremos cómo funciona, te daremos algunos ejemplos, y te
↪  guiaremos a que realices
    algunos ejercicios similares a las actividades de evaluación.''')
    soundmenudidac = playsound(ruta+'/Didactica1_1.mp3')
    print('''Si quieres estudiar la actividad didáctica de manera
    ↪  fluida presiona 'a' seguido
    de enter, o si prefieres poder moverte entre sus diferentes
↪  secciones presiona 'd' y enter.''')
    menudidac=input('Fluidez didáctica:')
    didac=''
    while(didac=='' or didac=='1' or didac=='2' or didac=='3' or
    ↪  didac=='4'):
      if (menudidac=='d'):
        print ('Avance por partes')
        soundrep1 = playsound(ruta+'/Rep1.mp3')
        print('''Presiona uno para la definición informal, dos para
        ↪  los ejemplos en la vida
        cotidiana, tres para las instrucciones de los ejercicios, o
↪  cuatro para los ejercicios
        prácticos. Cualquier otro número para salir.''')
        didac=input('Opción menú didáctica:')
      else:
        print ('Avance fluido')
      if(didac=='' or didac=='1'):
        print ('Definición Informal')
        sounddidactica2 = playsound(ruta+'/Didactica2.mp3')
        print('''La diferencia entre conjuntos es una operación que
        ↪  'elimina' del primer
        conjunto los elementos que aparecen en el segundo
↪  conjunto.''')
      if(didac=='' or didac=='2'):
        print ('Ejemplos')
        ejemp='0'
        if (menudidac=='d'):
          soundselecejem = playsound(ruta+'/Didactica3_1.mp3')
          print('''Hay cinco ejemplos, presiona el número de ejemplo
          ↪  que quieres escuchar
          seguido de un enter o cero si quieres escucharlos
↪  todos.''')
          ejemp=input('Número de ejemplo:')
        if (ejemp=='0' or ejemp=='1'):
          print ('Ejemplo 1')
          sounddidactica3 = playsound(ruta+'/Didactica3.mp3')
          print('''Ejemplo 1: El conjunto A tendrá la propiedad que
          ↪  serán todas las personas
          a las que les gusta lo dulce, y el conjunto B tendrá la
↪  propiedad que serán todas
          las personas a las que les gusta lo salado. Como
↪  recordarás, tenemos que quitar del
          conjunto A a todos los elementos que también aparezcan en
↪  el conjunto B, es decir,
          del conjunto A que son todas las personas a las que les
↪  gusta lo dulce, vamos a quitar
          a todas las personas que también les gusta lo salado, y
↪  nuestro conjunto diferencia
          son todas las personas a las que les gusta lo dulce, pero
↪  no les gusta lo salado.''')
        if (ejemp=='0' or ejemp=='2'):
          print ('Ejemplo 2')
          sounddidactica4 = playsound(ruta+'/Didactica4.mp3')
          print('''Ejemplo 2: El conjunto A tendrá la propiedad que
          ↪  serán todas las personas
          a las que les gusta el rock, y el conjunto B tendrá la
↪  propiedad que serán todas
          las personas a las que les gustan los boleros. Al igual que
↪  en el ejemplo anterior,
          del conjunto A que son todas las personas a las que les
↪  gusta el rock, vamos a quitar
          a todas las personas que también les gustan los boleros,
↪  porque estos están en el
          conjunto B. Quedándonos nuestro conjunto diferencia como
↪  todas las personas que les
          gusta el rock, pero que no les gustan los boleros.''')
        if (ejemp=='0' or ejemp=='3'):
          print ('Ejemplo 3')
          sounddidactica5 = playsound(ruta+'/Didactica5.mp3')
          print('''Ejemplo 3: Ahora revisemos un caso donde la
          ↪  diferencia sea vacía. Es decir,
          donde todos los elementos del conjunto A también están en
↪  el conjunto B. Pongamos que
          el conjunto A tendrá la propiedad que serán todas las
↪  personas que saben Braille,
          y el conjunto B tendrá la propiedad que serán todas las
↪  personas que saben leer
          o escribir. Como podrás notar, todas las personas que saben
↪  Braille saben leer o
          escribir, por lo que, si al conjunto A que son todas las
↪  personas que saben Braille,
          le quitamos todas las personas que saben leer o escribir,
↪  la diferencia es vacía.''')
        if (ejemp=='0' or ejemp=='4'):
          print ('Ejemplo 4')
          sounddidactica6 = playsound(ruta+'/Didactica6.mp3')
          print('''Ejemplo 4: Aquí vamos a observar que la operación
          ↪  diferencia no es
          conmutativa, esto quiere decir que, sí importa si hacemos
↪  la operación diferencia
          del conjunto A con el conjunto B, o si la hacemos al revés,
↪  que hagamos la diferencia
          del conjunto B con el conjunto A, porque nos puede dar un
↪  resultado completamente
          diferente. Veamos: Si el conjunto A tiene la propiedad que
↪  son todas las personas
          que tienen más de 25 años, y el conjunto B tiene la
↪  propiedad que son todas las
          personas que tienen hijos, el conjunto diferencia del
↪  conjunto A con el conjunto B,
          son todas las personas que tienen más de 25 años y que no
↪  tienen hijos; pero si
          hacemos la diferencia en diferente orden, es decir, la
↪  diferencia del conjunto B con
          el conjunto A, nos daremos cuenta que no es lo mismo, ahora
↪  la respuesta son todas
          las personas que tienen hijos, pero que no tienen más de 25
↪  años.''')
        if (ejemp=='0' or ejemp=='5'):
          print ('Ejemplo 5')
          sounddidactica7 = playsound(ruta+'/Didactica7.mp3')
          print('''Ejemplo 5: Veamos un caso en el que haremos la
          ↪  diferencia de tres conjuntos:
          Pongamos el conjunto A con la propiedad de que son todas
↪  las personas que trabajan, el
          conjunto B con la propiedad de que son todas las personas
↪  que estudian la universidad,
          y el conjunto C con la propiedad que son todas las personas
↪  que hacen ejercicio, y
          el resultado serán todas las personas que cumplen la
↪  primera propiedad pero no las
          otras dos, es decir, que son todas las personas que
↪  trabajan, pero que no estudian
          la universidad, y que no hacen ejercicio.''')
      if(didac=='' or didac=='3'):
        print ('Instrucciones')
        sounddidactica8 = playsound(ruta+'/Didactica8.mp3')
        print('''A continuación te guiaremos a realizar algunos
        ↪  ejercicios similares a las
        actividades de evaluación, dándote las respuestas, para que
↪  te vayas familiarizando. Cabe
        aclarar que para las actividades de evaluación solo
↪  utilizarás las teclas del 0 al
        9 y el Enter. Si respondes incorrectamente, se te volverá a
↪  hacer la misma pregunta
        hasta que respondas correctamente.''')
        sounddidactica9 = playsound(ruta+'/Didactica9.mp3')
        print('''Instrucciones para los ejercicios: Primero te vamos
        ↪  a mencionar los conjuntos A
```

```
        y B con sus elementos, porque deberás quitar los elementos
↪  del conjunto A que aparezcan
        en el conjunto B, y los que queden serán la respuesta,
↪  después de eso escucharás el
        siguiente sonido para indicarte que ya puedes responder:''')
        soundpideresp = playsound(ruta+'/PideResp.mp3')
        soundsirespuestacorrecta =
        ↪  playsound(ruta+'/SiRespuestaCorrecta.mp3')
        print('''Si tu respuesta es correcta se escuchará:''')
        soundrespuestacorrecta =
        ↪  playsound(ruta+'/RespuestaCorrecta.mp3')
        soundsirespuestaincorrecta =
        ↪  playsound(ruta+'/SiRespuestaIncorrecta.mp3')
        print('''Y si tu respuesta es incorrecta se escuchará:''')
        soundrespuestaincorrecta =
        ↪  playsound(ruta+'/RespuestaIncorrecta.mp3')
        sounddidactica10 = playsound(ruta+'/Didactica10.mp3')
        print('''Los números de tu respuesta debererás ingresarlos
        ↪  uno a uno seguido de la
        tecla Enter, y en los casos que la respuesta es vacía, solo
↪  presiona Enter. Te sugerimos
        ayudarte con los dedos de las manos si se te dificulta
↪  memorizar. En esta actividad por
        ser de aprendizaje te daremos la respuesta, pero en las
↪  evaluaciones ya no será así.''')
        sounddidactica11 = playsound(ruta+'/Didactica11.mp3')
        print('''Si no alzanzaste a escuchar o simplemente quieres
        ↪  que te volvamos a repetir el
        ejercicio antes de ingresar tu respuesta, presiona la tecla
↪  de suma y después Enter.''')
        sounddidactica12 = playsound(ruta+'/Didactica12.mp3')
        print('''Si quieres entrar a explorar los elementos de los
        ↪  conjuntos mientras estás
        respondiendo porque se te olvidó o simplemente no escuchaste
↪  bien, presiona la tecla
        de asterisco, lo que te llevará al primer elemento del primer
↪  conjunto. Podrás moverte
        por el conjunto con las teclas a y d, y si quieres cambiar de
↪  conjunto, presiona s. Para
        salir de la exploración y continuar con tu respuesta,
↪  presiona la tecla menos.''')
      if(didac=='' or didac=='4'):
        i=0
        while i<npreg0: #Con esta instrucción se creará un bucle
        ↪  dependiendo que cuantas
        preguntas se requieran en la actividad.
          conjA=set() #Creamos un primer conjunto de números
          ↪  aleatorios.
          conjB=set() #Creamos un segundo conjunto de números
          ↪  aleatorios.
          LLENARCONJUNTOS(conjA,3,4)
          LLENARCONJUNTOS(conjB,3,4)
          conjC=set()
          conjC=conjA-conjB #Creamos el conjunto Diferencia
          ↪  (Respuesta)
          if len(conjC)!=0:
            i+=1
            print('  EJERCICIO DE PRÁCTICA', i)
            print('Conjunto A:', conjA)
            print('Conjunto B:', conjB)
            print('Conjunto Diferencia (Respuesta):', conjC)
            incorrecto=True
            while incorrecto: #Ponemos un while para que se repita la
            ↪  pregunta hasta que sea
            contestada correctamente
              soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')
              DICTACONJUNTO(conjA)
              soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')
              DICTACONJUNTO(conjB)
              conjD=set()
              soundconjuntoresp = playsound(ruta+'/ConjuntoResp.mp3')
              print('''La respuesta es:''')
              DICTACONJUNTO(conjC)
              print('Escribe tu respuesta:')
              soundpideresp = playsound(ruta+'/PideResp.mp3')
              repetir=False
              j=0
              while (j<(len(conjC))): #Recibimos el conjunto
              ↪  respuesta del teclado.
                j+=1
                resp=input()
                if (resp=='+'):
                  repetir=True
                  break
                if (resp==''):
                  break
                if (resp=='*'):
                  listA=list(conjA)
                  listB=list(conjB)
                  EXPLORACONJUNTOS (listA, listB, 1)
                  print('Continua tu respuesta:')
                  soundpideresp = playsound(ruta+'/PideResp.mp3')
                  j-=1
                else:
                  conjD.add(resp)
              if conjC==conjD: #Revisamos si el usuario respondió
              ↪  adecuadamente.
                incorrecto=False
                print("      ¡RESPUESTA CORRECTA!\n")
                soundrespuestacorrecta =
                ↪  playsound(ruta+'/RespuestaCorrecta.mp3')
              elif repetir==True:
                soundrepetir = playsound(ruta+'/Repetir.mp3')
                print('''Repitamos el ejercicio.''')
              else:
                print("      ¡RESPUESTA INCORRECTA!\n")
                soundrespuestaincorrecta =
                ↪  playsound(ruta+'/RespuestaIncorrecta.mp3')
      if (menudidac=='a'):
        menudidac='d'
    soundfelicitacionesdidac =
    ↪  playsound(ruta+'/FelicitacionesDidac.mp3')
    print('''Felicitaciones. Haz terminado la actividad para el
    ↪  aprendizaje de la diferencia,
    y tú puedes decidir si estás listo para pasar a las actividades
↪  de evaluación o vuelves
    a repetir la actividad de aprendizaje.''')


  elif act=='1':
    Fila_Datos=[]
    Filas_Datos=[]
    timeFA=0
    timeSEA=0
    timeEA=0
    timeCEA=0
    nRA=0
    nEA=0
    nREA=0
    errA=0
    soundevaluacion1 = playsound(ruta+'/Evaluacion1.mp3')
    print('''Haz ingresado a la 'Evaluación 1'. En esta parte se
    ↪  evaluará la diferencia de
    dos conjuntos en su nivel más básico.''')
    i=0
    while i<npreg1: #Con esta instrucción se creará un bucle
    ↪  dependiendo que cuantas preguntas
    se requieran en la actividad.
      conjA=set() #Creamos un primer conjunto de números aleatorios.
      conjB=set() #Creamos un segundo conjunto de números
      ↪  aleatorios.
      LLENARCONJUNTOS(conjA,4,5)
      LLENARCONJUNTOS(conjB,4,5)
      conjC=set()
      conjC=conjA-conjB #Creamos el conjunto Diferencia (Respuesta)
      if len(conjC)!=0:
        i+=1
        print('  PREGUNTA', i)
        print('Conjunto A:', conjA)
        print('Conjunto B:', conjB)
        print('Conjunto Diferencia (Respuesta):', conjC)
        incorrecto=True
        repetir=False
        time1SE=time.time()
        timeE=0
        nR=0
        nE=0
        err=0
        while incorrecto: #Ponemos un while para que se repita la
        ↪  pregunta hasta que sea
        contestada correctamente
          time1R=time.time()
          soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')
          DICTACONJUNTO(conjA)
          soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')
          DICTACONJUNTO(conjB)
          conjD=set()
          print('Escribe tu respuesta:')
          soundpideresp = playsound(ruta+'/PideResp.mp3')
          repetir=False
          time1F=time.time()
          time2R=time.time()
          time1SE+=time2R-time1R
          j=0
          while (j<(len(conjC))): #Recibimos el conjunto respuesta
          ↪  del teclado.
            j+=1
            resp=input()
            if (resp=='+'):
              repetir=True
              break
            if (resp=='*'):
              listA=list(conjA)
              listB=list(conjB)
              time1E=time.time()
              EXPLORACONJUNTOS (listA, listB, 1)
              time2E=time.time()
              time1F+=time2E-time1E
              time1SE+=time2E-time1E
              timeE+=time2E-time1E
              nE+=1
              print('Continua tu respuesta:')
```

```
            soundpideresp = playsound(ruta+'/PideResp.mp3')
            j-=1
          else:
            conjD.add(resp)
        if conjC==conjD: #Revisamos si el usuario respondió
        ↪  adecuadamente.
          timeFF=time.time()
          incorrecto=False
          print("      ¡RESPUESTA CORRECTA!\n")
          soundrespuestacorrecta =
          ↪  playsound(ruta+'/RespuestaCorrecta.mp3')
        elif repetir==True:
          soundrepetir = playsound(ruta+'/Repetir.mp3')
          print('''Repitamos el ejercicio.''')
          nR+=1
        else:
          print("      ¡RESPUESTA INCORRECTA!\n")
          soundrespuestacorrecta =
          ↪  playsound(ruta+'/RespuestaIncorrecta.mp3')
          err+=1
      timeFA+=timeFF-time1F
      timeSEA+=timeFF-time1SE
      timeEA+=timeE
      timeCEA+=(timeFF-time1SE)+timeE
      nRA+=nR
      nEA+=nE
      nREA+=nR+nE
      errA+=err
      Fila_Datos=[conjC, timeFF-time1F, timeFA, timeFF-time1SE,
      ↪  timeSEA, timeE, timeEA,
      (timeFF-time1SE)+timeE, timeCEA, nR, nRA, nE, nEA, nR+nE,
      ↪  nREA, err, errA]
      Filas_Datos.append(Fila_Datos)
  soundretroa1 = playsound(ruta+'/Retroa1.mp3')
  print('''En esta actividad se realizaron''')
  DICTANUMERO (str(npreg1))
  soundretroa2 = playsound(ruta+'/Retroa2.mp3')
  print('''preguntas, y tus errores fueron''')
  DICTANUMERO (str(errA))
  if (errA<(npreg1/2)):
    soundfelicitaciones1 = playsound(ruta+'/Felicitaciones1.mp3')
    print('''Felicitaciones. Haz concluido la 'Evaluación 1'.''')
    soundsiguienteactividad =
    ↪  playsound(ruta+'/SiguienteActividad.mp3')
    print('''Puedes pasar a la siguiente actividad.''')
  else:
    soundintentanuevamente =
    ↪  playsound(ruta+'/IntentaNuevamente.mp3')
    print('''Por la cantidad de respuestas incorrectas que tuviste
    ↪  en esta actividad te
    sugerimos volver a realizarla antes de pasar a la
↪  siguiente.''')
  df_dat1=pd.DataFrame(Filas_Datos, columns=['Conjunto respuesta',
  ↪  'Tiempo final en contestar
  correctamente', 'Tiempo final en contestar correctamente
↪  acumulado', 'Tiempo sin exploración
  por pregunta', 'Tiempo sin exploración acumulado', 'Tiempo de
↪  exploración por pregunta',
  'Tiempo de exploración acumulado', 'Tiempo con exploración por
  ↪  pregunta', 'Tiempo con
  exploración acumulado', 'Número de repeticiones por pregunta',
↪  'Número de repeticiones
  acumulado', 'Número de exploraciones por pregunta', 'Número de
↪  exploraciones acumulado',
  'Número de repeticiones más exploraciones por pregunta', 'Número
  ↪  de repeticiones más
  exploraciones acumulado', 'Errores por pregunta', 'Errores
↪  acumulados'])
  df_dat1
  df_dat1.to_csv(ruta+'/CD1/dat1_D_.csv')


elif act=='2':
  Fila_Datos=[]
  Filas_Datos=[]
  timeFA=0
  timeSEA=0
  timeEA=0
  timeCEA=0
  nRA=0
  nEA=0
  nREA=0
  errA=0
  soundevaluacion2 = playsound(ruta+'/Evaluacion2.mp3')
  print('''Haz ingresado a la 'Evaluación 2'. En esta parte se
  ↪  evaluará la diferencia de
  dos conjuntos con la posibilidad de que haya conjuntos
↪  vacíos.''')
  i=0
  while i<npreg2: #Con esta instrucción se creará un bucle
  ↪  dependiendo que cuantas preguntas
  se requieran en la actividad.
    conjA=set() #Creamos un primer conjunto de números aleatorios.
    conjB=set() #Creamos un segundo conjunto de números
    ↪  aleatorios.
    LLENARCONJUNTOS(conjA,5,6)
    LLENARCONJUNTOS(conjB,4,6)
    conjC=set()
    conjC=conjA-conjB #Creamos el conjunto diferencia (Respuesta)
    i+=1
    print('   PREGUNTA', i)
    print('Conjunto A:', conjA)
    print('Conjunto B:', conjB)
    print('Conjunto Diferencia (Respuesta):', conjC)
    incorrecto=True
    repetir=False
    time1SE=time.time()
    timeE=0
    nR=0
    nE=0
    err=0
    while incorrecto: #Ponemos un while para que se repita la
  ↪  pregunta hasta que sea contestada
    correctamente
      time1R=time.time()
      soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')
      DICTACONJUNTO(conjA)
      soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')
      DICTACONJUNTO(conjB)
      conjD=set()
      print('Escribe tu respuesta:')
      soundpideresp = playsound(ruta+'/PideResp.mp3')
      repetir=False
      time1F=time.time()
      time2R=time.time()
      time1SE+=time2R-time1R
      if (len(conjC)==0):
        resp=input()
        if (resp=='+'):
          repetir=True
        elif (resp==''):
          conjD=set()
        elif (resp=='*'):
          listA=list(conjA)
          listB=list(conjB)
          time1E=time.time()
          EXPLORACONJUNTOS (listA, listB, 1)
          time2E=time.time()
          time1F+=time2E-time1E
          time1SE+=time2E-time1E
          timeE+=time2E-time1E
          nE+=1
          print('Continua tu respuesta:')
          soundpideresp = playsound(ruta+'/PideResp.mp3')
          resp=input()
          if (resp==''):
            conjD=set()
          else:
            conjD.add(resp)
        else:
          conjD.add(resp)
      else:
        j=0
        while (j<(len(conjC))): #Recibimos el conjunto respuesta
        ↪  del teclado.
          j+=1
          resp=input()
          if (resp=='+'):
            repetir=True
            break
          if (resp=='*'):
            listA=list(conjA)
            listB=list(conjB)
            time1E=time.time()
            EXPLORACONJUNTOS (listA, listB, 1)
            time2E=time.time()
            time1F+=time2E-time1E
            time1SE+=time2E-time1E
            timeE+=time2E-time1E
            nE+=1
            print('Continua tu respuesta:')
            soundpideresp = playsound(ruta+'/PideResp.mp3')
            j-=1
          else:
            conjD.add(resp)
      if (conjC==conjD and repetir==False): #Revisamos si el
      ↪  usuario respondió adecuadamente.
        timeFF=time.time()
        incorrecto=False
        print("      ¡RESPUESTA CORRECTA!\n")
        soundrespuestacorrecta =
        ↪  playsound(ruta+'/RespuestaCorrecta.mp3')
      elif repetir==True:
        soundrepetir = playsound(ruta+'/Repetir.mp3')
        print('''Repitamos el ejercicio.''')
        nR+=1
      else:
        print("      ¡RESPUESTA INCORRECTA!\n")
        soundrespuestacorrecta =
        ↪  playsound(ruta+'/RespuestaIncorrecta.mp3')
        err+=1
    timeFA+=timeFF-time1F
```

```
      timeSEA+=timeFF-time1SE
      timeEA+=timeE
      timeCEA+=(timeFF-time1SE)+timeE
      nRA+=nR
      nEA+=nE
      nREA+=nR+nE
      errA+=err
      Fila_Datos=[conjC, timeFF-time1F, timeFA, timeFF-time1SE,
      ↪  timeSEA, timeE, timeEA,
      (timeFF-time1SE)+timeE, timeCEA, nR, nRA, nE, nEA, nR+nE, nREA,
      ↪  err, errA]
      Filas_Datos.append(Fila_Datos)
    soundretroa1 = playsound(ruta+'/Retroa1.mp3')
    print('''En esta actividad se realizaron''')
    DICTANUMERO (str(npreg2))
    soundretroa2 = playsound(ruta+'/Retroa2.mp3')
    print('''preguntas, y tus errores fueron''')
    DICTANUMERO (str(errA))
    if (errA<(npreg2/2)):
      soundfelicitaciones2 = playsound(ruta+'/Felicitaciones2.mp3')
      print('''Felicitaciones. Haz concluido la 'Evaluación 2'.''')
      soundsiguienteactividad =
      ↪  playsound(ruta+'/SiguienteActividad.mp3')
      print('''Puedes pasar a la siguiente actividad.''')
    else:
      soundintentanuevamente =
      ↪  playsound(ruta+'/IntentaNuevamente.mp3')
      print('''Por la cantidad de respuestas incorrectas que tuviste
      ↪  en esta actividad te
      sugerimos volver a realizarla antes de pasar a la
↪   siguiente.''')
    df_dat2=pd.DataFrame(Filas_Datos, columns=['Conjunto respuesta',
    ↪  'Tiempo final en contestar
    correctamente', 'Tiempo final en contestar correctamente
↪   acumulado', 'Tiempo sin exploración
    por pregunta', 'Tiempo sin exploración acumulado', 'Tiempo de
↪   exploración por pregunta',
    'Tiempo de exploración acumulado', 'Tiempo con exploración por
    ↪  pregunta', 'Tiempo con
    exploración acumulado', 'Número de repeticiones por pregunta',
↪   'Número de repeticiones
    acumulado', 'Número de exploraciones por pregunta', 'Número de
↪   exploraciones acumulado',
    'Número de repeticiones más exploraciones por pregunta', 'Número
    ↪  de repeticiones más
    exploraciones acumulado', 'Errores por pregunta', 'Errores
↪   acumulados'])
    df_dat2
    df_dat2.to_csv(ruta+'/CD2/dat2_D_.csv')


  elif act=='3':
    Fila_Datos=[]
    Filas_Datos=[]
    timeFA=0
    timeSEA=0
    timeEA=0
    timeCEA=0
    nRA=0
    nEA=0
    nREA=0
    errA=0
    soundevaluacion3 = playsound(ruta+'/Evaluacion3.mp3')
    print('''Haz ingresado a la 'Evaluación 3'. En esta parte se
    ↪  evaluará la diferencia del
    Conjunto A con el Conjunto B, y posteriormente la diferencia del
↪   Conjunto B con el Conjunto A,
    existiendo la posibilidad de que haya conjuntos vacíos. Nótese
↪   que la operación diferencia
    no es conmutativa, o sea, que no da el mismo resultado y depende
↪   del orden de los conjuntos
    al hacer la operación diferencia.''')
    i=0
    while i<npreg3: #Con esta instrucción se creará un bucle
    ↪  dependiendo que cuantas preguntas
    se requieran en la actividad.
      conjA=set() #Creamos un primer conjunto de números aleatorios.
      conjB=set() #Creamos un segundo conjunto de números
      ↪  aleatorios.
      LLENARCONJUNTOS(conjA,5,7)
      LLENARCONJUNTOS(conjB,5,7)
      conjC=set()
      conjC=conjA-conjB #Creamos el conjunto diferencia (Respuesta)
      i+=1
      print('   PREGUNTA', i, 'parte 1:')
      print('Conjunto A:', conjA)
      print('Conjunto B:', conjB)
      print('Conjunto Diferencia (Respuesta):', conjC)
      soundact3parte1 = playsound(ruta+'/Act3Parte1.mp3')
      print('''Parte 1: Diferencia del Conjunto A con el Conjunto
      ↪  B.''')
      incorrecto=True
      repetir=False
      time1SE=time.time()
      timeE=0
      nR=0
      nE=0
      err=0
      while incorrecto: #Ponemos un while para que se repita la
      ↪  pregunta hasta que sea contestada
      correctamente
        time1R=time.time()
        soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')
        DICTACONJUNTO(conjA)
        soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')
        DICTACONJUNTO(conjB)
        conjD=set()
        print('Escribe tu respuesta:')
        soundpideresp = playsound(ruta+'/PideResp.mp3')
        repetir=False
        time1F=time.time()
        time2R=time.time()
        time1SE+=time2R-time1R
        if (len(conjC)==0):
          resp=input()
          if (resp=='+'):
            repetir=True
          elif (resp==''):
            conjD=set()
          elif (resp=='*'):
            listA=list(conjA)
            listB=list(conjB)
            time1E=time.time()
            EXPLORACONJUNTOS (listA, listB, 1)
            time2E=time.time()
            time1F+=time2E-time1E
            time1SE+=time2E-time1E
            timeE+=time2E-time1E
            nE+=1
            print('Continua tu respuesta:')
            soundpideresp = playsound(ruta+'/PideResp.mp3')
            resp=input()
            if (resp==''):
              conjD=set()
            else:
              conjD.add(resp)
          else:
            conjD.add(resp)
        else:
          j=0
          while (j<(len(conjC))): #Recibimos el conjunto respuesta
          ↪  del teclado.
            j+=1
            resp=input()
            if (resp=='+'):
              repetir=True
              break
            if (resp=='*'):
              listA=list(conjA)
              listB=list(conjB)
              time1E=time.time()
              EXPLORACONJUNTOS (listA, listB, 1)
              time2E=time.time()
              time1F+=time2E-time1E
              time1SE+=time2E-time1E
              timeE+=time2E-time1E
              nE+=1
              print('Continua tu respuesta:')
              soundpideresp = playsound(ruta+'/PideResp.mp3')
              j-=1
            else:
              conjD.add(resp)
        if (conjC==conjD and repetir==False): #Revisamos si el
        ↪  usuario respondió adecuadamente.
          timeFF=time.time()
          incorrecto=False
          print("      ¡RESPUESTA CORRECTA!\n")
          soundrespuestacorrecta =
          ↪  playsound(ruta+'/RespuestaCorrecta.mp3')
        elif repetir==True:
          soundrepetir = playsound(ruta+'/Repetir.mp3')
          print('''Repitamos el ejercicio.''')
          nR+=1
        else:
          print("      ¡RESPUESTA INCORRECTA!\n")
          soundrespuestacorrecta =
          ↪  playsound(ruta+'/RespuestaIncorrecta.mp3')
          err+=1
      conjC=conjB-conjA #Creamos el conjunto diferencia (Respuesta)
      print('Parte 2:\nConjunto Diferencia (Respuesta):', conjC)
      soundact3parte2 = playsound(ruta+'/Act3Parte2.mp3')
      print('''Parte 2: Diferencia del Conjunto B con el Conjunto
↪   A.''')
      incorrecto=True
      repetir=False
      time1SE2=time.time()
      timeE2=0
      nR2=0
      nE2=0
      err2=0
      while incorrecto: #Ponemos un while para que se repita la
      ↪  pregunta hasta que sea contestada
```

```
    correctamente.
      time1R2=time.time()
      soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')
      DICTACONJUNTO(conjB)
      soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')
      DICTACONJUNTO(conjA)
      conjD=set()
      print('Escribe tu respuesta:')
      soundprideresp = playsound(ruta+'/PideResp.mp3')
      repetir=False
      time1F2=time.time()
      time2R2=time.time()
      time1SE2+=time2R2-time1R2
      if (len(conjC)==0):
        resp=input()
        if (resp=='+'):
          repetir=True
        elif (resp==''):
          conjD=set()
        elif (resp=='*'):
          listA=list(conjA)
          listB=list(conjB)
          time1E2=time.time()
          EXPLORACONJUNTOS (listB, listA, 2)
          time2E2=time.time()
          time1F2+=time2E2-time1E2
          time1SE2+=time2E2-time1E2
          timeE2+=time2E2-time1E2
          nE2+=1
          print('Continua tu respuesta:')
          soundprideresp = playsound(ruta+'/PideResp.mp3')
          resp=input()
          if (resp==''):
            conjD=set()
          else:
            conjD.add(resp)
        else:
          conjD.add(resp)
      else:
        j=0
        while (j<(len(conjC))): #Recibimos el conjunto respuesta
        ↪  del teclado.
          j+=1
          resp=input()
          if (resp=='+'):
            repetir=True
            break
          if (resp=='*'):
            listA=list(conjA)
            listB=list(conjB)
            time1E2=time.time()
            EXPLORACONJUNTOS (listB, listA, 2)
            time2E2=time.time()
            time1F2+=time2E2-time1E2
            time1SE2+=time2E2-time1E2
            timeE2+=time2E2-time1E2
            nE2+=1
            print('Continua tu respuesta:')
            soundprideresp = playsound(ruta+'/PideResp.mp3')
            j-=1
          else:
            conjD.add(resp)
      if (conjC==conjD and repetir==False): #Revisamos si el
      ↪  usuario respondió adecuadamente.
        timeFF2=time.time()
        incorrecto=False
        print("      ¡RESPUESTA CORRECTA!\n")
        soundrespuestacorrecta =
        ↪  playsound(ruta+'/RespuestaCorrecta.mp3')
      elif repetir==True:
        soundrepetir = playsound(ruta+'/Repetir.mp3')
        print('''Repitamos el ejercicio.''')
        nR2+=1
      else:
        print("      ¡RESPUESTA INCORRECTA!\n")
        soundrespuestacorrecta =
        ↪  playsound(ruta+'/RespuestaIncorrecta.mp3')
        err2+=1
    timeFA+=(timeFF-time1F)+(timeFF2-time1F2)
    timeSEA+=(timeFF-time1SE)+(timeFF2-time1SE2)
    timeEA+=timeE+timeE2
    timeCEA+=((timeFF-time1SE)+timeE)+((timeFF2-time1SE2)+timeE2)
    nRA+=nR+nR2
    nEA+=nE+nE2
    nREA+=(nR+nE)+(nR2+nE2)
    errA+=err+err2
    Fila_Datos=[conjC, (timeFF-time1F)+(timeFF2-time1F2), timeFA,
    (timeFF-time1SE)+(timeFF2-time1SE2), timeSEA, timeE+timeE2,
    ↪  timeEA,
    ((timeFF-time1SE)+timeE)+((timeFF2-time1SE2)+timeE2), timeCEA,
    ↪  nR+nR2, nRA, nE+nE2, nEA,
    (nR+nE)+(nR2+nE2), nREA, err+err2, errA]
    Filas_Datos.append(Fila_Datos)
  soundretroa1 = playsound(ruta+'/Retroa1.mp3')
  print('''En esta actividad se realizaron''')
  DICTANUMERO (str(npreg3))
  soundretroa2 = playsound(ruta+'/Retroa2.mp3')
  print('''preguntas, y tus errores fueron''')
  DICTANUMERO (str(errA))
  if (errA<(npreg3/2)):
    soundrfelicitaciones3 = playsound(ruta+'/Felicitaciones3.mp3')
    print('''Felicitaciones. Haz concluido la 'Evaluación 3'.''')
    soundsiguienteactividad =
    ↪  playsound(ruta+'/SiguienteActividad.mp3')
    print('''Puedes pasar a la siguiente actividad.''')
  else:
    soundintentanuevamente =
    ↪  playsound(ruta+'/IntentaNuevamente.mp3')
    print('''Por la cantidad de respuestas incorrectas que tuviste
    ↪  en esta actividad te
    sugerimos volver a realizarla antes de pasar a la
↪  siguiente.''')
  df_dat3=pd.DataFrame(Filas_Datos, columns=['Conjunto respuesta',
  ↪  'Tiempo final en contestar
  correctamente', 'Tiempo final en contestar correctamente
↪  acumulado', 'Tiempo sin exploración
  por pregunta', 'Tiempo sin exploración acumulado', 'Tiempo de
↪  exploración por pregunta',
  'Tiempo de exploración acumulado', 'Tiempo con exploración por
  ↪  pregunta', 'Tiempo con
  exploración acumulado', 'Número de repeticiones por pregunta',
↪  'Número de repeticiones
  acumulado', 'Número de exploraciones por pregunta', 'Número de
↪  exploraciones acumulado',
  'Número de repeticiones más exploraciones por pregunta', 'Número
  ↪  de repeticiones más
  exploraciones acumulado', 'Errores por pregunta', 'Errores
↪  acumulados'])
  df_dat3
  df_dat3.to_csv(ruta+'/CD3/dat3_D_.csv')

elif act=='4':
  Fila_Datos=[]
  Filas_Datos=[]
  timeFA=0
  timeSEA=0
  timeEA=0
  timeCEA=0
  nRA=0
  nEA=0
  nREA=0
  errA=0
  soundevaluacion4 = playsound(ruta+'/Evaluacion4.mp3')
  print('''Haz ingresado a la 'Evaluación 4'. En esta parte se
  ↪  evaluará la diferencia de
  tres conjuntos con la posibilidad de que haya conjuntos
↪  vacíos.''')
  i=0
  while i<npreg4: #Con esta instrucción se creará un bucle
  ↪  dependiendo que cuantas preguntas
  se requieran en la actividad.
    conjA=set() #Creamos un primer conjunto de números aleatorios.
    conjB=set() #Creamos un segundo conjunto de números
    ↪  aleatorios.
    conjC=set() #Creamos un tercer conjunto de números aleatorios.
    LLENARCONJUNTOS(conjA,7,8)
    LLENARCONJUNTOS(conjB,3,8)
    LLENARCONJUNTOS(conjC,3,8)
    conjD=set()
    conjD=conjA-conjB-conjC #Creamos el conjunto diferencia
    ↪  (Respuesta)
    i+=1
    print('   PREGUNTA', i)
    print('Conjunto A:', conjA)
    print('Conjunto B:', conjB)
    print('Conjunto C:', conjC)
    print('Conjunto Diferencia (Respuesta):', conjD)
    incorrecto=True
    repetir=False
    time1SE=time.time()
    timeE=0
    nR=0
    nE=0
    err=0
    while incorrecto: #Ponemos un while para que se repita la
    ↪  pregunta hasta que sea contestada
    correctamente
      time1R=time.time()
      soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')
      DICTACONJUNTO(conjA)
      soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')
      DICTACONJUNTO(conjB)
      soundconjuntoC = playsound(ruta+'/ConjuntoC.mp3')
      DICTACONJUNTO(conjC)
      conjE=set()
      print('Escribe tu respuesta:')
      time1=time.time()
      if (repetir==False):
        time1r=time.time()
      soundprideresp = playsound(ruta+'/PideResp.mp3')
      repetir=False
      time1F=time.time()
```

```
      time2R=time.time()
      time1SE+=time2R-time1R
      if (len(conjD)==0):
        resp=input()
        if (resp=='+'):
          repetir=True
        elif (resp==''):
          conjE=set()
        elif (resp=='*'):
          listA=list(conjA)
          listB=list(conjB)
          listC=list(conjC)
          time1E=time.time()
          EXPLORACONJUNTOS2 (listA, listB, listC, 1)
          time2E=time.time()
          time1F+=time2E-time1E
          time1SE+=time2E-time1E
          timeE+=time2E-time1E
          nE+=1
          print('Continua tu respuesta:')
          soundpideresp = playsound(ruta+'/PideResp.mp3')
          resp=input()
          if (resp==''):
            conjE=set()
          else:
            conjE.add(resp)
        else:
          conjE.add(resp)
      else:
        j=0
        while (j<(len(conjD))): #Recibimos el conjunto respuesta
        ↪  del teclado.
          j+=1
          resp=input()
          if (resp=='+'):
            repetir=True
            break
          if (resp=='*'):
            listA=list(conjA)
            listB=list(conjB)
            listC=list(conjC)
            time1E=time.time()
            EXPLORACONJUNTOS2 (listA, listB, listC, 1)
            time2E=time.time()
            time1F+=time2E-time1E
            time1SE+=time2E-time1E
            timeE+=time2E-time1E
            nE+=1
            print('Continua tu respuesta:')
            soundpideresp = playsound(ruta+'/PideResp.mp3')
            j-=1
          else:
            conjE.add(resp)
      if (conjD==conjE and repetir==False): #Revisamos si el
      ↪  usuario respondió adecuadamente.
        timeFF=time.time()
        incorrecto=False
        print("      ¡RESPUESTA CORRECTA!\n")
        soundrespuestacorrecta =
        ↪  playsound(ruta+'/RespuestaCorrecta.mp3')
      elif repetir==True:
        soundrepetir = playsound(ruta+'/Repetir.mp3')
        print('''Repitamos el ejercicio.''')
        nR+=1
      else:
        print("      ¡RESPUESTA INCORRECTA!\n")
        soundrespuestacorrecta =
        ↪  playsound(ruta+'/RespuestaIncorrecta.mp3')
        err+=1
    timeFA+=timeFF-time1F
    timeSEA+=timeFF-time1SE
    timeEA+=timeE
    timeCEA+=(timeFF-time1SE)+timeE
    nRA+=nR
    nEA+=nE
    nREA+=nR+nE
    errA+=err
    Fila_Datos=[conjC, timeFF-time1F, timeFA, timeFF-time1SE,
    ↪  timeSEA, timeE, timeEA,
    (timeFF-time1SE)+timeE, timeCEA, nR, nRA, nE, nEA, nR+nE, nREA,
    ↪  err, errA]
    Filas_Datos.append(Fila_Datos)
  soundretroa1 = playsound(ruta+'/Retroa1.mp3')
  print('''En esta actividad se realizaron''')
  DICTANUMERO (str(npreg4))
  soundretroa2 = playsound(ruta+'/Retroa2.mp3')
  print('''preguntas, y tus errores fueron''')
  DICTANUMERO (str(errA))
  if (errA<(npreg4/2)):
    soundrfelicitaciones4 = playsound(ruta+'/Felicitaciones4.mp3')
    print('''Felicitaciones. Haz concluido la 'Evaluación 4'.''')
    soundsiguienteactividad =
    ↪  playsound(ruta+'/SiguienteActividad.mp3')
    print('''Puedes pasar a la siguiente actividad.''')
  else:
    soundintentanuevamente =
    ↪  playsound(ruta+'/IntentaNuevamente.mp3')
    print('''Por la cantidad de respuestas incorrectas que tuviste
    ↪  en esta actividad te
    sugerimos volver a realizarla antes de pasar a la
↪   siguiente.''')
    df_dat4=pd.DataFrame(Filas_Datos, columns=['Conjunto respuesta',
    ↪  'Tiempo final en contestar
    correctamente', 'Tiempo final en contestar correctamente
↪   acumulado', 'Tiempo sin exploración
    por pregunta', 'Tiempo sin exploración acumulado', 'Tiempo de
↪   exploración por pregunta',
    'Tiempo de exploración acumulado', 'Tiempo con exploración por
    ↪  pregunta', 'Tiempo con
    exploración acumulado', 'Número de repeticiones por pregunta',
↪   'Número de repeticiones
    acumulado', 'Número de exploraciones por pregunta', 'Número de
↪   exploraciones acumulado',
    'Número de repeticiones más exploraciones por pregunta', 'Número
    ↪  de repeticiones más
    exploraciones acumulado', 'Errores por pregunta', 'Errores
↪   acumulados'])
    df_dat4
    df_dat4.to_csv(ruta+'/CD4/dat4_D_.csv')


  elif act=='5':
    i=0
    while i<npreg0: #Con esta instrucción se creará un bucle
    ↪  dependiendo que cuantas preguntas
    se requieran en la actividad.
      conjA=set() #Creamos un primer conjunto de números aleatorios.
      conjB=set() #Creamos un segundo conjunto de números
      ↪  aleatorios.
      LLENARCONJUNTOS(conjA,3,4)
      LLENARCONJUNTOS(conjB,3,4)
      conjC=set()
      conjC=conjA-conjB #Creamos el conjunto intersección
      ↪  (Respuesta).
      if len(conjC)!=0:
        i+=1
        print('  EJERCICIO DE PRÁCTICA', i)
        print('Conjunto A:', conjA)
        print('Conjunto B:', conjB)
        print('Conjunto Intersección (Respuesta):', conjC)
        incorrecto=True
        while incorrecto: #Ponemos un while para que se repita la
        ↪  pregunta hasta que sea
        contestada correctamente.
          soundconjuntoA = playsound(ruta+'/ConjuntoA.mp3')
          DICTACONJUNTO(conjA)
          soundconjuntoB = playsound(ruta+'/ConjuntoB.mp3')
          DICTACONJUNTO(conjB)
          conjD=set()
          soundconjuntoresp = playsound(ruta+'/ConjuntoResp.mp3')
          print('''La respuesta es:''')
          DICTACONJUNTO(conjC)
          print('Escribe tu respuesta:')
          soundpideresp = playsound(ruta+'/PideResp.mp3')
          repetir=False
          j=0
          while (j<(len(conjC))): #Recibimos el conjunto respuesta
          ↪  del teclado.
            j+=1
            resp=input()
            if (resp=='+'):
              repetir=True
              break
            if (resp==''):
              break
            if (resp=='*'):
              listA=list(conjA)
              listB=list(conjB)
              EXPLORACONJUNTOS (listA, listB, 1)
              print('Continua tu respuesta:')
              soundpideresp = playsound(ruta+'/PideResp.mp3')
              j-=1
            else:
              conjD.add(resp)
          if conjC==conjD: #Revisamos si el usuario respondió
          ↪  adecuadamente.
            incorrecto=False
            print("      ¡RESPUESTA CORRECTA!\n")
            soundrespuestacorrecta =
            ↪  playsound(ruta+'/RespuestaCorrecta.mp3')
          elif repetir==True:
            soundrepetir = playsound(ruta+'/Repetir.mp3')
            print('''Repitamos el ejercicio.''')
          else:
            print("      ¡RESPUESTA INCORRECTA!\n")
            soundrespuestaincorrecta =
            ↪  playsound(ruta+'/RespuestaIncorrecta.mp3')
        soundmas = playsound(ruta+'/Mas.mp3')
        print('''Si deseas otro ejercicio de práctica presiona la
        ↪  tecla de '+' seguido de un
        enter, sino solo enter.''')
        mas=input('¿Practicar de nuevo?:')
```

```
        if (mas=='+'):
          i-=1
  elif act=='6':
    print("Estás saliendo.")
  else:
    print("Opción erronea")
```

```
    soundrespuestaincorrecta =
    ↪  playsound(ruta+'/RespuestaIncorrecta.mp3')
soundsalida = playsound(ruta+'/Salida.mp3')
print('''Estás saliendo del programa para la enseñanza y evaluación
↪  de la diferencia. Muchas
gracias por participar.''')
```

Anexo C

Resumen de los datos por promedio

	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11	U12	U13	U14	U15	U16	U17	U18	Media
Un_N1	329.5	28.1	31.4	38.4	45.8	23.6	30.9	16.2	46.8	170.6	93.5	33.3	86.3	1.5	79.5	3.0	0.5	23.6	60.13
Un_N2	10.0	18.8	6.7	10.3	21.8	8.0	11.9	8.3	15.3	40.2	7.2	1.6	22.9	0.4	27.1	3.8	0.1	6.3	12.26
Un_N3	67.5	32.4	14.6	20.2	51.5	43.6	23.3	23.3	39.2	117.0	7.8	11.0	77.4	0.2	40.9	15.7	0.0	14.9	33.36
Un_N4	37.5	24.3	84.5	31.4	20.8	31.5	12.7	20.1	15.5	85.7	1.2	10.1	39.1	1.4	59.8	1.5	0.0	8.6	26.98

Cuadro C.1: Promedio por columnas de la operación Unión para cada uno de sus niveles.

	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11	U12	U13	U14	U15	U16	U17	U18	Media
In_N1	7.9	6.9	5.2	5.9	16.6	4.6	6.5	4.5	3.4	15.5	12.7	2.6	9.9	18.4	21.2	24.9	0.1	3.1	9.43
In_N2	7.8	8.5	4.0	11.4	15.6	11.2	6.1	15.4	4.3	11.9	48.3	4.5	10.4	126.8	39.5	16.0	6.8	5.6	19.6
In_N3	130.4	13.2	7.5	10.6	22.0	6.5	8.8	21.8	9.3	15.5	27.5	3.8	13.2	36.1	26.4	20.8	1.9	15.2	21.69
In_N4	3.5	25.5	11.2	5.7	50.6	6.0	18.6	9.5	20.1	4.3	7.2	6.0	41.4	13.9	10.3	50.6	0.6	54.6	18.86

Cuadro C.2: Promedio por columnas de la operación Intersección para cada uno de sus niveles.

	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11	U12	U13	U14	U15	U16	U17	U18	Media
Di_N1	43.1	39.0	36.5	40.8	20.4	41.8	41.9	117.0	27.6	97.6	10.8	22.6	29.5	44.6	45.7	72.7	21.9	8.8	42.35
Di_N2	53.1	11.4	4.3	31.0	52.4	49.8	16.4	92.6	12.7	128.8	10.5	8.8	16.0	301.2	152.7	80.9	10.2	11.0	57.98
Di_N3	44.0	30.2	54.9	47.3	134.6	34.9	15.8	188.5	28.1	65.0	96.5	10.7	118.1	821.5	169.7	41.0	22.4	25.6	108.26
Di_N4	49.6	28.0	27.0	40.9	200.5	94.3	20.0	124.5	42.8	618.2	30.2	8.1	54.1	75.1	111.2	15.1	30.4	32.2	89.01

Cuadro C.3: Promedio por columnas de la operación Diferencia para cada uno de sus niveles.

	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11	U12	U13	U14	U15	U16	U17	U18	Media
Co_N1	6.5	8.3	3.9	18.9	29.4	4.4	18.0	142.4	4.5	17.4	44.0	1.1	14.7	17.4	135.3	20.3	8.2	2.8	27.63
Co_N2	12.5	53.0	14.6	72.7	149.9	139.3	11.2	56.7	153.3	131.1	63.2	609.4	526.2	131.1	39.6	7.5	13.1	4.6	121.61
Co_N3	94.6	11.5	8.0	46.4	48.5	5.7	14.7	67.3	56.2	19.3	99.9	103.4	78.4	19.3	74.9	2.5	17.3	6.3	43.01
Co_N4	150.0	34.1	81.7	111.1	43.3	136.1	67.2	132.1	121.6	23.2	99.9	92.3	42.4	2.5	78.4	210.2	64.1	34.4	81.58

Cuadro C.4: Promedio por columnas de la operación Complemento para cada uno de sus niveles.

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	Media
Unión	111.1	25.9	34.3	25.1	35.0	26.7	19.7	17.0	29.2	103.4	27.4	14.0	56.4	0.9	51.8	6.0	0.2	13.3	33.18
Intersección	37.4	13.5	7.0	8.4	26.2	7.1	10.0	12.8	9.3	11.8	23.9	4.2	18.7	48.8	24.4	28.1	2.4	19.6	17.42
Diferencia	47.4	27.1	30.7	40.0	102.0	55.2	23.5	130.7	27.8	227.4	37.0	12.6	54.4	310.6	119.8	52.4	21.2	19.4	74.40
Complemento	65.9	26.7	27.1	62.3	67.8	71.4	27.7	99.6	83.9	47.8	76.7	201.5	165.4	42.6	82.1	60.1	25.7	12.0	68.46

Cuadro C.5: Promedio de todas las operaciones con todos sus niveles

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	Media
Media de usuario	65.4	23.3	24.8	34	57.8	40.1	20.2	65	37.6	97.6	41.2	58.1	73.7	100.7	69.5	36.6	12.4	16.1	48.56

Cuadro C.6: Promedio de todas las operaciones con todos sus niveles

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Un_N1	83.9	198.9	86.8	30.8	28.8	49.9	33.0	29.5	29.0	30.8
In_N1	14.8	14.7	9.3	9.7	11.1	5.7	5.9	7.8	4.6	10.7
Di_N1	81.8	39.3	72.9	23.9	41.2	36.9	27.7	44.6	31.0	24.1
Co_N1	82.2	62.1	41.4	18.9	8.3	7.7	22.7	15.3	9.0	8.5

Cuadro C.7: Promedio del nivel 1 de todas las operaciones

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Un_N2	17.3	13.0	11.9	9.9	16.1	11.3	10.1	10.4	12.1	10.5
In_N2	31.5	13.2	8.3	13.1	11.5	18.9	31.2	28.8	7.0	33.3
Di_N2	52.2	37.3	69.3	59.9	81.5	55.2	25.0	98.9	39.4	61.2
Co_N2	405.5	125.6	362.5	58.4	78.8	38.8	50.4	43.7	22.3	30.3

Cuadro C.8: Promedio del nivel 2 de todas las operaciones

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Un_N3	31.6	37.4	37.5	36.8	34.4	36.3	22.6	25.0	42.7	29.3
In_N3	39.0	54.1	12.8	13.2	28.0	11.0	17.7	17.9	12.0	11.2
Di_N3	570.3	82.2	51.6	45.8	56.1	34.1	34.4	101.2	54.7	52.3
Co_N3	68.1	36.2	46.3	32.0	32.1	91.3	47.1	38.8	19.9	18.3

Cuadro C.9: Promedio del nivel 3 de todas las operaciones

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Un_N4	25.7	66.2	18.4	14.6	34.3	20.9	20.9	21.1	21.0	26.8
In_N4	50.1	23.9	12.4	10.3	16.3	14.6	7.3	11.5	6.8	35.7
Di_N4	79.4	68.9	49.7	43.0	118.0	28.7	60.6	361.8	46.9	33.0
Co_N4	129.5	34.2	119.2	88.7	77.2	121.9	51.1	43.2	118.6	63.4

Cuadro C.10: Promedio del nivel 4 de todas las operaciones

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Media
Nivel 1	65.7	78.8	52.6	20.8	22.4	25.0	22.3	24.3	18.4	18.5	34.88
Nivel 2	126.6	47.3	113.0	35.3	47.0	31.0	29.2	45.4	20.2	33.8	52.89
Nivel 3	177.2	52.5	37.1	32.0	37.7	43.2	30.4	45.7	32.3	27.8	51.57
Nivel 4	71.2	48.3	49.9	39.1	61.5	46.5	35.0	109.4	48.3	39.7	54.13

Cuadro C.11: Promedio de todas las operaciones con todos sus niveles

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Media por pregunta	110.2	56.7	63.1	31.8	42.1	36.4	29.2	56.2	29.8	30

Cuadro C.12: Promedio de tiempo por realizar cada nivel

Índice alfabético

Bibliografía

[1] Andrew Churches. Taxonomía de bloom para la era digital. https://eduteka.icesi.edu.co/articulos/TaxonomiaBloomDigital, 2009.

[2] Instituto Nacional de Estadística y Geografía (INEGI). Caracteristicas de las personas con discapacidad visual. 2020.

[3] Instituto Nacional de Estadística y Geografía (INEGI). Discapacidad visual. https://www.inegi.org.mx/rnm/index.php/catalog/632/datafile/F13/V325, 2020.

[4] Organización Mundial de la Salud (OMS). Salud visual. https://www.who.int/es/news-room/fact-sheets/detail/blindness-and-visual-impairment, 2020.

[5] Julian J. Faraway. Practical regression and anova using r. 2000.

[6] INEGI. Estadísticas a propósito del día internacional de las personas con discapacidad (datos nacionales). https://www.inegi.org.mx/contenidos/saladeprensa/aproposito/2021/EAP_PersDiscap21.pdf, 2021.

[7] Miguel Elías Jaime Sánchez. Aprendizaje de ciencias a través de audio en niños ciegos. 2006.

[8] Abraham Magendzo. La diversidad y la no discriminación: un desafío para una educación moderna. http://publicaciones.horizonteenfermeria.uc.cl/index.php/pel/article/view/25659, 2000.

[9] Douglas Mongomery. *Probabilidad estadística aplicadas a la ingeniería*. Arizona, 2003.

[10] Mariella Aguirre Candray Napoleón Candray Pleitez. La historia de la ceguera. *Revista Española de Historia y Humanidades en Oftalmología*, 2019.

[11] Gerardo Vecilla Antolínes Raúl Martín Herranz. Manual de optometría. 2018.

[12] Jaime Sánchez. Una metodología para desarrollar y evaluar la usabilidad de entornos virtuales basados en audio para el aprendizaje y cogniciíon de usuarios ciegos. https://www.redalyc.org/pdf/3314/331427213011.pdf, 2010.

[13] C. Fuentes Vargas. *Estrategia didáctica para el aprendizaje de conceptos algebráicos en estudiantes con discapacidad visual*. CDMX, 2017.

[14] Luis Villoro. *Creer, Saber, Conocer*. Buenos Aires, Argentina, 1989.