

*Universitat Politècnica de Catalunya  
Facultat d'Informàtica de Barcelona*

## *Práctica de SBC*



**Andres Lucian Laptés Costan**

andres.lucian.laptés@estudiantat.upc.edu

**Adrián Salazar Parés**

adrian.salazar@estudiantat.upc.edu

**Marc Gimenez Codina**

marc.gimenez.codina@estudiantat.upc.edu

# Índice

<b>1. Identificación del problema</b>	<b>2</b>
1.1. El enunciado	2
1.2. Viabilidad de usar un SBC	3
1.3. Identificación de las fuentes de conocimiento	4
1.4. Objetivos del problema	5
<b>2. Conceptualización del problema</b>	<b>6</b>
2.1. Descripción de los conceptos del dominio	6
2.2. Descripción de los subproblemas	7
2.3. Ejemplos de conocimiento experto	8
2.4. Descripción informal del proceso de resolución	9
<b>3. Formalización del problema</b>	<b>10</b>
3.1. Desarrollo de la ontología	10
3.2. Documentación de la ontología	11
3.3. Proceso de razonamiento y metodología de resolución	17
<b>4. Implementación del problema</b>	<b>18</b>
4.1. Ontología	18
4.2. Módulos	18
4.3. Elaboración del heurístico	34
4.4. Notas sobre las instancias	35
4.5. Criterios de implementación	36
4.6. Prototipos	36
<b>5. Juegos de pruebas</b>	<b>37</b>
5.1. Juego de pruebas normal 1	37
5.2. Juego de pruebas normal 2	42
5.3. Juego de pruebas mucho tiempo	45
5.4. Juego de pruebas poco tiempo	50
5.5. Juego de pruebas todas las preferencias	51
5.6. Juego de pruebas ninguna preferencia	55
5.7. Juego de pruebas mucha gente y niños	56
<b>6. Conclusiones</b>	<b>58</b>
<b>7. Progreso trabajo Innovación</b>	<b>59</b>

# 1. Identificación del problema

## 1.1. El enunciado

El problema radica en los recorridos por los museos de arte. Los museos albergan una amplia variedad de piezas de diversas épocas históricas, estilos de arte y temáticas, lo que puede parecer agobiante para aquellos visitantes que cuentan con un tiempo limitado. Así pues, el objetivo es elaborar una solución determinada en el entendimiento de las obras y las particularidades de los visitantes, con el fin de potenciar el disfrute y la utilidad de la experiencia.

Un museo se puede visitar de muchas maneras y es necesario adaptar la visita a la persona o grupo de personas. Cada museo es diferente en la manera en que organiza sus obras, de modo que la solución del problema también deberá poder adaptarse a las características particulares del museo.

Una obra de arte puede tener muchas características, pero podemos considerar un conjunto limitado de ellas que permita su selección para organizar una visita. Se ha decidido usar las siguientes características, ya que el estilo/escuela/corriente se puede interpretar como una mezcla entre temática y época:

- ***Año de creación:***
- ***Época (temporal)***
- ***Nombre***
- ***Autor***
- ***Sala en que está ubicada***
- ***Temática***
- ***Dimensiones***
- ***Complejidad***
- ***Relevancia***

Los visitantes pueden tener intereses variados hacia épocas o estilos artísticos, así como ciertos temas o artistas específicos. A esto se suma la variación en el nivel de conocimiento del arte, que puede ir desde expertos hasta principiantes. Asimismo, las visitas de los visitantes a un museo pueden ser en grupo. Se considerará como grupo también que el grupo esté conformado por un solo individuo, formando así un grupo de una persona. El grupo puede ser de más de una persona, como es el caso de una familia, por lo que también hay que tener en cuenta, además del número de integrantes, cuántos de ellos son niños.

Los visitantes pueden ser de muchos tipos y tener intereses variados, donde las características más relevantes son:

- **Número de días de visita**
- **Duración de la visita en horas**
- **Características sobre el grupo:**
  - **Tamaño**
  - **Tipo de grupo**
  - **Conocimiento de arte de los integrantes del grupo**
- **Preferencias:** pueden ser por época, tema, autor, estilo/escuela pictórica.

En función de estas características se puede elaborar un **itinerario**, que especifique un orden de visita y una estimación del tiempo necesario para cada obra.

El itinerario se debe elaborar en base a las características de los visitantes:

- El **número de horas** totales de visita no puede ser mayor que el número de horas disponibles del grupo. Esto sirve para elaborar restricciones del problema.
- Las **características del grupo** son relevantes a la hora de determinar el tiempo de visita de cada obra. Por ejemplo: Un grupo mayor tiende a tardar más tiempo en visitar un museo que una sola persona.
- **Las preferencias de cada persona** se tienen que intentar maximizar. Las preferencias se pueden usar para dar prioridad a las obras, es decir, elaborar una clasificación heurística.

El programa recibirá información sobre el grupo visitante, que se utilizará para elaborar restricciones para resolver el problema.

## 1.2. Viabilidad de usar un SBC

Se ha elegido usar un SBC en este problema por los siguientes motivos:

- **Imposibilidad de usar métodos algorítmicos:** El problema es demasiado extenso para poder usar un método débil. Las condiciones de la búsqueda cambian con cada instancia del problema, y por consiguiente no se puede elaborar una función heurística que represente adecuadamente las características del problema.

- **El problema se puede plantear como un proceso de razonamiento:** Este problema se asemeja a un proceso de razonamiento humano, ya que los visitantes evalúan de forma consciente o inconsciente cómo organizar su visita siguiendo reglas de sentido común. Otras reglas propias del dominio también van a ser expresables como reglas. De hecho, para cada solución se podría detallar fácilmente el conjunto de pasos que se han seguido para llegar a la conclusión.
- **Estructura del problema:** El problema cuenta con distintas características que permiten descomponer el problema en distintas reglas o preferencias, lo que facilita su tratamiento con un SBC. Además se puede identificar una estructura para así poder dividir el problema en pequeños problemas que podemos resolver (por una parte, pedir información al usuario, por otra quitar todas aquellas obras que el visitante no tenga interés).
- **Disponibilidad de conocimiento experto:** Existe información detallada sobre las obras en páginas web y bases de datos de museos, lo que permite desarrollar técnicas de resolución basadas en ese conocimiento.

### 1.3. Identificación de las fuentes de conocimiento

Las fuentes de conocimiento a usar para generar las reglas del SBC serán las siguientes páginas web de museos de renombre, donde nos servirán para saber la estructura de nuestras obras, así como los elementos a tener en cuenta para desarrollar un itinerario..

Las páginas oficiales de museos ofrecen información detallada sobre sus colecciones, características de las obras y recomendaciones de itinerarios. Algunos ejemplos son:

- **Museo del Prado:** <https://www.museodelprado.es/>
- **National Gallery (Londres):** <https://www.nationalgallery.org.uk>
- **Museu Nacional d'Art de Catalunya:** <https://www.museunacional.cat/>
- **Louvre:** <https://www.louvre.fr/>
- **Museo Thyssen-Bornemisza** <http://www.museothyssen.org/>

De estas fuentes podemos obtener:

- **Datos objetivos de las obras:** autor, estilo, dimensiones, relevancia, ubicación y época.
- **Características de las obras:** Estilo/escuela/corriente/periodo pictórico, temática.

Además, también se puede encontrar información importante en artículos, guías o bases de datos de arte, las cuales pueden ser de universidades, museos o revistas de arte.

## 1.4. Objetivos del problema

El problema plantea el diseño de un sistema inteligente que optimiza y personaliza las visitas a museos de arte, teniendo en cuenta las características tanto del visitante como de las obras de la colección. Estos objetivos se pueden desglosar en los siguientes puntos clave:

- **Obtener información.** Se obtiene información sobre el visitante para poder aplicar las distintas reglas y poder adaptar las visitas según los intereses, conocimientos y preferencias de los usuarios, tales como épocas, estilos artísticos, autores o temáticas específicas.
- **Optimizar el tiempo disponible:** Dividir las visitas en sesiones diarias de duración limitada, distribuyendo el recorrido para que se respete el tiempo estipulado por los usuarios dividiendo el número de horas de visita, en función de las obras y de las características del grupo.
- **Obviar las obras de poca relevancia para el visitante:** No se considerarán las obras cuyos atributos no sean relevantes para el visitante. De esta forma, se maximizan las obras en las que el visitante tiene interés.
- **Adaptar el itinerario al conocimiento del visitante:** Ajustar el nivel de detalle y duración del tiempo de observación de cada obra según el perfil de conocimiento artístico del visitante.

Los resultados esperados del sistema son la generación de itinerarios personalizados que optimicen, en cada caso, la visita al museo en función de las preferencias, conocimientos y disponibilidad de tiempo de los visitantes. Dichos itinerarios se centrarán en las obras consideradas más relevantes para cada perfil, con mínimos desplazamientos entre salas y máximo aprovechamiento del tiempo. Además, estima el tiempo de observación de cada obra y proporciona información adecuada a ese nivel de conocimiento para enriquecer la experiencia del usuario en el ámbito artístico y cultural. Finalmente, el visitante habrá pasado una visita muy satisfactoria y organizada.

## 2. Conceptualización del problema

### 2.1. Descripción de los conceptos del dominio

El dominio de este problema es un museo de arte. El museo contiene obras, que pueden tener muchos atributos, pero algunos son más relevantes para este problema, como por ejemplo las dimensiones de la obra, la temática, o el pintor de la obra. Además, estas obras pueden estar organizadas de distintas maneras. Cada obra está en una sala pero la sala puede estar organizada según diversos criterios. Para simplificar, se puede suponer que sólo pueden estar ordenadas cronológicamente, temáticamente o según el artista. Finalmente, cada visitante tiene distintas características, sean preferencias, tamaño del grupo u otros factores como por ejemplo, si hay niños en el grupo. Por lo tanto, los elementos del dominio són los siguientes:

- **Pintor:** De la información de un pintor se puede guardar una gran cantidad de información pero para esta práctica solo se guarda el nombre del pintor, que lo identifica, así como la época a la cual pertenece, ya que el periodo histórico influye en el estilo de pintura.
- **Temática:** La temática básicamente dicta que a grandes rasgos de qué trata la obra, se guarda únicamente el nombre de la temática, que la identifica. Sirve para clasificar las obras según características particulares de las obras, no relacionadas con el autor.
- **Época:** Representa una etapa histórica de la humanidad donde solo se guarda el nombre de la época, que la identifica, así como el año de inicio y el año de finalización.
- **Obra:** Es el objeto en el que se centra el problema. Todos los elementos del dominio tienen algún tipo de relación con la obra, directa o indirecta. Por este motivo, la obra tiene una relación con el pintor que la ha pintado, tiene una o más temáticas, y pertenece a una época. Además, la obra tiene los atributos área y relevancia, que sirven para calcular el tiempo de visita así como para determinar qué obras ver, y el atributo nombre, que la identifica.
- **Visitante:** El visitante es el otro elemento relevante del problema, ya que define las reglas para determinar el itinerario a seguir. Puede tener o no una o más de una temática, un artista y una época favorita. También se guarda el número de personas, la cantidad de niños, el tiempo de visita por día y el número de días de visita.

- **Sala:** Las salas contienen obras, y cada sala tiene un número asignado, que las identifica. Las salas pueden estar ordenadas de distintas maneras:
  - **Sala ordenada por artista:** Está relacionada con un artista. Todas las obras de esa sala son de ese artista.
  - **Sala ordenada cronológicamente:** Está relacionada con una época. Todas las obras de esa sala son de esa época.
  - **Sala ordenada por temática:** Está relacionada con una temática. Todas las obras de esa sala son de esa temática.
- **Visita:** Los visitantes realizan una visita. Está relacionada con un museo.
- **Museo:** Un museo tiene un nombre, que lo identifica, y contiene salas.

El resultado final será una template que devuelve una lista de obras por cada día de visita, se explicará con más detalle la template en posteriores apartados.

## 2.2. Descripción de los subproblemas

Se divide el problema principal en 4 subproblemas bien diferenciadas:

- **Caracterización:** En este subproblema básicamente se recolecta toda la información necesaria para empezar el problema, aquí se obtienen las características relevantes de las obras para su clasificación y selección en un itinerario (e.g estilo, autor, temática, año principio, etc...).

Asimismo, también se obtiene la información necesaria o relevante de los visitantes mediante una serie de preguntas, por ejemplo la información sobre el nivel de conocimiento artístico, también el tiempo disponible para hacer la visita o cuántos participantes irán a la visita. Básicamente en este subproblema se recopila la información de las preguntas donde se entrará en detalle posteriormente.

- **Procesado:** En este subproblema la idea principal es filtrar las obras de arte de tal manera que solo se procesen obras de artes que al menos tenga un mínimo de interés en ellas. Aquí básicamente lo que se hace es eliminar aquellas obras que en que el visitante no tenga ni una sola preferencia.
- **Construcción:** Aquí en este subproblema donde lo que se hace es con las obras que sí interesan al visitante o a los visitantes, generamos un itinerario en el que se cumplan las preferencias y restricciones del visitante y, además, en este subproblema se genera con los datos de la fase de caracterización la solución.
- **Presentación:** En este subproblema se muestra al visitante las obras en donde se tiene que ir, básicamente en este subproblema se le muestra el resultado al visitante.



Estos subproblemas se resolverán de forma secuencial, ya que cada subproblema necesitan la salida del anterior para poder funcionar correctamente

## 2.3. Ejemplos de conocimiento experto

Después de la investigación, se ha recopilado la información clave necesaria para completar este proyecto. Aquí hay una lista de ejemplos de cómo se puede aplicar esta experiencia al diseño y optimización de itinerarios de museos:

### ***Clasificación de obras según su importancia para los visitantes***

- Se ha determinado que uno de los elementos que influyen a la hora de contemplar una obra de arte es la relevancia de esa obra. Usando información proporcionada por museos como el Louvre, que en su [guía de cómo visitar el museo](#) únicamente mencionan las obras más importantes, o [eventos](#) dónde dedican mucho tiempo a observar una única obra de gran importancia.

### ***Ajustar la duración de la visualización de una obra***

- Con base en la investigación sobre el comportamiento de los visitantes del museo, se ha visto que el tiempo promedio requerido para ver una obra cambia según su complejidad que viene dada principalmente por su tamaño. Hay [estudios](#) de expertos que resaltan la importancia de las dimensiones de la obra a la hora de observarla.

### ***Segmentación de visitantes según interés y nivel de conocimiento***

- Se tiene el nivel de conocimiento artístico (principiante, intermedio, experto) de los visitantes, que [son un factor importante](#) en la observación de una obra. Los expertos podrán pasar más tiempo observando una obra que una persona que no sabe captar otras sutilezas. Esta información permite personalizar las recomendaciones y personalizar itinerarios para cada perfil.

### ***Filtrar obras según los deseos de los visitantes***

- Los expertos [categorizan](#) las características de las obras (autor, período, tema). En consecuencia, se ha creado un sistema que filtra automáticamente las obras que no son relevantes para los visitantes en base a las preferencias de cada visitante. Por ejemplo, los visitantes interesados en el arte impresionista no se les añadirá a la ruta obras modernistas.

### ***Pautas de inclusión para diferentes grupos***

- Se tienen en cuenta las características únicas de cada grupo, ya sea tamaño u otras características como por ejemplo, el número de niños, ya que también son factores que influyen en el tiempo de visita de un museo.

Esta información se obtiene de fuentes reconocidas como museos o universidades, lo que refuerza la validez y eficacia de las decisiones implementadas en el sistema.

## 2.4. Descripción informal del proceso de resolución

Primero se imprimen por pantalla las preguntas para que el usuario pueda rellenarlas. Algunas preguntas son de elección única, donde podrá seleccionar una opción de las que aparezcan por pantalla, otras para que el usuario pueda elegir múltiples elecciones que se le mostrarán por pantalla. En caso de que el usuario introduzca un dato erróneo, se le volverá a preguntar. Una vez se tenga caracterizado al usuario se pasará al siguiente apartado.

Con los datos de la parte anterior se pasa al procesado, donde se quitan de la base de datos todas las obras tales que el usuario no tiene interés. En otras palabras, si la obra contiene una o más de las preferencias del visitante esta no se eliminará de la base de datos. En caso contrario de que no tenga ninguna de las preferencias del visitante esta se eliminará de la base de datos. Para ello se detectarán las restricciones que tiene el usuario y se quitarán todas aquellas obras que no coincidan con alguna preferencia del usuario. Después de haber filtrado todas estas obras se pasa al apartado de construcción.

En esta parte se construirá el itinerario. El primer paso para ello será construir un itinerario completamente aleatorio con las obras que han sobrevivido al filtrado. La construcción aleatoria seleccionará obras de una lista de posibles obras. Esta lista representará las obras factibles (obras que cumplan alguna de las preferencias del usuario, en caso de que tenga). Todas las obras serán factibles en caso de que no haya ninguna preferencia, dado que eso implicaría que en la fase anterior no se haya eliminado ninguna obra de la base de datos. Si hay pocas obras que cumplan con las preferencias, las obras se repetirían constantemente a lo largo del itinerario, por este motivo se ha decidido que cada vez que se inserte una obra se borre de la lista y se seleccione otra obra aleatoriamente para los siguientes días. En caso de que una lista se quede vacía, se volverá a llenar con las preferencias y, si se vacía demasiadas veces, se volverá a llenar con la lista completa de obras para evitar repetir las mismas obras demasiadas veces (aunque esto no evita la repetición al principio evitará que se repita la misma obra durante todos los días de visita).

Una vez se ha obtenido el itinerario aleatoriamente, se intenta mejorarlo poco a poco para que se acerque a las necesidades temporales del usuario, dado que como es aleatorio es posible que no le de tiempo de visitar todas las obras o que se quede muy corto de tiempo. Por ese mismo motivo se calcula un heurístico que nos diga cómo de bueno es el itinerario y se intentará ir cambiando obras una a una hasta que no pueda mejorar más. A la hora de hacer la mejora se dará más importancia a obras que cumplan más las preferencias del usuario y a obras que le dedique menor tiempo posible para así que en conjunto poder ver el máximo de obras posibles dependiendo de las características del visitante.

Finalmente, el último apartado es el de presentación. Con el itinerario ya generado, solo queda mostrar el itinerario al usuario para cada día de la visita; las obras que tiene que visitar, en qué orden y en qué sala están.

### 3. Formalización del problema

En este apartado se explicará detalladamente la ontología, tanto como se ha construido y todos sus conceptos y atributos.

#### 3.1. Desarrollo de la ontología

En esta práctica se ha logrado tener una ontología inicial bastante buena y bien construida, por lo tanto la diferencia entre la ontología inicial y la ontología final es prácticamente nula. De hecho la primera versión es prácticamente la misma que la versión final haciendo cambios de tipo pasar algunas clases hijas de otras a un simple atributo booleano dado que dichas clases no almacenan ninguna información relevante ni tenía ninguna relación importante en el sistema.

En resumen, se ha construido la ontología utilizando **Protegé** para luego exportar el resultado a **Clips**. Lo primero que se ha hecho es hacer un listado de los atributos y clases que guarden alguna información de nuestro problema, un listado parecido a como se explica en el siguiente apartado **“3.2 Documentación de la ontología”**. Lo que se ha hecho con este listado es recopilar la información útil que puede servir para resolver el problema y no dejarse ninguna información. Posteriormente se han creado las clases en **Protegé**, añadiendo las relaciones lógicas del problema que se explicaran en más en detalle en el siguiente apartado y finalmente se ha exportado la ontología a formato **“.ttl”**.

Para exportar la ontología se ha utilizado la librería *python owl2else* que permite transformar a **Clips** archivos de formato **“.ttl”** a formato **“.clp”**, utilizando el siguiente comando:

```
Unset
owl2clips --input ontologia.ttl --output ontologia.clp --format turtle
```

Este comando crea el archivo **“ontologia.clp”** a partir de **“ontologia.ttl”**. Que será el archivo que se utilizara en este proyecto.

## 3.2. Documentación de la ontología

En este apartado se verá en detalle tanto las clases de la ontología, como sus atributos y las relaciones entre las diversas clases.

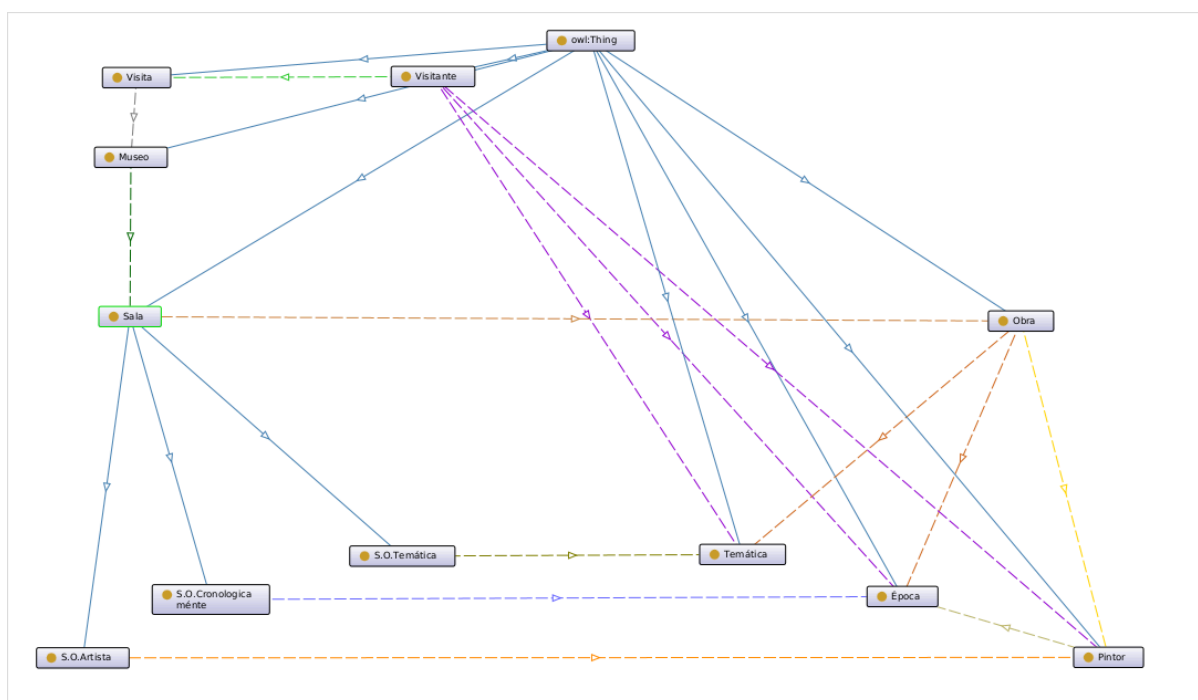


Fig. 1 Diagrama de la ontología

Los pocos cambios que se han realizado respecto a la ontología inicial han consistido principalmente en añadir atributos y relaciones nuevas para que se pueda manipular más fácilmente la información que hay.

El mayor cambio que se ha hecho fue básicamente eliminar dos subclases de una visita, dado que una visita podía ser en grupo o individual, la cosa que dichas clases no contenían ni información adicional relevante ni relaciones entre otras clases, por lo tanto se ha decidido finalmente poner un atributo booleano que decía si dicho grupo es individual o en grupo.

### 3.2.1 Clases

#### **Visitante:**

- Esta clase es la más importante y representa a un visitante o un grupo, básicamente es la clase que representa al usuario y la que guarda la información de las preguntas. En base a la información recolectada y almacenada en sus atributos el resultado final se verá moldeado para que se obtenga el mejor resultado posible acorde a los atributos de esta clase. Los atributos de esta clase son:

- **días\_de\_visita:** Representa el número de días en cual el visitante o grupo de visitantes realizarán dicha visita. (Integer)
- **duración:** Representa la cantidad de tiempo que se tiene disponible cada día, para facilitar el sistema se tiene en cuenta que cada día hay la misma cantidad de tiempo, es decir, es la misma **duración** para todos los **días\_de\_visita**. (Integer)
- **número\_personas:** Representa la cantidad de personas que hay en el grupo, básicamente esta clase puede representar o un visitante o un grupo de visitantes en base a este atributo. (Integer)
- **niños:** Representa la cantidad de niños que hay en el grupo, es decir, cuántos niños hay entre **número\_personas**. (Integer)
- **preferencia\_de\_temática:** Representa el nombre de la preferencia que tiene el visitante o visitantes con las temáticas, no es la relación sino el nombre de la clase a la que está relacionada. La utilidad de este atributo es para facilitar las funciones. (String)
- **preferencia\_artista:** Representa el nombre de la preferencia que tiene el visitante o visitantes con los pintores, no es la relación sino el nombre de la clase a la que está relacionada. La utilidad de este atributo es para facilitar las funciones. (String)
- **preferencia\_de\_epoca:** Representa el nombre de la preferencia que tiene el visitante o visitantes con las épocas, no es la relación sino el nombre de la clase a la que está relacionada. La utilidad de este atributo es para facilitar las funciones. (String)

Las relaciones entre clases son las siguientes:

- **acude:** Un visitante o grupo de visitantes está relacionado con una visita, dado que estas personas acuden a una visita. (Relación Verde Fig 1 entre Visitante-Visita)
- **prefiere:** Un visitante o grupo de visitantes está relacionado con una Época, Temática o una Pintor dado que estas personas tienen alguna preferencia. (Relación Morado Fig 1 entre Visitante-Época, Visitante-Temática, Visitante-Pintor)

### **Visita:**

- Esta clase representa una visita. Básicamente contiene la información de a qué museo está asociado y la información relevante. Esta clase no tiene atributos, simplemente es una clase que es importante por sus relaciones para facilitar la lectura y estructura de la ontología, en otras palabras, sirve como un nexo para vincular a los visitantes con los museos y las salas correspondientes, permitiendo una representación clara de las interacciones entre estos elementos. La relaciones que tiene esta clase es:
  - **se\_realiza\_en**: Representa la relación de un visita y un museo. Dado que una visita se realiza en un museo. (Relación Marrón Fig 1 entre Visita y Museo)

### **Museo:**

- Representa la instancia de un Museo, dicha clase es útil dado que en algún sitio se tiene que almacenar las obras. Las obras pertenecen a un Museo en el cual se expone. Esta clase es esencial dentro de la ontología, dado que proporciona una estructura lógica para la organización de los elementos exhibidos. Los atributos de esta clase son los siguientes:
  - **nombre**: Representa el nombre de dicho Museo, es el atributo que permite identificar a un Museo. (String)

Las relaciones de esta clase son las siguientes:

- **tiene**: Representa la relación de un Museo y una Sala, dado que un museo puede contener diferentes salas. Esta relación refleja cómo un museo puede contener múltiples salas. (String)

### **Sala:**

- La clase **Sala** representa el espacio dentro del museo que puede tener diferentes obras. Es una clase central para estructurar donde están las obras. Sus atributos son los siguientes:
  - **número**: Representa un identificador único o un número asignado a la sala dentro del museo. (Integer)

Las relaciones de esta clase son las siguientes (incluyendo subclases):

- **contiene**: Almacena las obras contenidas dentro de una sala específica. Esto permite gestionar qué elementos están disponibles en dicha sala. (Relación Fig 1 de Naranja entre Sala y Obra)
- **Subclase S.O.Cronológicamente**: Básicamente dicta que es una subclase de Sala, en otras palabras, la superclase de S.O.Cronológicamente es la clase Sala. (Relación Azul Fig 1 entre Sala y S.O.Cronológicamente)

- **Subclase S.O.Temática:** Básicamente dicta que es una subclase de Sala, en otras palabras, la superclase de S.O.Temática es la clase Sala. (Relación Azul Fig 1 entre Sala y S.O.Temática)
- **Subclase S.O.Artista:** Básicamente dicta que es una subclase de Sala, en otras palabras, la superclase de S.O.Artista es la clase Sala. (Relación Azul Fig 1 entre Sala y S.O.Artista)

### **S.O.Artista:**

- La clase **S.O.Artista** representa una sala organizada exclusivamente con obras de un único artista famoso. Es una especialización de la clase Sala. Esta clase no tiene atributos y las relaciones son las siguientes:
  - **organizado\_según:** Indica al artista por el cual se ha organizado esta sala. Esto asegura que solo se incluyan obras de dicho artista. (Relación Amarilla Fig 1 entre S.O.Artista y Pintor)

### **S.O.Cronológicamente:**

- La clase **S.O.Cronológicamente** representa una sala organizada en un orden cronológico, destacando el progreso histórico de las obras exhibidas. Es una clase sin atributos y es una subclase de Sala, donde las relaciones son las siguientes:
  - **organizado\_por:** Especifica el criterio cronológico utilizado para organizar las obras. (Relación Azul Fig 1 entre S.O.Cronológicamente y Época)

### **S.O.Temática:**

- La clase **S.O.Temática** modela una sala donde las obras están organizadas por temáticas específicas. Es una subclase de Sala que no tiene atributos, donde sus relaciones son las siguientes:
  - **organizada\_por:** Define la temática que se utiliza como criterio de organización. (Relación Marron Fig 1 entre S.O.Temática y Temática)

### **Obra:**

- La clase **Obra** modela una instancia específica de una obra de arte, con sus respectivos atributos y relaciones que permiten identificarla y contextualizarla, es el corazón del problema dado que es el motivo de la visita de los visitantes. Sus atributos son:
  - **área:** Representa cómo de grande es la pintura, para cálculos posteriores el área es de gran utilidad dado que se ha interpretado que cuanto más grande es el área más compleja es la pintura dado que aparecen más objetos. (Integer)
  - **año\_de\_creación:** Representa el año en que fue creada la obra. (Integer)
  - **nombre:** Representa el nombre o título de la obra. (String)
  - **relevancia:** Representa un valor o peso asignado a la obra, indicando su importancia dentro de una colección. (Integer)

La relaciones de esta clase son las siguientes:

- **dibujada\_por:** Relacionar la obra con el pintor que la creó. Dado que una obra es dibujada por alguien. (Relación Amarilla Fig 1 entre Obra y Pintor)
- **pertenece\_a:** Relaciona la obra con una época específica. Dado que una obra pertenece a una época en concreto. (Relación Naranja Fig 1 entre Obra y Época)
- **pertenece\_a\_la:** Relaciona la obra con la temática a la que pertenece, dado que cada obra pertenece a una temática determinada. (Relación Naranja Fig 1 entre Obra y Temática)

### **Pintor:**

- La clase **Pintor** representa a un pintor reconocido o famoso y contiene su información relevante, sus atributos són los siguientes:
  - **nacionalidad:** Representa la nacionalidad de un pintor famoso. (String)
  - **nombre:** Representa el nombre de dicho pintor famoso. (String)



Sus relaciones son las siguientes:

- **es\_de:** Relaciona a un pintor con una época específica. (Relación Granate Fig 1 entre Pintor y Época)

### **Temática:**

- La clase **Temática** define el concepto temático de una obra o sala, permitiendo clasificar y filtrar elementos por categorías específicas, esta clase no tiene relaciones dado que el sentido de esta clase es que las obras son de una temática en específico y hay salas organizadas por temáticas. Los atributos de dicha clase son:
  - **nombre:** Especifica el nombre de la temática. (*String*)

### **Época:**

- La clase **Época** representa un período histórico según criterios cronológicos o estilísticos. Sus atributos son:
  - **inicio:** Especifica el año de inicio del periodo histórico. (*Integer*)
  - **fin:** Define el año de finalización del periodo histórico. (*Integer*)
  - **nombre:** Representa el nombre de la época. (*String*)

### 3.3. Proceso de razonamiento y metodología de resolución

En la resolución del problema de planificación de itinerarios, partimos de una base en la que disponemos de un conjunto de obras que hay en todo el museo guardadas en memoria. Luego de hacer las preguntas en memoria solo se quedarán aquellas obras que tengan al menos una de las preferencias, en otras palabras, aquellas obras que no tengan absolutamente ningún interés para el visitante se eliminarán de la memoria. Posterior a este gran filtrado en memoria solo quedarán aquellas obras que han sido previamente seleccionadas según sus preferencias, lo que garantiza que no se consideren obras irrelevantes o que no cumplan con las necesidades del visitante. El objetivo principal es, por tanto, generar un itinerario que maximice el aprovechamiento del tiempo disponible, y no tendría sentido malgastar el tiempo en obras irrelevantes para los visitantes.

Dado que este es un problema de síntesis, la resolución se centra en construir una solución inicial (en este caso, un itinerario) que sea funcional y, posteriormente, mejorarlo iterativamente hasta alcanzar un resultado óptimo. Este problema puede dividirse en tres tareas principales:

1. **Especificación:** Identificar y definir las restricciones que deben cumplirse, tales como tiempo máximo disponible, intereses del visitante y tamaño del grupo.
2. **Diseño:** Generar una configuración inicial del itinerario que cumpla con las restricciones especificadas.
3. **Ensamblaje:** Construir un sistema final que integre los elementos seleccionados, optimizando las preferencias y características del visitante.

Para abordar este problema de síntesis, se utiliza un enfoque más de resolución constructiva con una estrategia de mínimo compromiso, es decir, se parte de una solución inicial completa y se va refinando progresivamente, donde se tiene que asegurar de cumplir con las restricciones establecidas y priorizando las preferencias del visitante, es decir, utilizar una heurística. En lugar de aplicar el operador con menor impacto sobre las restricciones, se ha optado por decisiones que acerquen la solución a un itinerario óptimo, dado que las restricciones principales ya se gestionaron en etapas previas del proceso.

La solución inicial se construye de forma sencilla y secuencial, llenando un itinerario con obras seleccionadas aleatoriamente de una lista con obras que tienen alguna de las preferencias del visitante, al añadir una obra en el itinerario se elimina de la lista de esta manera se evita las repeticiones de obras. Así, se garantiza un itinerario funcional que maximiza respetar las preferencias del usuario.

Es importante señalar que la asignación inicial incluye cierta aleatoriedad. Esta solución inicial, aunque funcional, probablemente no cumplirá completamente con las necesidades del visitante, es decir, que no sea la mejor distribución optimizando el tiempo. Por lo tanto, al tener este itinerario inicial aleatorio habría que mejorarlo en términos temporales. Por lo tanto, el objetivo será mejorarla iterativamente.

## 4. Implementación del problema

### 4.1. Ontología

#### 4.1.1 Construcción de la Ontología

Para esta construcción de la ontología se ha usado la herramienta Protegé. Se guardó la ontología en formato .ttl y luego se ha exportado a CLIPS con la ayuda de owl2clips. Finalmente, se ha incluido la ontología exportada al fichero.

#### 4.1.2 Razonamiento de la solución

A continuación se verá el desarrollo específico de los distintos módulos del sistema. La entrada y la salida son bastante triviales, todo programa va a recibir una determinada entrada y va a producir una determinada salida. En este proyecto se han separado estos módulos para facilitar el procesamiento de los datos y la construcción de la solución y para separar el procesamiento inicial y final de los datos (básicamente transformaciones de formato) del procesamiento real que se hará en los otros 2 módulos.

Respecto a los otros 2 módulos, el módulo de procesamiento-obras se basa en un par de reglas y una función de filtrado (todas ellas con la misma prioridad) y el módulo de síntesis se basa en un par de reglas con distintas prioridades (se tienen que ejecutar en un orden específico) y unas funciones que calculan el objetivo, generalmente una solución aleatoria y mejora la solución inicial hasta que no pueda más.

Además se ha aprovechado de que CLIPS prueba una misma regla con todas las instancias posibles.

### 4.2. Módulos

#### 4.2.1. Módulo main

La función principal de este módulo es imprimir un mensaje inicial anunciando el inicio del programa y declarar el orden en el que se ejecutarán los módulos. Esto se realiza con la siguiente regla:

##### Regla **initialRule**

- **Propósito:** Mostrar un mensaje introductorio al inicio del programa y declarar el orden en el que se ejecutarán los módulos.
- **Condiciones:** Ninguna condición.
- **Acciones:**
  - Muestra el mensaje inicial del programa.
  - Establece el orden siguiente de los módulos y entra al primer módulo:
    - recopilacion-informacion-visitante, procesamiento-obras, síntesis

## 4.2.2. Módulo recopilación

### Introducción

El módulo de recopilación es responsable de recolectar información sobre los visitantes del museo para facilitar el procesamiento en módulos posteriores. Este módulo utiliza un *template* llamado `informacion-visitante`.

### Descripción del Template

El *template* `informacion-visitante` organiza la información del visitante en los siguientes *slots*:

1. **`numero-personas`**  
Número total de personas en el grupo visitante. Valor por defecto: `-1`.
2. **`numero-niños`**  
Cantidad de niños en el grupo. Valor por defecto: `-1`.
3. **`dias-visita`**  
Días previstos para la visita. Valor por defecto: `-1`.
4. **`duracion-visita`**  
Duración estimada en horas. Valor por defecto: `-1`.
5. **`conocimiento`**  
Nivel de conocimiento sobre el museo (escala definida). Valor por defecto: `-1`.
6. **`preferencia-tematica`**  
Temáticas preferidas. Es un multislot.
7. **`preferencia-artista`**  
Artistas preferidos. Es un multislot.
8. **`preferencia-epoca`**  
Épocas o periodos artísticos de interés. Es un multislot.

### Función del Template

El *template* `informacion-visitante` centraliza la información proporcionada por los visitantes para que otros módulos del sistema puedan consultarla y utilizarla en la generación del itinerario solución. Los valores por defecto aseguran que los slots tengan un estado inicial, incluso si algunos datos aún no han sido ingresados.

## Funciones definidas para el módulo

Estas funciones están diseñadas para interactuar con el usuario, recolectar datos específicos y asegurar que las respuestas cumplan con las restricciones esperadas antes de ser procesadas por las reglas del sistema.

### Función **pregunta-numerica**

- **Propósito:** Solicitar al usuario un número dentro de un rango especificado.
- **Definición:** `(deffunction MAIN::pregunta-numerica (?pregunta ?rangini ?rangfi)`
- **Parámetros:**
  1. `?pregunta`: Mensaje de la pregunta que se presentará al usuario.
  2. `?rangini`: Límite inferior del rango permitido.
  3. `?rangfi`: Límite superior del rango permitido.
- **Funcionamiento:**

Muestra la pregunta con el rango permitido y solicita una entrada. Si la respuesta no está dentro del rango, se repite la pregunta hasta obtener una respuesta válida.
- **Retorno:** El valor numérico ingresado por el usuario dentro del rango.

### Función **pregunta-multiSelect**

- **Propósito:** Permitir al usuario seleccionar una opción válida de una lista de valores.
- **Definición:** `(deffunction MAIN::pregunta-multiSelect (?pregunta $?valores-validos)`
- **Parámetros:**
  1. `?pregunta`: Mensaje de la pregunta que se presentará al usuario.
  2. `?$valores-validos`: Lista de valores permitidos para la respuesta.
- **Funcionamiento:**

Muestra la pregunta y los valores válidos. Solicita una entrada y valida si la respuesta pertenece a la lista de valores permitidos o es la palabra "fin". Si no es válida, repite la pregunta hasta obtener una respuesta correcta.
- **Retorno:** El valor seleccionado por el usuario que está dentro de los valores permitidos.

## Reglas del modulo

Lo siguiente son las reglas del módulo de recopilación, su objetivo es interactuar con el usuario para recolectar los datos del visitante.

### Regla **mensaje-inicial**

- **Propósito:** Mostrar un mensaje introductorio al inicio de la recopilación de información.
- **Condiciones:** No debe existir una instancia de `informacion-visitante`.
- **Acciones:** Muestra un mensaje indicando que se iniciará la recopilación de datos

---

### Regla **establecer-numero-personas**

- **Propósito:** Solicitar el número de personas en el grupo visitante.
  - **Acciones:**
    - Utiliza la función **pregunta-numerica** para solicitar un valor entre 1 y 50.
    - Crea o actualiza una instancia de **informacion-visitante** con el valor proporcionado en el slot **numero-personas**.
- 

### Regla **establecer-numero-niños**

- **Propósito:** Solicitar el número de niños en el grupo visitante.
  - **Condiciones:** El valor actual del slot **numero-niños** debe ser menor a 0.
  - **Acciones:**
    - Usa **pregunta-numerica** para solicitar un número entre 1 y el total de personas del grupo.
    - Actualiza el slot **numero-niños**.
- 

### Regla **establecer-dias-visita**

- **Propósito:** Solicitar la cantidad de días que se planean dedicar a la visita.
  - **Condiciones:**
    - El slot **dias-visita** debe tener un valor menor a 0.
  - **Acciones:**
    - Solicita un valor entre 1 y 7 utilizando **pregunta-numerica**.
    - Actualiza el slot **dias-visita**.
- 

### Regla **establecer-duracion-visita**

- **Propósito:** Preguntar la duración promedio de la visita por día en minutos.
  - **Condiciones:**
    - El slot **duracion-visita** debe ser menor a 0.
  - **Acciones:**
    - Solicita un valor entre 1 y 720 minutos (12h).
    - Actualiza el slot **duracion-visita**.
- 

### Regla **establecer-preferencia-tematica**

- **Propósito:** Preguntar la temática preferida del visitante.
- **Condiciones:**
  - El slot **preferencia-tematica** no debe de contener el valor "fin".
- **Acciones:**
  - Usa **find-all-instances** para obtener una lista de temáticas disponibles.

- Solicita una respuesta con `pregunta-multiSelect`.
- Añade al slot `preferencia-tematica` la respuesta.

---

#### Regla `establecer-preferencia-artista`

- **Propósito:** Preguntar al visitante por su artista favorito.
- **Condiciones:**
  - El slot `preferencia-artista` no debe de contener el valor "fin".
- **Acciones:**
  - Obtiene una lista de artistas disponibles con `find-all-instances`.
  - Solicita una selección usando `pregunta-multiSelect`.
  - Añade al slot `preferencia-artista` la respuesta..

---

#### Regla `establecer-preferencia-epoca`

- **Propósito:** Solicitar al visitante que indique su época o periodo artístico preferido.
- **Condiciones:**
  - El slot `preferencia-epoca` no debe de contener el valor "fin".
- **Acciones:**
  - Usa `find-all-instances` para obtener una lista de épocas disponibles.
  - Solicita una selección con `pregunta-multiSelect`.
  - Añade al slot `preferencia-epoca` la respuesta..

---

#### Regla `establecer-conocimiento`

- **Propósito:** Evaluar el nivel de conocimiento artístico del visitante.
- **Condiciones:**
  - El slot `conocimiento` debe ser menor a 0.
- **Acciones:**
  - Solicita un valor entre 1 y 100 usando `pregunta-numerica`.
  - Actualiza el slot `conocimiento`.

---

#### Regla `imprimir-informacion-visitante`

- **Propósito:** Imprimir en consola los datos recopilados del visitante
- **Condiciones:**
  - Todos los slots del template `informacion-visitante` tienen valores asignados.
- **Acciones:**
  - Imprime una separación visual con líneas.
  - Muestra un mensaje de encabezado: "Datos del visitante:".
  - Imprime, en líneas separadas, los valores asignados a cada slot de `informacion-visitante`.

### 4.2.3. Módulo procesamiento

Las reglas de este módulo son las siguientes:

#### Regla **filtrar-obras-sin-preferencia**

- **Propósito:** Eliminar las obras para las que el usuario no tenga ninguna preferencia.
- **Acciones:**
  - Obtiene una obra con `?obra <- (object (is-a Obra))`.
  - Comprueba que la obra pertenezca a las preferencias usando `intersection$`.
  - Si no pertenece a ninguna, la elimina de la lista de obras usando una función del módulo, `eliminar-obra`.

#### Regla **print-remaining-obras**

- **Propósito:** Imprimir todas las obras después del filtrado.
- **Acciones:**
  - Obtiene la lista de obras restantes con `find-all-instances`.
  - Si no hay ninguna obra en la lista, lo escribe por pantalla.
  - Si hay obras en la lista, las escribe por pantalla, recorriendo la lista usando `foreach`.

Las funciones de este módulo son las siguientes:

#### Función **eliminar-Obra**

- **Propósito:** Eliminar una obra de la lista.
- **Definición:** `(deffunction procesamiento-obras::eliminar-Obra (?Obra))`
- **Parámetros:**
  1. `?Obra`: Obra a eliminar de la lista.
- **Funcionamiento:** Muestra el nombre de la obra a eliminar y seguidamente la elimina.
- **Retorno:** Nada.



#### 4.2.4. Módulo síntesis

Este módulo se encarga de utilizar y calcular la heurística. También imprime los resultados con el uso de una función.

##### Función **ImprimirInfoObra**

- **Propósito:** Imprimir información sobre una obra específica y la sala en la que se encuentra.
- **Definición:** (deffunction sintesis::ImprimirInfoObra (?obra ?conocimiento ?num-personas ?num-niños)
- **Parámetros:**
  1. **?obra:** Obra cuya información se desea imprimir. Debe ser una instancia válida de la clase **Obra**.
  2. **?conocimiento ?num-personas ?num-niños :** son parámetros para calcular el tiempo que necesitan los visitantes para ver la obra
- **Funcionamiento:**
  1. Obtiene todas las instancias de la clase Sala.
  2. Itera sobre la lista de instancias para determinar si alguna contiene la obra especificada.
  3. Cuando se encuentra la sala que contiene la obra:
    - Obtiene el nombre de la obra utilizando get-nombre.
    - Obtiene el número de la sala utilizando get-número.
    - Obtiene El tiempo que necesitan los visitantes para ver la obra con la función **calcular-tiempo-obra**
    - Imprime la siguiente información:
      - El nombre de la obra.
      - La sala correspondiente.
      - El número de sala asociado.
      - El tiempo que necesitan los visitantes para ver la obra

## **Función: calcular-tiempo-obra**

### **Propósito Fundamental:**

La función **calcular-tiempo-obra** evalúa y devuelve el tiempo estimado necesario para que un grupo de visitantes pueda observar una obra específica en el museo. Este cálculo se basa en múltiples factores relacionados con las características de la obra, la composición del grupo visitante y el nivel de conocimiento de los visitantes sobre la obra, donde a cada variable se le dará una ponderación entre (0, 1), cuanto más cercana a 1 más importancia se le da a esa variable en el cálculo del tiempo, es decir, que tardará más.

### **Parámetros de Entrada:**

1. **?obra:**
  - Representa la instancia de la obra que se está analizando.
  - Proporciona información relevante como el área física y la relevancia de la obra (obtenida mediante métodos como **get-area** y **get-relevancia**).
2. **?conocimiento:**
  - Nivel de conocimiento que el grupo tiene sobre arte.
  - Este valor se normaliza en una escala del 1 al 10 para calcular su impacto en el tiempo requerido.
3. **?num-personas:**
  - Total de personas (incluyendo a niños) en el grupo visitante.
4. **?num-niños:**
  - Número de niños en el grupo visitante.
  - Afecta el tiempo estimado con un peso menor que el de los adultos, reflejando que los niños tienden a interactuar con la obra de manera diferente.

### **Algoritmo de Cálculo**

El tiempo estimado se determina a través de un modelo polinómico que pondera los diferentes factores de influencia. La fórmula completa de cálculo es:

$$T = 0.5 \times (1 + (0.8 \times R) + (0.004 \times A) + (1 + 0.15 \times \frac{C}{10}) + (1 + 0.08 \times P) + (1 + 0.05 \times N))$$

**Fig 2 Fórmula para calcular el tiempo de una obra**

### Detalles del Algoritmo:

1. **Área de la Obra ?area (A):**
  - Multiplicador base: 0.004.
  - La magnitud del multiplicador es pequeña para compensar la magnitud del área, que está en cm<sup>2</sup>.
  - Representa el impacto del tamaño físico de la obra en el tiempo requerido para observar.
  - A mayor área, mayor será el tiempo estimado, dado que se supone que cuanto más área más elementos aparecen en el.
2. **Relevancia de la Obra ?relevancia (R):**
  - Factor de ponderación: 0.8.
  - Ajusta el tiempo estimado según la importancia intrínseca de la obra.
  - Una obra con mayor relevancia requiere más tiempo para ser apreciada.
3. **Conocimiento del Visitante ?conocimiento (C):**
  - Peso significativo: 0.15.
  - Representa cómo el nivel de conocimiento del grupo influye en la profundidad de su interacción con la obra.
  - Se normaliza dividiendo por 10, con mayor conocimiento resultando en un aumento proporcional del tiempo estimado.
4. **Composición del Grupo:**
  - **Adultos ?num-personas (P):**
    - Peso asignado: 0.08.
    - Indica que un grupo mayor de personas incrementa ligeramente el tiempo necesario debido a la necesidad de acomodar interacciones y discusiones.
  - **Niños ?num-niños (N):**
    - Peso asignado: 0.05.
    - Refleja que los niños también afectan la duración, aunque en menor medida que los adultos.
5. **Constante Base:**
  - Se multiplica todo el cálculo por una constante base de 0.5, representando un ajuste para normalizar los tiempos estimados en base al tiempo que se tarda en apreciar una obra de arte.

### Funcionamiento General:

1. La función utiliza la obra proporcionada para extraer las propiedades relevantes: el área y la relevancia.
  - Estas propiedades son recuperadas usando `get-area` y `get-relevancia`.
2. Evalúa el nivel de conocimiento del visitante (`?conocimiento`) y lo incluye en el cálculo ponderado.
3. Agrega los efectos individuales de los adultos (`?num-personas`) y niños (`?num-niños`) presentes en el grupo.
4. Se aplica la fórmula de la Fig 2.

5. Devuelve el tiempo total estimado (**?time**) en minutos, considerando la suma ponderada de los factores antes mencionados.

### Función: **distribucion-inicial**

#### Propósito Fundamental:

La función **distribucion-inicial** tiene como objetivo distribuir un conjunto de obras de forma aleatoria pero balanceada entre los días de visita planificados, con el objetivo que en cada día tengan el mismo número de obras y que no todas las obras sean visitadas el primer día o los primeros días. Este algoritmo es útil para generar una asignación inicial estructurada, que puede ser refinada posteriormente en procesos de optimización. El enfoque combina **aleatoriedad controlada** para evitar sesgos y **estructura uniforme** para garantizar que ningún día quede sobrecargado o vacío.

#### Parámetros de Entrada:

1. **?obras:**
  - Lista de todas las obras disponibles para la distribución.
  - Cada elemento representa una obra que será asignada a uno de los días disponibles.
2. **?dias-visita:**
  - Número total de días planificados para realizar las visitas.
  - Asimismo, representa la cantidad máxima de días en los que se distribuirán las obras.

## Detalles del Algoritmo:

### Inicialización de Variables:

La función comienza estableciendo un conjunto de variables clave:

1. **?distribucion:**
  - Lista vacía que almacenará el resultado final de la distribución.
  - Formato esperado: ((obra-1 dia-1) (obra-2 dia-2) ...).
2. **?puntero:**
  - Índice inicial para seleccionar obras aleatoriamente.
  - Comienza en 1 y se ajusta dinámicamente según las obras restantes.
3. **?dia:**
  - Contador que indica el día al que se están asignando obras en un momento dado.
  - Comienza en 1 y aumenta de forma secuencial hasta alcanzar el valor de ?dias-visita.
4. **?obras-dia:**
  - Contador que rastrea cuántas obras han sido asignadas al día actual.
  - Se reinicia a 0 cada vez que se cambia al siguiente día.
5. **?max-obras-dia:**
  - Número máximo de obras permitidas por día, calculado mediante:

$$max-obras-dia = \lfloor \frac{longitud\ ?obras}{?dias-visita} \rfloor + 1$$

- Esta fórmula asegura una distribución uniforme, con un margen adicional para manejar casos en los que el número total de obras no sea divisible exactamente por los días, además el número de obras viene dado implícitamente por la longitud de la lista de ?obras.

## Funcionamiento General:

### 1. Selección Aleatoria de Obras:

- Utiliza la función random para seleccionar un índice dentro del rango actual de la lista de obras.
- La obra seleccionada se extrae mediante nth\$ y luego se elimina de la lista con delete\$ para evitar duplicados.

### 2. Asignación a un Día:

- La obra seleccionada se asigna al día actual (?dia) y se añade a la lista ?distribucion.
- Cada asignación tiene el formato (obra, dia).

### 3. Control de Distribución:

- Después de cada asignación, el contador **?obras-día** se incrementa.
- Si el contador alcanza el valor de **?max-obras-día**, la función cambia al siguiente día (**?día**) y reinicia el contador.

### 4. Finalización del Proceso:

- El ciclo continúa hasta que todas las obras han sido distribuidas, es decir, hasta que la longitud de la lista **?obras** sea igual a 0, al llegar a 0 y no haya más obras disponibles el proceso ha acabado y por lo tanto se retorna la lista **?distribucion**.

### Función: **evaluar-distribucion**

#### Propósito Fundamental:

La función **evaluar-distribucion** calcula un puntaje de calidad para una distribución de obras asignadas a días específicos, considerando restricciones de tiempo, balance entre días y cobertura completa de las obras. Este algoritmo permite evaluar si una distribución es eficiente y cumple con los objetivos del sistema de planificación, básicamente esta es la función en la cual se calcula el heurístico, donde se explicara con más detalle en el apartado “4.3 Elaboración del Heurístico”.

#### Parámetros de Entrada:

1. **?distribucion:**
  - Lista que contiene pares alternados de obras y los días asignados.
  - Ejemplo: (**obra1**, 1, **obra2**, 1, **obra3**, 2) asigna **obra1** y **obra2** al día 1, y **obra3** al día 2.
2. **?duracion-visita:**
  - Tiempo máximo permitido en minutos para cada día de la visita.
  - Representa el límite de tiempo diario asignado para observar las obras.
3. **?dias-visita:**
  - Número total de días disponibles para la visita.
4. **?conocimiento:**
  - Nivel de conocimiento del grupo visitante sobre las obras.
  - Afecta el tiempo estimado para observar cada obra.
5. **?num-personas:**
  - Cantidad total de personas en el grupo visitante.
  - Incluye adultos y niños, influenciando el tiempo de observación de las obras.
6. **?num-niños:**
  - Número de niños en el grupo visitante.
  - Se les asigna un peso menor en el cálculo del tiempo, reflejando su interacción más breve con las obras.

## Proceso de Evaluación:

### Fase 1: Inicialización del Sistema de Puntuación

- **Variables iniciales:**
  - **?total-score**: Inicializado en 0, este valor acumula la puntuación calculada.
  - **?rango-distribucion**: Lista que almacena el tiempo acumulado de observación para cada día. Inicializada con 0 para todos los días.
- **Preparación de tiempos por día:**
  - Se recorre la lista de días (**?dias-visita**) y se establece inicialmente que todos tienen 0 minutos asignados.

### Fase 2: Cálculo de Tiempos por Día

- **Recorrido por las obras:**
  - Para cada par de elementos en **?distribucion**:
    1. Se selecciona la obra (**?obra**) y el día asignado (**?dia**).
    2. Se calcula el tiempo requerido para observar la obra usando la función **calcular-tiempo-obra**, que considera:
      - Tamaño y relevancia de la obra.
      - Conocimiento del grupo visitante.
      - Composición del grupo (adultos y niños).
    3. Se acumula el tiempo estimado en el día correspondiente dentro del **?rango-distribucion**.
- **Penalización por exceso de tiempo:** se añade una penalización al **?total-score**

### Fase 3: Análisis de Distribución Temporal

- **Identificación de extremos:**
  - Se encuentran los tiempos máximos (**?max-tiempo**) y mínimos (**?min-tiempo**) entre los días registrados en **?rango-distribucion**.
- **Penalización por desequilibrio:**
  - Se aplica una penalización basada en la diferencia entre **?max-tiempo** y **?min-tiempo**

### Fase 4: Cobertura de Obras

- **Verificación de asignaciones completas:**
  - Si la longitud de **?distribucion** es menor que el doble del número total de obras disponibles
  - Esto garantiza que todas las obras sean consideradas en la distribución.

### Fase Final: Retorno del Puntaje Total

- El puntaje final (**?total-score**) combina penalizaciones por:
  1. Excesos de tiempo diario.
  2. Desequilibrios entre días.
  3. Falta de cobertura de obras.
- Se retorna **?total-score**, representando la calidad de la distribución propuesta.

### Funcionamiento General:

1. Inicializa el sistema de evaluación con tiempos y variables en 0.
2. Recorre la distribución para calcular tiempos acumulados por día.
3. Aplica penalizaciones por exceso y desequilibrio temporal.
4. Verifica cobertura completa de obras y aplica penalizaciones si corresponde.
5. Retorna la puntuación final como medida de calidad de la distribución.

### Función: hill-climbing

#### Propósito Fundamental:

La función **hill-climbing** aplica un algoritmo de optimización local para encontrar la mejor distribución de obras en un número limitado de días, maximizando una puntuación basada en criterios predefinidos (como balance temporal y respeto a las restricciones de duración). Este método busca mejorar iterativamente una solución inicial hasta alcanzar un punto en el que no es posible mejorar más, considerado como el óptimo local.

#### Parámetros de Entrada:

1. **?obras:**
  - Representa la lista de todas las obras que deben ser distribuidas.
  - Contiene instancias que incluyen características como relevancia y duración estimada.
2. **?duracion-visita:**
  - Tiempo máximo disponible por día para visitar obras.
  - Se utiliza como restricción para penalizar distribuciones que exceden este límite.
3. **?dias-visita:**
  - Número total de días disponibles para distribuir las obras.
  - Determina el rango de posibles asignaciones diarias.
4. **?conocimiento:**
  - Nivel de conocimiento del grupo sobre arte, en una escala normalizada.
  - Afecta el tiempo requerido para observar cada obra.
5. **?num-personas:**
  - Número total de personas en el grupo visitante.
  - Influye en el tiempo de observación de cada obra.
6. **?num-niños:**
  - Número de niños en el grupo visitante.
  - Tiene un peso menor que los adultos en la estimación del tiempo requerido.



## Funcionamiento General:

### 1. Generación de la Solución Inicial:

- Se genera una **distribución inicial** llamando a la función `distribucion-inicial`, que asigna obras de manera uniforme o heurística en los días disponibles.
- La calidad de esta distribución se evalúa con `evaluar-distribucion`, devolviendo una puntuación inicial `?mejor-score`.

### 2. Proceso Iterativo de Optimización:

- El algoritmo emplea un bucle `while` que continúa ejecutándose mientras se encuentre una **mejora** en la distribución actual:
  - **Exploración de Vecinos:**
    - Se generan distribuciones vecinas modificando un día asignado a una obra o eliminando asignaciones.
    - Cada vecino es evaluado usando `evaluar-distribucion`.
  - **Comparación de Puntajes:**
    - Si la puntuación del vecino supera la puntuación de la distribución actual (`?mejor-score`), se actualiza la mejor distribución (`?mejor-distribucion`).
  - **Criterio de Mejora:**
    - Si al menos una mejora ocurre, el algoritmo continúa explorando. Si no, termina el proceso.

### 3. Exploración de Eliminaciones:

- El algoritmo evalúa distribuciones vecinas donde se eliminan obras no esenciales.
- Si alguna distribución resultante tiene un puntaje mejor, se actualiza como la nueva mejor distribución.

### 4. Terminación:

- Cuando no se encuentran mejoras adicionales, el algoritmo concluye imprimiendo que la distribución es óptima y retornando esta distribución.

### Regla **organizar-e-imprimir-obras**

- **Propósito:** Repartir las obras en distintos días, asegurando que no excedan el tiempo permitido, utilizando las anteriores funciones. Después imprime las obras, repartidas en los distintos días.
- **Acciones:**
  - Obtiene los atributos relevantes del visitante.
  - Obtiene una lista de obras con **find-all-instances**.
  - Inicializa las variables **?total-time**, **?selected-obras** y **?dia**.
  - Obtiene **?distribucion-optimizada** pasando los parametros descritos anteriormente.
  - Si hay obras en la lista, recorre la lista usando **foreach**, imprimiendo por pantalla la información de la obra con el uso de la función **ImprimirInfoObra**.

### 4.3. Elaboración del heurístico

El heurístico se divide en dos partes: el cálculo del tiempo, que es basado en el conocimiento, y la distribución de obras en distintos días. El heurístico diseñado para evaluar la distribución de obras en días específicos combina distintos factores que influyen en la calidad de la planificación. Su objetivo principal es proporcionar una métrica que refleje el equilibrio, la eficiencia y la cobertura completa en la asignación de obras, penalizando desviaciones significativas de estos criterios.

El cálculo del heurístico, representado por una puntuación total (total-score) , se fundamenta en la suma de penalizaciones derivadas de excesos de tiempo diario, desequilibrios entre días y la falta de cobertura completa de las obras asignadas. Formalmente, la fórmula matemática que define el heurístico es:

$$total-score = - \sum_{i=1}^n P_{tiempo}(d_i) - P_{desequilibrio} - P_{cobertura}$$

**Fig 3 Fórmula matemática del heurístico**

$P_{tiempo}(d_i)$  es la penalización asociada a los excesos de tiempo diario para el día  $d_i$ .

$P_{desequilibrio}$  es la penalización por la diferencia entre los tiempos asignados al día con la mayor carga y el día con la menor carga.

$P_{cobertura}$  es la penalización por la falta de asignación de todas las obras disponibles en la distribución.

El heurístico considera los siguientes aspectos clave:

1. **Tiempo de observación diario:** Se acumula el tiempo estimado requerido para observar cada obra asignada a un día específico, calculado mediante una función que incluye factores como la relevancia de la obra, el tamaño de su área, el nivel de conocimiento del grupo visitante, y la composición del grupo en términos de adultos y niños. Si el tiempo total de un día excede el límite permitido, se aplica una penalización proporcional al exceso.
2. **Equilibrio entre días:** Para fomentar una distribución uniforme de las obras, se penalizan las diferencias significativas entre el día con mayor tiempo acumulado (max) y el día con menor tiempo acumulado (min), mediante una penalización proporcional al desequilibrio, calculada como:

$$P_{desequilibrio} = k_{des}(max - min)$$

Donde  $k_{des}$  es un factor de ponderación para ajustar la sensibilidad a desequilibrios.

**Cobertura completa de obras:** Se verifica que todas las obras disponibles hayan sido asignadas en la distribución. Si el número de obras asignadas es menor al número total requerido, se aplica una penalización proporcional al número de obras no consideradas.

El uso de este heurístico se justifica por su capacidad para reflejar de manera integral los objetivos del sistema de planificación. Al balancear múltiples criterios de evaluación en una única métrica, el heurístico facilita el proceso de optimización, permitiendo identificar rápidamente distribuciones que cumplen con las restricciones y optimizan el uso del tiempo disponible.

Además, al ser computacionalmente eficiente, este heurístico se integra de manera efectiva en algoritmos iterativos como *hill-climbing*, que evalúan y refinan soluciones mediante comparaciones basadas en esta métrica. Esto resulta en una herramienta robusta para la toma de decisiones en entornos donde las restricciones temporales y logísticas son críticas.

## 4.4 Notas sobre las instancias

Para inicializar el programa se ha elaborado una base de datos de obras de arte, pintores, temáticas, épocas y salas. Para generar todas las instancias se ha utilizado la herramienta chat GPT para generar diferentes obras las cuales hemos revisado y adaptado, básicamente son obras inventadas siguiendo el formato de nuestras fuentes de información. Una vez obtenidos las diferentes obras, relacionamos sus atributos con las respectivas instancias de época, sala, temática. Una vez obtenidos todos los datos se han ido creando las instancias, indicando sus respectivos datos y relaciones.

<i><b>Obra</b></i>	<i><b>Pintor</b></i>	<i><b>Época</b></i>	<i><b>Temática</b></i>	<i><b>A.Creacion</b></i>	<i><b>Relevancia</b></i>	<i><b>Área</b></i>
Sombras Cibernéticas	Neo Dalí	Era de Las Máquinas	Futurismo	2190	9	135
Caballeros Legendarios	Sir Arthur Lancelot	Edad Medieval	Caballerosidad	1210	10	145
Futurismo Sombras	Morpheus Turner	Edad de Las Sombras	Arte Simbólico Mítico	1475	7	150

**Fig 4 Ejemplo de Obras**

## 4.5 Criterios de implementación

Lo más destacable de esta práctica es que se ha implementado utilizando CLIPS, un lenguaje especializado en sistemas expertos, cuya sintaxis tiene similitudes con LISP y emplea una notación estilo polaco. CLIPS pertenece al paradigma declarativo basado en reglas, en contraste con lenguajes más procedurales.

Una característica adicional importante de CLIPS es su orientación a objetos, que ha sido particularmente útil para trabajar con la ontología propuesta. Gracias a los métodos `getter` y `setter` generados automáticamente por el lenguaje, se ha logrado manejar las entidades de manera eficiente.

Para garantizar la calidad del código, se han aplicado varias estrategias, como dividirlo en módulos para incrementar la modularidad y facilitar su comprensión en caso de futuras modificaciones. También se ha evitado el uso de variables globales para prevenir estados inesperados y aprovechar la independencia de los módulos. Por consiguiente, el sistema cuenta con alta escalabilidad, reusabilidad y extensibilidad. Asimismo, se ha trabajado con un único *template*.

## 4.6 Prototipos

Para llevar a cabo esta práctica, se empleó una metodología incremental. Inicialmente, desarrollamos un prototipo funcional básico (aunque lejos de ser óptimo) y, posteriormente, se realizaron mejoras graduales hasta alcanzar el prototipo final.

En total, se construyeron dos prototipos completamente funcionales, donde se ha considerado funcional el primer prototipo el cual solo hacía hasta el módulo de filtrado. A continuación, se describen las características principales de cada uno de ellos.

### 4.6.1 Primer prototipo

El primer prototipo permitía al usuario ingresar datos y aplicaba algunas de las restricciones establecidas. Sin embargo, no se estaba completamente seguro de que el filtro funciona de manera adecuada. Una vez procesada la entrada, el programa enviaba las obras a visitar que eran las que habían sobrevivido al filtrado.

Este prototipo fue entregado en la fase intermedia del proyecto. Aunque cumplía con la funcionalidad básica de solo visitar obras con al menos algún interés, no lograba respetar muchas de las restricciones ni las preferencias tanto en tiempo ni en qué día se estaba visitando. Por ello, el segundo prototipo se centró en solucionar estas deficiencias para ofrecer un sistema más completo y preciso además de acabar el módulo de síntesis, dado que este módulo ofrecía errores y se consideró mejor enviar un prototipo sin este modulo pero que si funcionara.

## 4.6.2 Segundo Prototipo y final

En la segunda versión, se incorporaron tanto las restricciones como las preferencias del usuario además del módulo de síntesis que distribuye las obras entre los días de visitas, lo optimiza e imprime la mejor solución encontrada. El proceso comenzaba eliminando todas las obras que el usuario no tenía ningún tipo de interés. A continuación, se generaba un itinerario inicial de manera aleatoria utilizando las obras con alguno de los intereses permitidos. Posteriormente, este itinerario se optimiza tomando en cuenta los datos del visitante (cuántos niños hay, ... etc). Este enfoque permitió un sistema más refinado y totalmente funcional.

# 5. Juegos de pruebas

## 5.1 Juego de pruebas normal 1

### Objetivo

El objetivo de este juego de pruebas es comprobar el funcionamiento normal del programa, es decir, que funciona sin tener en cuenta nuestros casos extremos. En definitiva, la idea de este juego de pruebas es que el programa funciona para la gran mayoría de datos, por lo que el resultado esperado es una distribución de las obras entre todos los días disponibles bastante equitativa y que en todos los días no se supere el tiempo previsto por día.

### Entrada

Se utilizara dos familias estándar dos padres y un niño, están en unas vacaciones de paso y solo disponen de 3 días para poder visitar, como quieren aprovechar la tarde o la mañana para visitar la ciudad, por lo tanto, solo disponen 4 horas al día para la visita, puede ser por la mañana o por la tarde. Son simplemente turistas y su conocimiento de arte en conjunto podría decirse que es una 40 entre 100, básicamente sus conocimientos son estándares. Donde se seleccionan 3 preferencias por categoría.

Datos del visitante:

```
-----  
  
Numero de personas: 6 personas  
Numero de ninos: 2 ninos  
Dias de visita: 3 dias  
Duracion de la visita por dia: 240 minutos  
Preferencia tematica: ([Simbolismo_Magico] [Arte_Cuantico] [Futurismo]  
[Ingenieria_Espacial] [Magia_Tenebrosa] fin)  
Preferencia artista: ([Quantum_Michelangelo] [Solaris_Van_Gogh]  
[Chronos_El_Grande] fin)  
Preferencia epoca: ([Edad_de_Las_Sombras] [Era_de_Las_Maquinas]  
[Era_de_Los_Cristales] fin)  
Nivel de conocimiento artisitico: 40  
-----
```

## Salida

-----  
Intentando eliminar obra sin preferencias: ("Paisajes Retro 2")  
Eliminando obra: ("Paisajes Retro 2")  
Intentando eliminar obra sin preferencias: ("Universo Biopunk")  
Eliminando obra: ("Universo Biopunk")  
Intentando eliminar obra sin preferencias: ("Caballeros Legendarios")  
Eliminando obra: ("Caballeros Legendarios")  
Intentando eliminar obra sin preferencias: ("La Explosión Biótica")  
Eliminando obra: ("La Explosión Biótica")  
Intentando eliminar obra sin preferencias: ("Luz Verde")  
Eliminando obra: ("Luz Verde")  
Intentando eliminar obra sin preferencias: ("Retro Paisajes")  
Eliminando obra: ("Retro Paisajes")  
Intentando eliminar obra sin preferencias: ("Revolución Viva")  
Eliminando obra: ("Revolución Viva")  
Intentando eliminar obra sin preferencias: ("Caballeros Luminosos")  
Eliminando obra: ("Caballeros Luminosos")  
Intentando eliminar obra sin preferencias: ("Luz de Las Sombras")  
Eliminando obra: ("Luz de Las Sombras")

-----  
Obras restantes después del filtrado:

- ("Sueño Cibernético")
- ("Explosión Cristalina")
- ("La Máquina del Tiempo")
- ("Sombras Mágicas")
- ("Futuro Simbólico")
- ("El Universo Crístico")
- ("Bosque de Sombras")
- ("La Estrella Rota")
- ("Tiempo y Espacio")
- ("Sombras Futuristas")
- ("La Máquina de Sombras")
- ("Caballeros Cibernéticos")
- ("La Luz Infinita")
- ("El Cristal Sombrío")
- ("Retro Vision")
- ("El Sol Entre Máquinas")
- ("El Código Mágico")
- ("Luz Infinita 2")
- ("Sombras Cibernéticas")
- ("Futurismo Sombras")
- ("Cristales Luminosos")
- ("Estrella Crística")

- ("Tiempo y Magia")
- ("Oscuridad Lumínica")

-----

Mejorando la Distribución actual: -162 de score

Mejorando la Distribución actual: -136 de score

Mejorando la Distribución actual: -31 de score

La distribución no se puede mejorar, distribución óptima

Obras:

Día 1:

Obra: ("Futuro Simbólico")

Sala: [Sala\_Futurismo]

Número de Sala: (8)

Tiempo estimado para la obra: 8.1165568 minutos

Obra: ("Oscuridad Lumínica")

Sala: [Sala\_Magia\_y\_Luz]

Número de Sala: (3)

Tiempo estimado para la obra: 4.6261248 minutos

Obra: ("La Estrella Rota")

Sala: [Sala\_Epoca\_Medieval]

Número de Sala: (5)

Tiempo estimado para la obra: 5.3294208 minutos

Obra: ("Sombras Cibernéticas")

Sala: [Sala\_Neo\_Dali]

Número de Sala: (1)

Tiempo estimado para la obra: 5.7670272 minutos

Obra: ("Sombras Futuristas")

Sala: [Sala\_Neo\_Dali]

Número de Sala: (1)

Tiempo estimado para la obra: 5.3971456 minutos

Obra: ("El Universo Crístico")

Sala: [Sala\_Epoca\_Modernista]

Número de Sala: (7)

Tiempo estimado para la obra: 6.8349952 minutos

Obra: ("La Máquina del Tiempo")

Sala: [Sala\_Quantum\_Michelangelo]

Número de Sala: (2)

Tiempo estimado para la obra: 9.37728 minutos

Obra: ("Explosión Cristalina")



Sala: [Sala\_Naturaleza\_y\_Paisaje]  
Número de Sala: (4)  
Tiempo estimado para la obra: 5.8451712 minutos

Día 2:  
Obra: ("Cristales Luminosos")  
Sala: [Sala\_Naturaleza\_y\_Paisaje]  
Número de Sala: (4)  
Tiempo estimado para la obra: 5.6732544 minutos

Obra: ("Futurismo Sombras")  
Sala: [Sala\_Futurismo]  
Número de Sala: (8)  
Tiempo estimado para la obra: 5.157504 minutos

Obra: ("Sombras Mágicas")  
Sala: [Sala\_Magia\_y\_Luz]  
Número de Sala: (3)  
Tiempo estimado para la obra: 4.532352 minutos

Obra: ("Luz Infinita 2")  
Sala: [Sala\_Magia\_y\_Luz]  
Número de Sala: (3)  
Tiempo estimado para la obra: 4.81888 minutos

Obra: ("Sueño Cibernético")  
Sala: [Sala\_Neo\_Dali]  
Número de Sala: (1)  
Tiempo estimado para la obra: 5.5534336 minutos

Obra: ("Caballeros Cibernéticos")  
Sala: [Sala\_Neo\_Dali]  
Número de Sala: (1)  
Tiempo estimado para la obra: 9.63776 minutos

Obra: ("La Luz Infinita")  
Sala: [Sala\_Magia\_y\_Luz]  
Número de Sala: (3)  
Tiempo estimado para la obra: 9.37728 minutos

Obra: ("Tiempo y Magia")  
Sala: [Sala\_Quantum\_Michelangelo]  
Número de Sala: (2)  
Tiempo estimado para la obra: 8.439552 minutos

Día 3:

Obra: ("Tiempo y Espacio")

Sala: [Sala\_Quantum\_Michelangelo]

Número de Sala: (2)

Tiempo estimado para la obra: 8.439552 minutos

Obra: ("Estrella Crística")

Sala: [Sala\_Epoca\_Modernista]

Número de Sala: (7)

Tiempo estimado para la obra: 7.2621824 minutos

Obra: ("Bosque de Sombras")

Sala: [Sala\_Naturaleza\_y\_Paisaje]

Número de Sala: (4)

Tiempo estimado para la obra: 5.2043904 minutos

Obra: ("El Cristal Sombrío")

Sala: [Sala\_Naturaleza\_y\_Paisaje]

Número de Sala: (4)

Tiempo estimado para la obra: 6.1890048 minutos

Obra: ("Retro Vision")

Sala: [Sala\_Epoca\_Modernista]

Número de Sala: (7)

Tiempo estimado para la obra: 2.5683328 minutos

Obra: ("El Sol Entre Máquinas")

Sala: [Sala\_Epoca\_Industrial]

Número de Sala: (6)

Tiempo estimado para la obra: 8.908416 minutos

Obra: ("El Código Mágico")

Sala: [Sala\_Magia\_y\_Luz]

Número de Sala: (3)

Tiempo estimado para la obra: 6.407808 minutos

Obra: ("La Máquina de Sombras")

Sala: [Sala\_Futurismo]

Número de Sala: (8)

Tiempo estimado para la obra: 5.1262464 minutos

## Conclusión

El juego de pruebas realizado demuestra que el programa cumple con los objetivos previstos. Se logra una distribución equilibrada de las obras a lo largo de los tres días de visita, respetando las restricciones de tiempo máximo por día (240 minutos). Además, la selección de obras se alinea con las preferencias temáticas, artísticas y de época indicadas por los visitantes, tras el proceso de filtrado de obras no preferidas.

En términos de optimización, el programa genera distribuciones con una puntuación negativa inicial que mejora progresivamente hasta alcanzar una distribución óptima. Cada día incluye una variedad de obras en diferentes salas, permitiendo a los visitantes disfrutar de una experiencia enriquecedora sin saturar ninguna categoría ni exceder el tiempo disponible.

Este resultado confirma que el programa puede procesar datos de entrada estándar y producir resultados que cumplen con las expectativas funcionales: respetar las limitaciones de tiempo, distribuir las obras de forma equitativa, y ajustarse a las preferencias de los usuarios. Por lo tanto, el sistema responde correctamente a las necesidades de la mayoría de usuarios en un entorno de uso normal.

## 5.2 Juego de pruebas normal 2

### Objetivo

El objetivo de este problema es mostrar el funcionamiento normal del programa, esta vez para un grupo de visitantes grande, con un alto conocimiento de arte, que visitan una sola tarde. Un posible escenario para este juego de pruebas sería un grupo de estudiantes de arte visitando un museo como parte de un proyecto de clase. El resultado esperado es que al ser un grupo mucho más grande y con menos tiempo, no van a poder visitar todas las obras de las temáticas interesantes, por lo tanto, el programa decidirá prescindir de algunas obras. Para más consistencia con el caso hipotético, las preferencias de los visitantes van a ser únicamente artistas, ya que tiene sentido que unos estudiantes sólo quieran ver las obras de un artista determinado, o unos cuantos.

### Entrada

Datos del visitante:

```
-----  
Numero de personas: 30 personas  
Numero de ninos: 0 ninos  
Dias de visita: 1 dias  
Duracion de la visita por dia: 180 minutos  
Preferencia tematica: (fin)  
Preferencia artista: ([Neo_Dali] [Dark_Raphael] [Stella_Kandinsky]  
[Sir_Arthur_Lancelot] [Solaris_Van_Gogh] fin)  
Preferencia epoca: (fin)
```

Nivel de conocimiento artistico: 95

## Salida

-----

Intentando eliminar obra sin preferencias: ("Tiempo y Magia")  
Eliminando obra: ("Tiempo y Magia")  
Intentando eliminar obra sin preferencias: ("Estrella Crística")  
Eliminando obra: ("Estrella Crística")  
Intentando eliminar obra sin preferencias: ("Paisajes Retro 2")  
Eliminando obra: ("Paisajes Retro 2")  
Intentando eliminar obra sin preferencias: ("Universo Biopunk")  
Eliminando obra: ("Universo Biopunk")  
Intentando eliminar obra sin preferencias: ("Futurismo Sombras")  
Eliminando obra: ("Futurismo Sombras")  
Intentando eliminar obra sin preferencias: ("Luz Infinita 2")  
Eliminando obra: ("Luz Infinita 2")  
Intentando eliminar obra sin preferencias: ("El Código Mágico")  
Eliminando obra: ("El Código Mágico")  
Intentando eliminar obra sin preferencias: ("Retro Vision")  
Eliminando obra: ("Retro Vision")  
Intentando eliminar obra sin preferencias: ("La Explosión Biótica")  
Eliminando obra: ("La Explosión Biótica")  
Intentando eliminar obra sin preferencias: ("La Luz Infinita")  
Eliminando obra: ("La Luz Infinita")  
Intentando eliminar obra sin preferencias: ("La Máquina de Sombras")  
Eliminando obra: ("La Máquina de Sombras")  
Intentando eliminar obra sin preferencias: ("Luz Verde")  
Eliminando obra: ("Luz Verde")  
Intentando eliminar obra sin preferencias: ("Tiempo y Espacio")  
Eliminando obra: ("Tiempo y Espacio")  
Intentando eliminar obra sin preferencias: ("La Estrella Rota")  
Eliminando obra: ("La Estrella Rota")  
Intentando eliminar obra sin preferencias: ("Bosque de Sombras")  
Eliminando obra: ("Bosque de Sombras")  
Intentando eliminar obra sin preferencias: ("El Universo Crístico")  
Eliminando obra: ("El Universo Crístico")  
Intentando eliminar obra sin preferencias: ("Retro Paisajes")  
Eliminando obra: ("Retro Paisajes")  
Intentando eliminar obra sin preferencias: ("Revolución Viva")  
Eliminando obra: ("Revolución Viva")  
Intentando eliminar obra sin preferencias: ("Sombras Mágicas")  
Eliminando obra: ("Sombras Mágicas")  
Intentando eliminar obra sin preferencias: ("La Máquina del Tiempo")  
Eliminando obra: ("La Máquina del Tiempo")

-----

Obras restantes después del filtrado:

- ("Sueño Cibernético")
- ("Luz de Las Sombras")
- ("Explosión Cristalina")
- ("Caballeros Luminosos")
- ("Futuro Simbólico")
- ("Sombras Futuristas")
- ("Caballeros Cibernéticos")
- ("El Cristal Sombrío")
- ("El Sol Entre Máquinas")
- ("Sombras Cibernéticas")
- ("Caballeros Legendarios")
- ("Cristales Luminosos")
- ("Oscuridad Lumínica")

-----

Mejorando la Distribución actual: -14798 de score

Mejorando la Distribución actual: -8372 de score

Mejorando la Distribución actual: -2851 de score

Mejorando la Distribución actual: -400 de score

La distribución no se puede mejorar, distribución óptima

Obras:

Día 1:

Obra: ("Oscuridad Lumínica")

Sala: [Sala\_Magia\_y\_Luz]

Número de Sala: (3)

Tiempo estimado para la obra: 14.64312 minutos

Obra: ("Luz de Las Sombras")

Sala: [Sala\_Magia\_y\_Luz]

Número de Sala: (3)

Tiempo estimado para la obra: 13.42286 minutos

Obra: ("Cristales Luminosos")

Sala: [Sala\_Naturaleza\_y\_Paisaje]

Número de Sala: (4)

Tiempo estimado para la obra: 17.95761 minutos

Obra: ("Sombras Futuristas")

Sala: [Sala\_Neo\_Dali]

Número de Sala: (1)

Tiempo estimado para la obra: 17.08364 minutos

Obra: ("Caballeros Luminosos")

Sala: [Sala\_Epoca\_Medieval]

Número de Sala: (5)  
Tiempo estimado para la obra: 20.7774 minutos

Obra: ("El Cristal Sombrío")  
Sala: [Sala\_Naturaleza\_y\_Paisaje]  
Número de Sala: (4)  
Tiempo estimado para la obra: 19.59012 minutos

Obra: ("Caballeros Cibernéticos")  
Sala: [Sala\_Neo\_Dali]  
Número de Sala: (1)  
Tiempo estimado para la obra: 30.5065 minutos

Obra: ("Sombras Cibernéticas")  
Sala: [Sala\_Neo\_Dali]  
Número de Sala: (1)  
Tiempo estimado para la obra: 18.25443 minutos

Obra: ("Futuro Simbólico")  
Sala: [Sala\_Futurismo]  
Número de Sala: (8)  
Tiempo estimado para la obra: 25.69142 minutos

## Conclusión

El juego de pruebas produce los resultados esperados. Como se puede ver en el cálculo de la score, al principio se excede el tiempo, lo que se ve reflejado en una score negativa muy baja. No obstante, a base de eliminar obras se acaba reduciendo el tiempo hasta que deja de ser mayor al tiempo máximo. Eso se ve reflejado en la score, que es -400 finalmente, ya que algunas obras no se pueden ver, y eso penaliza, aunque no tanto como excederse del tiempo.

Además se puede ver como el tiempo por obra es mayor en comparación con los otros ejemplos, que es lo que cabe esperar cuando el visitante es un grupo grande y con más conocimiento artístico.

## 5.3 Juego de pruebas mucho tiempo

### Objetivo

El objetivo principal de este juego de pruebas consiste en ver cómo reacciona el sistema basado en conocimiento si le introducimos una entrada donde el visitante dispone de mucho tiempo para visitar el museo.

### Entrada

Utilizaremos una familia estándar de 4 personas y 2 niños como visitantes. Visitarán tanto cómo pueden el museo, por lo tanto 7 días y 720 minutos cada día (12h). Seleccionaremos

unas 2-4 preferencias de cada categoría para que tengan una decente cantidad de obras para ver. Los visitantes tienen un nivel de conocimiento artístico medio de unos 50 puntos de 100. Esta entrada nos da demasiado tiempo para visitar el museo, por lo que posiblemente no dispondremos de suficientes obras para cubrir todo el tiempo.

-----  
Datos del visitante:

Numero de personas: 4 personas

Numero de ninos: 2 ninos

Dias de visita: 7 dias

Duracion de la visita por dia: 720 minutos

Preferencia tematica: ([Ingenieria\_Espacial] [Futurismo] fin)

Preferencia artista: ([Neo\_Dali] [Morpheus\_Turner] fin)

Preferencia epoca: ([Oscurantismo\_Luminoso] [Era\_de\_Los\_Cristales]  
[Edad\_de\_La\_Luz] [Neomedievalismo] fin)

Nivel de conocimiento artisitico: 50  
-----

## Salida

Obras restantes después del filtrado:

- ("Sueño Cibernético")
  - ("Luz de Las Sombras")
  - ("Explosión Cristalina")
  - ("La Máquina del Tiempo")
  - ("El Universo Crístico")
  - ("Bosque de Sombras")
  - ("La Estrella Rota")
  - ("Luz Verde")
  - ("Sombras Futuristas")
  - ("La Máquina de Sombras")
  - ("Caballeros Cibernéticos")
  - ("La Luz Infinita")
  - ("El Cristal Sombrío")
  - ("El Código Mágico")
  - ("Luz Infinita 2")
  - ("Sombras Cibernéticas")
  - ("Futurismo Sombras")
  - ("Cristales Luminosos")
  - ("Estrella Crística")
  - ("Oscuridad Lumínica")
- 

Día 1:

Obra: ("Sombras Futuristas")

Sala: [Sala\_Neo\_Dali]  
Número de Sala: (1)  
Tiempo estimado para la obra: 5.264952 minutos

Obra: ("El Universo Crístico")  
Sala: [Sala\_Epoca\_Modernista]  
Número de Sala: (7)  
Tiempo estimado para la obra: 6.667584 minutos

Obra: ("Luz Infinita 2")  
Sala: [Sala\_Magia\_y\_Luz]  
Número de Sala: (3)  
Tiempo estimado para la obra: 4.70085 minutos

Día 2:  
Obra: ("La Máquina del Tiempo")  
Sala: [Sala\_Quantum\_Michelangelo]  
Número de Sala: (2)  
Tiempo estimado para la obra: 9.1476 minutos

Obra: ("Oscuridad Lumínica")  
Sala: [Sala\_Magia\_y\_Luz]  
Número de Sala: (3)  
Tiempo estimado para la obra: 4.512816 minutos

Obra: ("Sueño Cibernético")  
Sala: [Sala\_Neo\_Dali]  
Número de Sala: (1)  
Tiempo estimado para la obra: 5.417412 minutos

Día 3:  
Obra: ("Sombras Cibernéticas")  
Sala: [Sala\_Neo\_Dali]  
Número de Sala: (1)  
Tiempo estimado para la obra: 5.625774 minutos

Obra: ("La Estrella Rota")  
Sala: [Sala\_Epoca\_Medieval]  
Número de Sala: (5)  
Tiempo estimado para la obra: 5.198886 minutos

Obra: ("Bosque de Sombras")  
Sala: [Sala\_Naturaleza\_y\_Paisaje]  
Número de Sala: (4)



Tiempo estimado para la obra: 5.076918 minutos

Día 4:

Obra: ("Caballeros Cibernéticos")

Sala: [Sala\_Neo\_Dali]

Número de Sala: (1)

Tiempo estimado para la obra: 9.4017 minutos

Obra: ("El Código Mágico")

Sala: [Sala\_Magia\_y\_Luz]

Número de Sala: (3)

Tiempo estimado para la obra: 6.25086 minutos

Día 5:

Obra: ("La Luz Infinita")

Sala: [Sala\_Magia\_y\_Luz]

Número de Sala: (3)

Tiempo estimado para la obra: 9.1476 minutos

Obra: ("Luz de Las Sombras")

Sala: [Sala\_Magia\_y\_Luz]

Número de Sala: (3)

Tiempo estimado para la obra: 4.136748 minutos

Obra: ("La Máquina de Sombras")

Sala: [Sala\_Futurismo]

Número de Sala: (8)

Tiempo estimado para la obra: 5.000688 minutos

Día 6:

Obra: ("Estrella Crística")

Sala: [Sala\_Epoca\_Modernista]

Número de Sala: (7)

Tiempo estimado para la obra: 7.084308 minutos

Obra: ("Luz Verde")

Sala: [Sala\_Magia\_y\_Luz]

Número de Sala: (3)

Tiempo estimado para la obra: 3.537072 minutos

Obra: ("El Cristal Sombrío")

Sala: [Sala\_Naturaleza\_y\_Paisaje]

Número de Sala: (4)

Tiempo estimado para la obra: 6.037416 minutos

Día 7:

Obra: ("Cristales Luminosos")

Sala: [Sala\_Naturaleza\_y\_Paisaje]

Número de Sala: (4)

Tiempo estimado para la obra: 5.534298 minutos

Obra: ("Explosión Cristalina")

Sala: [Sala\_Naturaleza\_y\_Paisaje]

Número de Sala: (4)

Tiempo estimado para la obra: 5.702004 minutos

Obra: ("Futurismo Sombras")

Sala: [Sala\_Futurismo]

Número de Sala: (8)

Tiempo estimado para la obra: 5.03118 minutos

La salida resultante nos indica que, tal y como habíamos previsto con la entrada, no hay suficientes obras para llenar todo el tiempo que estaremos en el museo. Como consecuencia, se seleccionan todas las obras que tienen alguna de nuestras preferencias y se reparten entre todos los días que queremos visitar. Como resultado, en cada sesión de cada día no tenemos suficientes obras para cubrir todo el tiempo, en conclusión es el resultado esperado.

## 5.4 Juego de pruebas poco tiempo

### Objetivo

El objetivo principal de este juego de pruebas consiste en ver cómo reacciona el sistema basado en conocimiento si le introducimos una entrada donde el visitante dispone de muy poco tiempo para visitar el museo.

### Entrada

Utilizaremos una familia estándar de 4 personas y 2 niños como visitantes. Visitarán tan poco como posible el museo, por lo tanto 1 día y 1 minuto. Seleccionaremos unas 2-4 preferencias de cada categoría para que tengan una decente cantidad de obras para ver. Los visitantes tienen un nivel de conocimiento artístico medio de unos 50 puntos de 100. Esta entrada no nos da tiempo suficiente para ver ninguna obra, por lo que posiblemente no nos dará ninguna obra a ver.

-----  
Datos del visitante:

Numero de personas: 4 personas  
Numero de ninos: 2 ninos  
Dias de visita: 1 dias  
Duracion de la visita por dia: 1 minutos  
Preferencia tematica: ([Ingenieria\_Espacial] [Futurismo] fin)  
Preferencia artista: ([Neo\_Dali] [Chronos\_El\_Grande] fin)  
Preferencia epoca: ([Edad\_de\_La\_Luz] [Paleofuturismo] fin)  
Nivel de conocimiento artisitico: 50  
-----

### Salida

-----  
Obras restantes después del filtrado:

- ("Sueño Cibernético")
- ("La Máquina del Tiempo")
- ("Retro Paisajes")
- ("El Universo Crístico")
- ("La Estrella Rota")
- ("Luz Verde")
- ("Sombras Futuristas")
- ("La Máquina de Sombras")
- ("Caballeros Cibernéticos")
- ("La Luz Infinita")
- ("Retro Vision")
- ("El Código Mágico")
- ("Luz Infinita 2")
- ("Sombras Cibernéticas")

- ("Paisajes Retro 2")
- ("Estrella Crística")

-----

Día 1:

Podemos observar que, tal y como habíamos previsto, no nos asigna ninguna obra en el itinerario, aunque sí que nos quedan obras disponibles según nuestras preferencias después del filtrado.

## 5.5 Juego de pruebas todas las preferencias

### Objetivo

El objetivo principal de este juego de pruebas consiste en ver cómo reacciona el sistema basado en conocimiento si le introducimos una entrada donde el visitante tiene todas las temáticas, épocas y pintores como preferencias.

### Entrada

Utilizaremos una familia estándar de 4 personas y 2 niños como visitantes. La visita durará unos 2 días y cada sesión de 45 minutos. Seleccionaremos todas las preferencias. Los visitantes tienen un nivel de conocimiento artístico medio de unos 50 puntos de 100. Esta entrada nos coge por lo tanto todas las obras del museo como preferidas, posiblemente no siendo posible visitar todas las obras.

-----

Datos del visitante:

Numero de personas: 4 personas

Numero de ninos: 2 ninos

Dias de visita: 2 dias

Duracion de la visita por dia: 45 minutos

Preferencia tematica: ([Arte\_Bio\_Organico] [Arte\_Cuantico] [Arte\_Luminico]  
[Arte\_Simbolico\_Mitico] [Caballerosidad] [Constructivismo\_Cristico]  
[Futurismo] [Impresionismo\_Cibernetico] [Ingenieria\_Espacial]  
[Magia\_Tenebrosa] [Paisajismo\_Intergalactico] [Simbolismo\_Magico] fin)

Preferencia artista: ([Neo\_Dali] [Dark\_Raphael] [Stella\_Kandinsky]  
[Quantum\_Michelangelo] [Hecate\_Sombria] [Sir\_Arthur\_Lancelot]  
[Solaris\_Van\_Gogh] [Flora\_Bosch] [Juno\_Warhol] [Chronos\_El\_Grande]  
[Morpheus\_Turner] [Lucifer\_Corot] [Artemis\_Renoir] fin)

Preferencia epoca: ([Edad\_Medieval] [Edad\_de\_La\_Luz] [Edad\_de\_Las\_Sombras]  
[Era\_de\_Las\_Maquinas] [Era\_de\_Los\_Cristales] [Futuro\_Optimista]  
[Neomedievalismo] [Oscurantismo\_Luminoso] [Oscurantismo\_Magico]  
[Paleofuturismo] [Revolucion\_Biopunk] fin)

Nivel de conocimiento artisitico: 50

-----

## Salida

-----  
Obras restantes después del filtrado:

- ("Sueño Cibernético")
- ("Luz de Las Sombras")
- ("Explosión Cristalina")
- ("La Máquina del Tiempo")
- ("Sombras Mágicas")
- ("Caballeros Luminosos")
- ("Futuro Simbólico")
- ("Revolución Viva")
- ("Retro Paisajes")
- ("El Universo Crístico")
- ("Bosque de Sombras")
- ("La Estrella Rota")
- ("Tiempo y Espacio")
- ("Luz Verde")
- ("Sombras Futuristas")
- ("La Máquina de Sombras")
- ("Caballeros Cibernéticos")
- ("La Luz Infinita")
- ("El Cristal Sombrío")
- ("La Explosión Biótica")
- ("Retro Vision")
- ("El Sol Entre Máquinas")
- ("El Código Mágico")
- ("Luz Infinita 2")
- ("Sombras Cibernéticas")
- ("Caballeros Legendarios")
- ("Futurismo Sombras")
- ("Universo Biopunk")
- ("Paisajes Retro 2")
- ("Cristales Luminosos")
- ("Estrella Crística")
- ("Tiempo y Magia")
- ("Oscuridad Lumínica")

-----  
Día 1:

Obra: ("La Luz Infinita")

Sala: [Sala\_Magia\_y\_Luz]

Número de Sala: (3)

Tiempo estimado para la obra: 9.1476 minutos

Obra: ("Futuro Simbólico")

Sala: [Sala\_Futurismo]

Número de Sala: (8)  
Tiempo estimado para la obra: 7.917756 minutos

Obra: ("Estrella Crística")  
Sala: [Sala\_Epoca\_Modernista]  
Número de Sala: (7)  
Tiempo estimado para la obra: 7.084308 minutos

Obra: ("La Máquina del Tiempo")  
Sala: [Sala\_Quantum\_Michelangelo]  
Número de Sala: (2)  
Tiempo estimado para la obra: 9.1476 minutos

Obra: ("El Sol Entre Máquinas")  
Sala: [Sala\_Epoca\_Industrial]  
Número de Sala: (6)  
Tiempo estimado para la obra: 8.69022 minutos

Día 2:  
Obra: ("Paisajes Retro 2")  
Sala: [Sala\_Naturaleza\_y\_Paisaje]  
Número de Sala: (4)  
Tiempo estimado para la obra: 2.800182 minutos

Obra: ("Sombras Cibernéticas")  
Sala: [Sala\_Neo\_Dali]  
Número de Sala: (1)  
Tiempo estimado para la obra: 5.625774 minutos

Obra: ("Sueño Cibernético")  
Sala: [Sala\_Neo\_Dali]  
Número de Sala: (1)  
Tiempo estimado para la obra: 5.417412 minutos

Obra: ("Revolución Viva")  
Sala: [Sala\_Epoca\_Industrial]  
Número de Sala: (6)  
Tiempo estimado para la obra: 3.572646 minutos

Obra: ("Bosque de Sombras")  
Sala: [Sala\_Naturaleza\_y\_Paisaje]  
Número de Sala: (4)  
Tiempo estimado para la obra: 5.076918 minutos

Obra: ("Cristales Luminosos")

Sala: [Sala\_Naturaleza\_y\_Paisaje]  
Número de Sala: (4)  
Tiempo estimado para la obra: 5.534298 minutos

Obra: ("Explosión Cristalina")  
Sala: [Sala\_Naturaleza\_y\_Paisaje]  
Número de Sala: (4)  
Tiempo estimado para la obra: 5.702004 minutos

Obra: ("La Explosión Biótica")  
Sala: [Sala\_Epoca\_Industrial]  
Número de Sala: (6)  
Tiempo estimado para la obra: 3.958878 minutos

Obra: ("El Universo Crístico")  
Sala: [Sala\_Epoca\_Modernista]  
Número de Sala: (7)  
Tiempo estimado para la obra: 6.667584 minutos

En la salida obtenida podemos fijarnos que en efecto tenemos todas las obras del museo y como consecuencia si que nos asigna tantas obras como pueda con nuestro tiempo disponible a cada día. Pero no vemos todas las obras por falta de tiempo.

## 5.6 Juego de pruebas ninguna preferencia

### Objetivo

El objetivo principal de este juego de pruebas consiste en ver cómo reacciona el sistema basado en conocimiento si le introducimos una entrada donde el visitante no tiene ninguna preferencia.

### Entrada

Utilizaremos una familia estándar de 4 personas y 2 niños como visitantes. La visita durará unos 2 días y cada sesión de 40 minutos. No seleccionaremos ninguna preferencia. Los visitantes tienen un nivel de conocimiento artístico medio de unos 50 puntos de 100. Esta entrada no nos da ninguna preferencia como consecuencia posiblemente no haya obras por visitar.

-----  
Datos del visitante:

Numero de personas: 4 personas  
Numero de ninos: 2 ninos  
Dias de visita: 2 dias  
Duracion de la visita por dia: 40 minutos  
Preferencia tematica: (fin)  
Preferencia artista: (fin)  
Preferencia epoca: (fin)  
Nivel de conocimiento artisitico: 50  
-----

### Salida

-----  
Obras restantes después del filtrado:  
No quedan obras después del filtrado.  
-----

Día 1:

Día 2:

Podemos observar que, tal y como habíamos previsto, no nos queda ninguna obra después de filtrar todas las obras que no tienen ninguna de nuestras preferencias. Por lo tanto, al final nos queda una respuesta vacía por cada día.



## 5.7 Juego de pruebas mucha gente y niños

### Objetivo

El objetivo principal de este juego de pruebas consiste en ver cómo reacciona el sistema basado en conocimiento si le introducimos una entrada donde hay muchos visitantes y una gran variedad de niños.

### Entrada

Utilizaremos un grupo del máximo permitido de personas (50) y con 30 niños como visitantes. La visita durará unos 2 días y cada sesión de 90 minutos (un poco más que otros juegos de pruebas para que se pueda ver al menos alguna obra asignada algún día). Seleccionaremos unas 2-4 preferencias de cada categoría para que tengan una decente cantidad de obras para ver. Los visitantes tienen un nivel de conocimiento artístico medio de unos 50 puntos de 100. En esta entrada por lo tanto vamos a tener un gran grupo de gente que va a hacer qué visitar cada obra dure bastante tiempo.

-----  
Datos del visitante:

Numero de personas: 50 personas

Numero de ninos: 30 ninos

Dias de visita: 2 dias

Duracion de la visita por dia: 90 minutos

Preferencia tematica: ([Arte\_Luminico] [Futurismo]  
[Impresionismo\_Cibernetico] fin)

Preferencia artista: ([Quantum\_Michelangelo] [Chronos\_El\_Grande]  
[Flora\_Bosch] fin)

Preferencia epoca: ([Edad\_Medieval] [Oscurantismo\_Luminoso] fin)

Nivel de conocimiento artisitico: 50  
-----

### Salida

-----  
Obras restantes después del filtrado:

- ("Sueño Cibernético")
- ("Luz de Las Sombras")
- ("La Máquina del Tiempo")
- ("Caballeros Luminosos")
- ("Revolución Viva")
- ("El Universo Crístico")
- ("La Estrella Rota")
- ("Tiempo y Espacio")
- ("La Máquina de Sombras")
- ("La Explosión Biótica")
- ("Sombras Cibernéticas")
- ("Caballeros Legendarios")

- ("Universo Biopunk")
- ("Estrella Crística")
- ("Tiempo y Magia")
- ("Oscuridad Lumínica")

-----

Día 1:

Obra: ("Tiempo y Magia")

Sala: [Sala\_Quantum\_Michelangelo]

Número de Sala: (2)

Tiempo estimado para la obra: 70.875 minutos

Día 2:

Obra: ("Sombras Cibernéticas")

Sala: [Sala\_Neo\_Dali]

Número de Sala: (1)

Tiempo estimado para la obra: 48.43125 minutos

Obra: ("Revolución Viva")

Sala: [Sala\_Epoca\_Industrial]

Número de Sala: (6)

Tiempo estimado para la obra: 30.75625 minutos

Podemos observar en la salida que, como consecuencia del gran grupo de gente, la duración para visitar cada obra ha aumentado significativamente. Por lo tanto, ahora necesitamos unos 40 minutos como mínimo para casi todas las obras. Como resultado, solo el día 2 tiene más de una obra para visitar. Aunque sí tenemos más obras con preferencia que se podrían visitar si tuviéramos más tiempo.

## 6.Conclusiones

Después de redactar el informe sobre esta práctica, podemos destacar varias conclusiones importantes:

- ***Sobre la construcción del SBC:*** Hemos logrado implementar las cuatro fases necesarias para el desarrollo de la práctica. El código que hemos desarrollado cumple con los pasos de caracterización, donde identificamos al usuario para diseñar una visita adaptada a sus necesidades; procesamiento, en el que eliminamos las obras que no tiene interés para el usuario; construcción, donde generamos el itinerario óptimo; y finalmente, presentación, en la que mostramos al usuario su dieta personalizada.
- ***Uso del lenguaje CLIPS:*** La elección de CLIPS como lenguaje de programación presentó muchos retos, principalmente debido a la dificultad para encontrar ejemplos más allá de los materiales proporcionados en la asignatura. Sin embargo, a pesar de estas limitaciones, hemos logrado avanzar en la construcción del SBC, aunque la sintaxis y las particularidades del lenguaje ralentizaron en ciertos momentos el desarrollo.
- ***Limitaciones del SBC:*** Reconocemos que el sistema construido es bastante básico y que una ampliación más robusta requeriría más recursos, el más importante siendo fuentes primarias de conocimiento experto en vez de artículos encontrados escritos por expertos que no necesariamente cubren en su totalidad las necesidades de este trabajo. Aun así, consideramos que este SBC representa una aproximación inicial que sirve como un primer paso hacia la creación de un generador más consistente y funcional de itinerarios.
- ***Posibles mejoras:*** Una de las principales áreas de mejora sería considerar una mayor variedad de obras. Esto permitiría generar una mayor diversidad de itinerarios para un mismo visitante, garantizando alternativas más variadas y adaptadas a las preferencias individuales. También el detalle que consideramos que si un usuario no tiene ninguna preferencia implica que el usuario no tienen ningún interés en ver dicha obra, si esta implicación no es cierta por lo tanto al no tener ninguna preferencia lo que se tendría que hacer es básicamente tener el mismo resultado que con todas las preferencias.
- ***Testing y validación:*** Hemos diseñado las pruebas necesarias para evaluar diversos aspectos del sistema Sin embargo, reconocemos que sería necesario realizar más pruebas para verificar exhaustivamente cada componente del SBC. Dado el alcance del problema, harían falta pruebas adicionales para evaluar todas las situaciones posibles y garantizar una solución integral. No obstante, muchas de esas situaciones son casos muy improbables que, aunque puedan producir soluciones poco precisas, no ocurrirían en la vida real. Finalmente, debido a la naturaleza del problema, no se puede asegurar que el código es realmente correcto sin probarlo en un caso real, pero en base al conocimiento que tenemos, los resultados parecen acertados.

## 7. Progreso trabajo Innovación

Actualmente, el trabajo de innovación está en progreso. Hasta este momento, hemos recopilado información de diversas fuentes. Además, hemos dedicado tiempo a analizar y filtrar la información para asegurar su veracidad y aplicabilidad. La información recopilada ha sido clasificada y organizada en categorías específicas que faciliten su integración en el documento final. En paralelo, hemos estructurado el contenido del trabajo, dividiéndolo en secciones claras y coherentes para asegurar que la redacción sea lógica. Este esquema incluye apartados destinados a la introducción, la información relevante y conclusiones, entre otros. Ahora, nos encontramos en la fase final, donde nos estamos enfocando en la redacción y revisión de los apartados. Este proceso incluye la verificación del contenido, la corrección de posibles errores, y la integración de los elementos recopilados. En resumen, el proyecto ha avanzado significativamente y se encuentra en su etapa de cierre.