

Universidad de La Habana

**FACULTAD DE MATEMÁTICA Y
COMPUTACIÓN**

**ESTADÍSTICAS PARA CONOCER
AL AÑO**

**Proyecto Final de Modelos de
Optimización II**
EQUIPO 6

Integrantes:

Olivia González Peña C-411

Sheyla Cruz Castro C-412

Laura Brito Guerrero C-412

Juan Carlos Casteleiro Wong C-411

0.1. Planteamiento del problema

Usuario final: Fernando Rodríguez Flores @fernan2rodriguez

0.1.1. Objetivos

El objetivo de este proyecto es diseñar e implementar un sistema (¿de encuestas? ¿de preguntas? ¿de tarjetas por el día de las madres?) que permita obtener información sobre un grupo de clases al comienzo del curso. Entre lo que se desea conocer están los intereses de los estudiantes (qué les gusta y a qué le dedican su tiempo libre). Cuánto conocen sobre determinados temas (videojuegos, películas, libros). Si juegan, leen o consumen series, y en esos casos qué tipos de juegos, libros, o series. Por ejemplo, en caso de que a alguien le guste la poesía, qué tipo de poesía, qué autores. ¿Qué tipo de música? Etc. ¿Cuál es su relación con las asignaturas que ya vieron? ¿Cuáles le gustaron? ¿Cuáles no les gustaron? ¿Cuáles les dio absolutamente lo mismo? ... etc.

El objetivo de esta primera parte es determinar qué tipo de elementos y referencias se pueden incluir en las conferencias y clases prácticas de forma que sean relevantes para la mayor cantidad de personas posibles.

También interesa conocer los niveles de creatividad y de sentido del humor que tengan los estudiantes. Esto es a través de los *tests* y *cuestionarios* ya establecidos para esos fines.

Algunas de estas informaciones conviene que sea *con nombre*, otras a lo mejor conviene que sea anónima.

Como parte del proyecto de debe:

1. Crear un mecanismo que permita recoger esta información *de la mejor manera posible*. *De la mejor manera posible* significa que quizás la mejor opción no sea ponerles una encuesta con varios cientos de incisos ... o a lo mejor sí.
2. Crear un sistema que permita analizar la información recogida.
3. Mostrar la información recopilada de una manera que resulte útil para la planificación de las asignaturas.

0.2. Modelo de solución

0.2.1. Implementación del Backend

Responsable: Laura Brito Guerrero

En el diseño de la lógica del problema se necesita la implementación de diversas estructuras de datos las cuales nos permite el manejo de la información más fácil para con el conocimiento de las mismas aplicar los algoritmos correspondientes que llevan a la solución del problema planteado. La idea central a seguir se basa en el análisis estadístico de las respuestas de los estudiantes a las encuestas, tales análisis se realizan tanto de manera individual como general. Para dar un

mayor entendimiento se pasa a explicar detalladamente la implementación de la misma.

poll.py

En este fichero se recogen los datos de la aplicación, es decir, a los estudiantes, los temas a considerar en las encuestas y las respuestas de los estudiantes a las mismas. Dicha información se recoge en forma de diccionario. Se tiene en consideración que las encuestas pueden ser tanto anónimas o no, esto no afecta a los algoritmos diseñados.

En el método **analize** se recoge esta información y conforme a esto se crean las estructuras de datos correspondientes, primeramente se diseña un contexto de temas (**Theme_Context**), el cual ordena eficientemente los temas siendo estos a la vez subtemas de otros temas, esto se diseña jerárquicamente.

A la vez, los estudiantes se recopilan en **Student_Preferences**, donde se guardan los datos de los estudiantes en caso preciso y las respuestas de los mismos a las encuestas.

Se analizan los estudiantes en pares no repetidos, se tiene la certeza que si se compara el estudiante i con el estudiante j se obtienen los mismos resultados que si se analiza al estudiante j con el i . Se comparan los criterios de preferencia de los estudiantes, qué tan alejados o cercanos están los mismos en sus gustos. Antes de explicar el proceso de comparación se pasa a explicar la organización de las preferencias de cada estudiante.

Una vez que se recojan las respuestas de cada estudiante (*Student_Preferences*) en **answers** (diccionario con llave *tema* y valor correspondiente: *respuesta del estudiante en ese tema*), se pasa a analizar sus preferencias (**preferences**) en orden descendente.

Se implementa una clase **AVLSearchBinaryTree_Insert** la cual crea un AVL con valores en los nodos (*tema, cantidad de respuestas al tema*), mientras mayor cantidad de aciertos tenga ese tema (que se considera una hoja de la jerarquía de los temas, ya que los temas *padres* se van desglosando en subtemas *hijos*, los cuales a su vez tienen su propia jerarquía, hasta caer en un subtema *hoja*, en la cual los estudiantes desarrollan o *marcan* sus gustos correspondientes), mayor es su preferencia. Estos nodos se organizan por el segundo valor de la tupla en forma ascendente. Luego de estar confeccionado el AVL, se realiza un recorrido entre-orden y devuelve una lista ordenada de las preferencias y las mismas se guardan en orden inverso en **preferences**.

Como por cada estudiante se tiene un orden en sus preferencias, se analizan los pares de estudiantes, esto se implementa en **invert_count**. Se toman **s1.preferences** y **s2.preferences** (**s1** y **s2** estudiantes) y se comparan, donde sin pérdida de generalidad se considera como arreglo ordenado a **s1.preferences**. Entonces, se ordena **s2.preferences** respecto a **s1.preferences**, y dependiendo de la cantidad de inversiones que tenga $tema_{i2}$ (el tema i de **s2**) con respecto a $tema_{i1}$ (tema i de **s1**) se establece la lejanía o cercanía del $tema_i$ que presentan ambos estudiantes. El criterio de cercanía (**s.likes**) o lejanía (**s.dislikes**), lo brinda un porcentaje, si la cantidad de inversiones es menor que un p por ciento

(no se define porque cada tema o encuesta tiene su propio patrón definido) entonces el $tema_i$ es común entre ambos estudiantes en su orden de preferencias. En caso contrario, los gustos de ambos estudiantes con respecto al tema están alejados. Después de conocer la relación entre ambos estudiantes se añade a las estructuras de datos correspondientes en la instancia de **s1** y **s2**. Luego de recorrer y analizar los gustos de todos los pares de estudiantes, se recurre al análisis general estadístico, el cual se guarda en la instancia de **Theme_Context**, y se implementa en el método **theme_context.stadistics_result**.

hierarchy.py

En dicho fichero se implementa la clase **Theme_Context** y **Theme_Scope**. En **Theme_Scope** se encuentra la jerarquía de los temas, y en **Theme_Context** se guardan estos temas y se realizan los análisis referentes a los mismos, como por ejemplo el análisis estadístico con respecto a las respuestas de los estudiantes en 0.2.1.

likes_dislikes.py

Se encuentra la clase **Student_Preferences**, donde se guardan los datos de cada estudiante, sus respuestas a las encuestas, la relación que guarda este con los $n - 1$ estudiantes restantes, sus preferencias con respecto a los temas.

invert_array.py

Se realiza el análisis **invert_count** mencionado en 0.2.1.

stadistics.py

Se implementa la clase **Stadistics_Op**, la cual se encarga de realizar el análisis estadístico por tema. Se implementa el cálculo de las medidas de Tendencia Central y las medidas de Dispersión, necesarias para una correcta interpretación. Dicho análisis por tema se instancia en 0.2.1, donde se guarda en el diccionario **stadistics** (*tema : análisis estadístico*).