



## **Trabajo Practico N°3** **Al SUM de la Biblioteca**

Materia: Programación III.

Profesor: Javier Marengo.

Patricia Bagnes.

Alumno: Ramos Aranibar Javier.



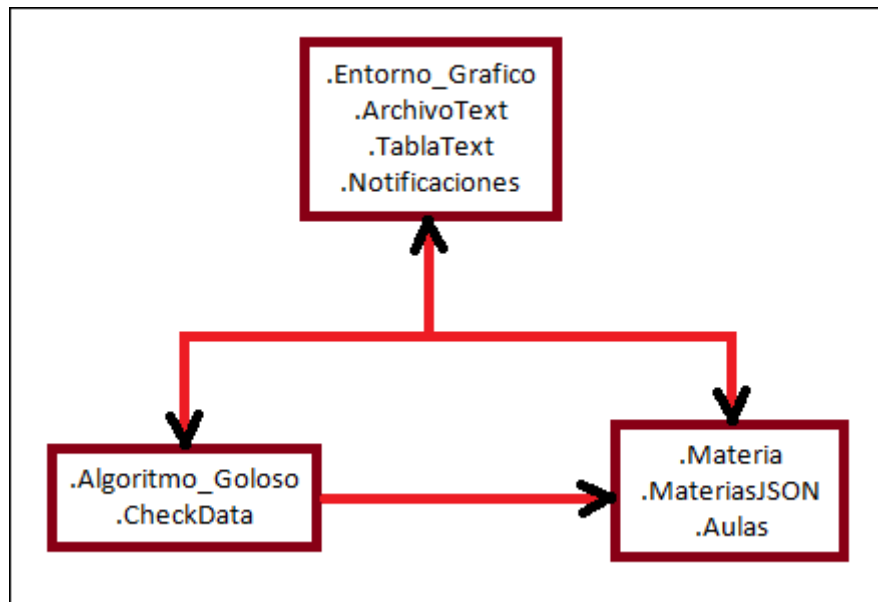
# **2017**

## Introducción

El presente trabajo practico tendrá como objetivo desarrollar la Aplicación “Al SUM de la biblioteca”, el cual asigna automáticamente aulas a materias de una universidad, con el objetivo de respetar las restricciones de los horarios. Para ello se implemento un algoritmo (goloso) que nos permitirá asignar las aulas, dándonos una solución, que puede no ser la más optima.

## Diagrama de Clases

El siguiente diagrama, representa todas las clases utilizadas y su relación entre ellas. A continuación se explicara con más detalles, las rutas tomadas.



Observemos que se puede diferenciar 3 grupos de clases, Capa de Presentación, Capa de Datos y Capa de Negocio.

.Capa de Presentación: Entorno\_Grafico, ArchivoText, TablaText y Notificaciones.

.Capa de Datos: Materia, MateriasJSON y Aulas.

.Capa de Negocios: Algoritmo\_Goloso, CheckData.

Veamos la función de las clases:

- Entorno \_Grafico: es la clase principal sobre el cual se inicializa el frame y parte todo accionar (a través de los JBoton) hacia las otras clases.
- ArchivoText, TablaTex y Notificaciones: Son parte de la inicialización de la clase principal. Además de contener métodos que setean sus variables de instancia, que posteriormente serán visualizados por el usuario.
- CheckData: Contiene métodos que analizan los datos ingresados por el usuario, y retorna una respuesta sobre la Clase Notificaciones.
- Algoritmo\_Goloso: Contiene un conjunto de métodos que procesara un conjunto de datos y posteriormente retornara un conjuntos de aulas con materias asignadas.  
Cabe aclarar que la estrategia que se utilizo para asignar las materias, implementando el algoritmo Goloso, fue ordenar las materias por el diferencial de tiempo entre Hora Inicial y Hora Final.

Ej

Materias:

Matemática I	Cod.: Ma-01	Horario: 8 a 12	-----> Su Dif. Es: 4 hs.
Física II	Cod.: Fi-01	Horario: 13 a 18	-----> Su Dif. Es: 5 hs.
Química I	Cod.: Qu-01	Horario: 20 a 22	-----> Su Dif. Es: 2 hs.

Aplicando mi algoritmo, obtendremos el siguiente orden de soluciones: Química I, Matemática I y Física II.

Luego, las asignaciones, serán en ese orden, pero respetando que no se interpongan horarios.

## Implementación

- Se optó por utilizar un único JFrame, dado que la complejidad del problema no demanda más de una ventana.
  - La aplicación me permite elegir los archivos sobre el cual voy a aplicar mi Asignador (fig. 1).
  - Además me permite visualizar el archivo elegido (fig. 1).
  - Se pueden agregar materias a gusto del usuario. Solo basta ingresar el nombre de la materia, el código del mismo (Comisión), y el horario (fig. 3).
- Como un punto a favor del usuario, se evalúan los datos, y se le da un informe breve de los errores de ingreso que podría a ver cometido.
- Además se puede eliminar cada materia de manera individual, con tan solo ingresando su código (fig. 3).
  - o eliminar todas las materias del archivo.
  - Eliminar todos el contenido del archivo, contiene una ventana de seguridad, que sirve de chequeo por si el usuario presiona accidentalmente ese botón (fig. 5).
  - El orden de las materias presentes en la tabla, se encuentran en orden según el horario (fig. 2).
  - Por último, se implemento un extra. Que posee los siguientes datos.
    - \_ Cantidad de aulas utilizadas.
    - \_ Usabilidad de cada aula en %  
Ej.: Cada Aula posee 14 hs de disponibilidad (22 - 8), y es usado 7 hs, entonces está siendo usado un 50%.
    - \_ Disponibilidad en horario. Muestra los horarios que se encuentran libre el aula.  
Ej.: Un aula tiene 2 materias asignadas:  
Matemática de 10 a 12, y Química de 14 a 18.  
Entonces la disponibilidad horaria, será: de 8 a 10, de 12 a 14 y de 18 a 22.

