UNIVERSIDAD DE ANTIOQUIA

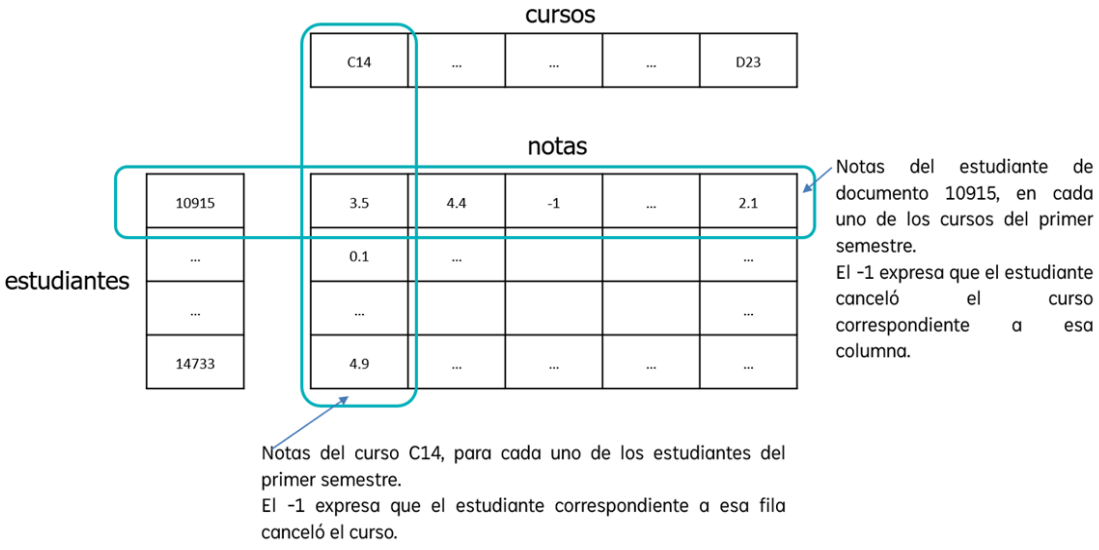
# Práctica 7. Ordenamiento, búsqueda y predicción.

## 1. Objetivos

- Aprender a desarrollar programas aplicando criterios de eficiencia en la cantidad de instrucciones ejecutadas y en el manejo de las estructuras de datos.
- Implementar operaciones administrativas básicas como ordenamiento y búsqueda sobre listas.
- Aprender a analizar, comparar, implementar y reutilizar algoritmos de ordenamiento existentes.
- Conocer de manera experimental el concepto de regresión lineal para hacer predicciones.

## PARTE 1

En la Universidad de Antioquia se desea generar algunas estadísticas acerca de los estudiantes del primer semestre de una carrera. Para ello, se cuenta con los números de documento de los estudiantes del primer semestre, el código de todos los cursos del primer semestre y la respectiva nota que cada estudiante ha obtenido en cada uno de los cursos. El punto de partida es la siguiente estructura de datos a partir de la cuál usted debe llevar a cabo su desarrollo:



La estructura está conformada por dos listas y una matriz (lista bidimensional). En el diagrama se puede observar que dichas estructuras están desacopladas, lo cual quiere decir que no están anidadas entre sí. Sin embargo, deben administrarse de forma conjunta dado que los elementos en las listas se encuentran dispuestos de tal manera que guardan una relación posicional con aquellos dentro de la matriz. Esto se puede constatar en las acotaciones presentadas con líneas azules alrededor de las filas y columnas de la matriz. En dicha matriz, se pueden identificar las notas del estudiante *i* en el curso *j*. El valor -1 dentro de la matriz indica que el estudiante i canceló el curso j, mientras que el valor -2 indica que el estudiante *i* no se ha matriculado en el curso *j*.

Esta práctica debe resolverse usando la estructura de datos propuesta y considerando sus propias implementaciones de funciones para administrar las listas involucradas. De ser necesario, puede definir variables o estructuras de datos auxiliares para satisfacer los requisitos del programa.

## Requerimientos del programa

*Requisitos funcionales*

El programa debe iniciar mostrando al usuario un menú principal del cual se seleccionan todas las funcionalidades:

1. **Cargar datos desde archivo**: Permite leer el archivo **database.csv** y cargar las estructuras de datos del programa.
2. **Eliminar estudiante**: Permite eliminar el estudiante correspondiente a un número de documento dado (y sus notas).
3. **Mayor nota de estudiante**: Retorna la nota y el código del curso en el cual ha obtenido su mayor nota un estudiante dado.
4. **Ordenar promedios estudiantes**: Imprime por pantalla a los estudiantes ordenados según su promedio de notas, de mayor a menor. Para esta funcionalidad usted debe usar obligatoriamente una estrategia de **ordenamiento burbuja**.
5. **Ordenar estudiantes por cantidad de cursos**: Ordena los estudiantes según la cantidad de materias cursadas e imprime el resultado por la pantalla. Para esta funcionalidad usted debe usar obligatoriamente una estrategia de **ordenamiento por selección**.

*Requisitos no funcionales*

- Eficiencia en el diseño de las estructuras de datos y en los procesos. Lea el apartado eficiencia más abajo en este documento.
- El código generado debe hacer uso de módulos y estar debidamente documentado.
- Para la construcción de esta solución, en ningún caso se debe hacer uso de diccionarios.

*Archivo de datos*

Los datos a ser procesados por el programa están almacenados en un archivo dado de nombre **notas_estudiantes.csv** cuyo formato se muestra a continuación:

```
código_curso_1,código_curso_2,código_curso_3, …,código_curso_n
doc_est_1,doc_est_2,doc_est_3, …,doc_est_n
nota_e1_c1,nota_e1_c2,nota_e1_c3, …,nota_e1_cn
nota_e2_c1,nota_e2_c2,nota_e2_c3, …,nota_e2_cn
…
nota_en_c1,nota_en_c2,nota_en_c3, …,nota_en_cn
```

En la primera línea, se tienen los códigos de cada uno de los cursos, separados por coma. En la segunda, los documentos de los estudiantes. Y luego se tiene una línea por cada estudiante, en la se encuentran separadas por coma las notas en cada curso.

*Eficiencia*

En esta práctica, uno de los elementos fundamentales de la evaluación será la eficiencia. Documente por escrito, a manera de informe en word, dos funciones en las que haya logrado reducir la cantidad de iteraciones requeridas a través de alguna estrategia analítica (por ejemplo, incrementando la complejidad conceptual). Incluya la versión original y la versión optimizada del código de la función, acompañándola de la explicación del análisis que aplicó para lograr la optimización.

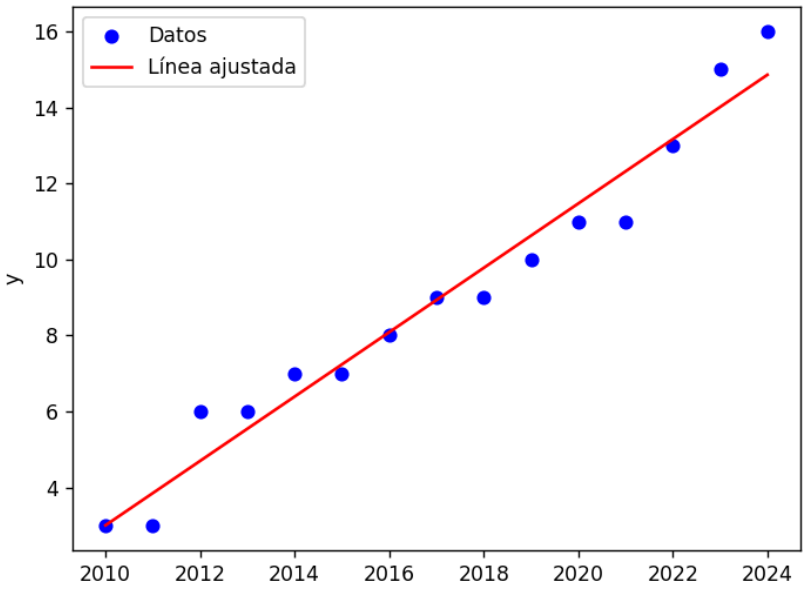
Es importante que considere realizar la menor cantidad de movimientos de datos posibles. Para lo cual se sugiere utilizar alguna estrategia de indexación. Es decir, utilización de índices como referencia a las posiciones en listas. Durante la presentación de este documento, el profesor brindará un ejemplo sobre esta estrategia.

# PARTE 2

La universidad está interesada en estimar la cantidad de estudiantes que habrá matriculados en los próximos semestres. Se requiere entonces utilizar la información de los semestres anteriores para construir un modelo estadístico que permita estimar el futuro.

## Regresión lineal

Una regresión lineal es una función lineal que pretende modelar el comportamiento de un conjunto de datos medidos de un sistema. Esta función lineal se puede usar como una aproximación al fenómeno real y permite estimar valores de la función para datos desconocidos. La siguiente figura muestra puntos azules que representan datos que fueron medidos o recolectados y en rojo una función lineal que describe de manera aproximada los datos reales.




Para el caso del problema propuesto, se tiene un archivo CSV con datos históricos de la cantidad de estudiantes matriculados en el programa académico cada año. Podemos entonces plantear una regresión lineal para esos datos de la siguiente forma:

$$y = ax + b$$

Donde $x$ es el año y $y$ es la cantidad de estudiantes matriculados. De otra parte, $a$ y $b$ son los parámetros que debemos encontrar para que la regresión lineal se ajuste lo mejor posible a los datos disponibles.

Para tener una manera objetiva de medir qué tan buena es una regresión lineal, podemos calcular el Mean Absolute Error (MAE) así:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Esta medida de error calcula la diferencia entre el dato real y el estimado por la regresión lineal, saca el valor absoluto y luego la media de todos los errores calculados. Entre menor sea el error, mejor es la regresión lineal y más se acerca el modelo a los datos reales.

Existen algoritmos para calcular valores apropiados para $a$ y $b$ pero no se estudiarán en este curso.

## Requerimientos del programa

Agregue una opción más al menú principal para predecir estudiantes matriculados en el futuro. Al seleccionar esta opción se le debe pedir un año en el futuro al usuario y el programa debe responder la **cantidad de estudiantes que estima estarán matriculados ese año** así como **el error del modelo**. También deberá generar un archivo que contenga **una gráfica cómo la mostrada en esta guía**, donde aparezcan los datos históricos, la regresión lineal y el punto específico que fue estimado de acuerdo a la solicitud del usuario.

Esta parte del programa debe cumplir con lo siguiente:

- Abrir el archivo **hist_matriculados.csv** y cargar los datos en dos listas.
- Una función que reciba una lista con valores $x$ (cantidad de estudiantes matriculados) y los parámetros $a$ y $b$ de la función lineal, y que retorne otra lista con los correspondientes valores $y$ calculados por la regresión lineal.
- Hacer uso de la función **plot_data()** que se encuentra en el módulo **plots.py** que acompaña esta guía.
- Los valores de $a$ y $b$ deben ser seleccionados por tanteo, mediante varios intentos donde se observe que el error disminuye y que la gráfica de la regresión lineal se ajusta razonablemente a los datos.