Datos y Matrices

Modelación de la ingeniería a través de la matemática computacional (TC1003B)

M.C. Xavier Sánchez Díaz
`sax@tec.mx`

Tecnológico de Monterrey

Outline

Tipos de datos

Datos y operaciones

Existen distintos tipos de datos con los que podemos trabajar en una computadora:

- Números enteros (*integer numbers*)
- Números decimales (*floating point numbers*)
- Cadenas de caracteres alfanuméricos (*strings*)

En este curso sólo nos preocuparemos por números.
Cuando **operamos** con estos datos, generamos nueva información que podríamos necesitar en el futuro.

Tipos de datos

Datos y operaciones

Existen distintos tipos de datos con los que podemos trabajar en una computadora:

- Números enteros (*integer numbers*)
- Números decimales (*floating point numbers*)
- Cadenas de caracteres alfanuméricos (*strings*)

En este curso sólo nos preocuparemos por números.

Cuando **operamos** con estos datos, generamos nueva información que podríamos necesitar en el futuro.

Tipos de datos

Datos y operaciones

Existen distintos tipos de datos con los que podemos trabajar en una computadora:

- Números enteros (*integer numbers*)
- Números decimales (*floating point numbers*)
- Cadenas de caracteres alfanuméricos (*strings*)

En este curso sólo nos preocuparemos por números.

Cuando **operamos** con estos datos, generamos nueva información que podríamos necesitar en el futuro.

Tipos de datos

Datos y operaciones

Existen distintos tipos de datos con los que podemos trabajar en una computadora:

- Números enteros (*integer numbers*)
- Números decimales (*floating point numbers*)
- Cadenas de caracteres alfanuméricos (*strings*)

En este curso sólo nos preocuparemos por números.

Cuando **operamos** con estos datos, generamos nueva información que podríamos necesitar en el futuro.

Tipos de datos

Datos y operaciones

Existen distintos tipos de datos con los que podemos trabajar en una computadora:

- Números enteros (*integer numbers*)
- Números decimales (*floating point numbers*)
- Cadenas de caracteres alfanuméricos (*strings*)

En este curso sólo nos preocuparemos por números.

Cuando **operamos** con estos datos, generamos nueva información que podríamos necesitar en el futuro.

Tipos de datos

Datos y operaciones

Existen distintos tipos de datos con los que podemos trabajar en una computadora:

- Números enteros (*integer numbers*)
- Números decimales (*floating point numbers*)
- Cadenas de caracteres alfanuméricos (*strings*)

En este curso sólo nos preocuparemos por números.
Cuando **operamos** con estos datos, generamos nueva información que podríamos necesitar en el futuro.

Datos como resultados

Datos y operaciones

Antes de usar la computadora o la calculadora para hacer cálculos, solíamos hacer las operaciones a mano.
Por ejemplo, si queremos calcular 1270×35, una manera de hacerlo podría ser. . .

Datos como resultados

Datos y operaciones

$$
\begin{aligned}
1270 \times 35 &= \\
&= (1200 + 70) \times (7)(5) \\
&= (12)(7)(5)(100) + (7)(7)(5)(10)
\end{aligned}
$$

$$
\begin{aligned}
12 \times 5 &= 60 \\
60 \times 7 = 6 \times 7 \times 10 &= 420 \\
420 \times 100 &= 42000
\end{aligned}
$$

$$
\begin{aligned}
7 \times 7 &= 49 \\
49 \times 10 &= 490 \\
490 \times 5 = 490 \times 10/2 &= 2450
\end{aligned}
$$

$$42000 + 2450 = 44450 \quad \square$$

Datos como resultados

Datos y operaciones

$$1270 \times 35 =$$
$$= (1200 + 70) \times (7)(5)$$
$$= (12)(7)(5)(100) + (7)(7)(5)(10)$$

$$12 \times 5 = 60$$
$$60 \times 7 = 6 \times 7 \times 10 = 420$$
$$420 \times 100 = 42000$$

$$7 \times 7 = 49$$
$$49 \times 10 = 490$$
$$490 \times 5 = 490 \times 10/2 = 2450$$

$$42000 + 2450 = 44450 \quad \square$$

Datos como resultados

Datos y operaciones

$$
\begin{aligned}
1270 \times 35 &= \\
&= (1200 + 70) \times (7)(5) \\
&= (12)(7)(5)(100) + (7)(7)(5)(10)
\end{aligned}
$$

$$
\begin{aligned}
12 \times 5 &= 60 \\
60 \times 7 = 6 \times 7 \times 10 &= 420 \\
420 \times 100 &= 42000
\end{aligned}
$$

$$
\begin{aligned}
7 \times 7 &= 49 \\
49 \times 10 &= 490 \\
490 \times 5 = 490 \times 10/2 &= 2450
\end{aligned}
$$

$$42000 + 2450 = 44450 \quad \square$$

Datos como resultados

Datos y operaciones

$$
\begin{aligned}
1270 \times 35 = & \\
&= (1200 + 70) \times (7)(5) \\
&= (12)(7)(5)(100) + (7)(7)(5)(10)
\end{aligned}
$$

$$
\begin{aligned}
12 \times 5 &= 60 \\
60 \times 7 = 6 \times 7 \times 10 &= 420 \\
420 \times 100 &= 42000
\end{aligned}
$$

$$
\begin{aligned}
7 \times 7 &= 49 \\
49 \times 10 &= 490 \\
490 \times 5 = 490 \times 10/2 &= 2450
\end{aligned}
$$

$$42000 + 2450 = 44450 \quad \square$$

Datos como resultados

Datos y operaciones

$$
\begin{aligned}
1270 \times 35 = & \\
& = (1200 + 70) \times (7)(5) \\
& = (12)(7)(5)(100) + (7)(7)(5)(10)
\end{aligned}
$$

$$
\begin{aligned}
12 \times 5 &= 60 \\
60 \times 7 = 6 \times 7 \times 10 &= 420 \\
420 \times 100 &= 42000
\end{aligned}
$$

$$
\begin{aligned}
7 \times 7 &= 49 \\
49 \times 10 &= 490 \\
490 \times 5 = 490 \times 10/2 &= 2450
\end{aligned}
$$

$$42000 + 2450 = 44450 \quad \square$$

Datos como resultados

Datos y operaciones

$$
\begin{aligned}
1270 \times 35 &= \\
&= (1200 + 70) \times (7)(5) \\
&= (12)(7)(5)(100) + (7)(7)(5)(10)
\end{aligned}
$$

$$
\begin{aligned}
12 \times 5 &= 60 \\
60 \times 7 = 6 \times 7 \times 10 &= 420 \\
420 \times 100 &= 42000
\end{aligned}
$$

$$
\begin{aligned}
7 \times 7 &= 49 \\
49 \times 10 &= 490 \\
490 \times 5 = 490 \times 10/2 &= 2450
\end{aligned}
$$

$$42000 + 2450 = 44450 \quad \square$$

Datos como resultados

Datos y operaciones

$$\begin{aligned} 1270 \times 35 = & \\ &= (1200 + 70) \times (7)(5) \\ &= (12)(7)(5)(100) + (7)(7)(5)(10) \end{aligned}$$

$$\begin{aligned} 12 \times 5 = 60 \\ 60 \times 7 = 6 \times 7 \times 10 = 420 \\ 420 \times 100 = 42000 \end{aligned}$$

$$\begin{aligned} 7 \times 7 = 49 \\ 49 \times 10 = 490 \\ 490 \times 5 = 490 \times 10/2 = 2450 \end{aligned}$$

$$42000 + 2450 = 44450 \quad \square$$

Datos como resultados

Datos y operaciones

$$
\begin{aligned}
1270 \times 35 &= \\
&= (1200 + 70) \times (7)(5) \\
&= (12)(7)(5)(100) + ({\color{red}7})({\color{red}7})(5)(10)
\end{aligned}
$$

$$
\begin{aligned}
12 \times 5 &= 60 \\
60 \times 7 = 6 \times 7 \times 10 &= 420 \\
420 \times 100 &= 42000
\end{aligned}
$$

$$
\begin{aligned}
{\color{red}7} \times {\color{red}7} &= 49 \\
49 \times 10 &= 490 \\
490 \times 5 = 490 \times 10/2 &= 2450
\end{aligned}
$$

$$42000 + 2450 = 44450 \quad \square$$

Datos como resultados

Datos y operaciones

$$
\begin{aligned}
1270 \times 35 &= \\
&= (1200 + 70) \times (7)(5) \\
&= (12)(7)(5)(100) + (7)(7)(5)(10)
\end{aligned}
$$

$$
\begin{aligned}
12 \times 5 &= 60 \\
60 \times 7 = 6 \times 7 \times 10 &= 420 \\
420 \times 100 &= 42000
\end{aligned}
$$

$$
\begin{aligned}
7 \times 7 &= 49 \\
49 \times 10 &= 490 \\
490 \times 5 = 490 \times 10/2 &= 2450
\end{aligned}
$$

$$42000 + 2450 = 44450 \quad \square$$

Datos como resultados

Datos y operaciones

$$
\begin{aligned}
1270 \times 35 = \\
&= (1200 + 70) \times (7)(5) \\
&= (12)(7)(5)(100) + (7)(7)(5)(10)
\end{aligned}
$$

$$
\begin{aligned}
12 \times 5 &= 60 \\
60 \times 7 = 6 \times 7 \times 10 &= 420 \\
420 \times 100 &= 42000
\end{aligned}
$$

$$
\begin{aligned}
7 \times 7 &= 49 \\
49 \times 10 &= 490 \\
490 \times 5 = 490 \times 10/2 &= 2450
\end{aligned}
$$

$$42000 + 2450 = 44450 \quad \square$$

Datos como resultados

Datos y operaciones

$$
\begin{aligned}
1270 \times 35 = \\
&= (1200 + 70) \times (7)(5) \\
&= (12)(7)(5)(100) + (7)(7)(5)(10)
\end{aligned}
$$

$$
\begin{aligned}
12 \times 5 &= 60 \\
60 \times 7 = 6 \times 7 \times 10 &= 420 \\
420 \times 100 &= 42000
\end{aligned}
$$

$$
\begin{aligned}
7 \times 7 &= 49 \\
49 \times 10 &= 490 \\
490 \times 5 = 490 \times 10/2 &= 2450
\end{aligned}
$$

$$42000 + 2450 = 44450 \quad \square$$

Datos como resultados

Datos y operaciones

La operación completa se hace poco a poco, y por tanto necesitamos "recordar" ciertos pasos intermedios que ya tenemos calculados.

Así como nosotros tenemos que tener en claro cuáles son esos pasos intermedios, la computadora debe saber *dónde está* la información que tiene que leer para trabajar y hacer cálculos más elaborados.

Para eso, podemos usar las estructuras de datos, para **ordenarlos** de manera conveniente y poder tener acceso a ellos de manera que se vayan necesitando.

Datos como resultados

Datos y operaciones

La operación completa se hace poco a poco, y por tanto necesitamos "recordar" ciertos pasos intermedios que ya tenemos calculados.

Así como nosotros tenemos que tener en claro cuáles son esos pasos intermedios, la computadora debe saber *dónde está* la información que tiene que leer para trabajar y hacer cálculos más elaborados.

Para eso, podemos usar las estructuras de datos, para **ordenarlos** de manera conveniente y poder tener acceso a ellos de manera que se vayan necesitando.

Datos como resultados

Datos y operaciones

La operación completa se hace poco a poco, y por tanto necesitamos "recordar" ciertos pasos intermedios que ya tenemos calculados.

Así como nosotros tenemos que tener en claro cuáles son esos pasos intermedios, la computadora debe saber *dónde está* la información que tiene que leer para trabajar y hacer cálculos más elaborados.

Para eso, podemos usar las estructuras de datos, para **ordenarlos** de manera conveniente y poder tener acceso a ellos de manera que se vayan necesitando.

Variables

Estructuras matemáticas

La forma más simple de guardar **un solo dato** es usando una variable.
En álgebra, hemos usado estas *variables* para expresar qué hace una función y saber su resultado:

$$2y = 2x + 5z + 6$$

- x es una variable que tiene algún valor que no conocemos en este momento
- y es otra variable (porque tiene distinto nombre) y tampoco sabemos su valor ahora, pero sabemos cómo calcularla
- z es otra variable más (porque tiene otro nombre distinto a los dos anteriores)

Variables

Estructuras matemáticas

La forma más simple de guardar **un solo dato** es usando una variable.
En álgebra, hemos usado estas *variables* para expresar qué hace una función y saber su resultado:

$$2y = 2x + 5z + 6$$

- x es una variable que tiene algún valor que no conocemos en este momento
- y es otra variable (porque tiene distinto nombre) y tampoco sabemos su valor ahora, pero sabemos cómo calcularla
- z es otra variable más (porque tiene otro nombre distinto a los dos anteriores)

Variables

Estructuras matemáticas

La forma más simple de guardar **un solo dato** es usando una variable.
En álgebra, hemos usado estas *variables* para expresar qué hace una función y saber su resultado:

$$2y = 2x + 5z + 6$$

- x es una variable que tiene algún valor que no conocemos en este momento
- y es otra variable (porque tiene distinto nombre) y tampoco sabemos su valor ahora, pero sabemos cómo calcularla
- z es otra variable más (porque tiene otro nombre distinto a los dos anteriores)

Variables

Estructuras matemáticas

La forma más simple de guardar **un solo dato** es usando una variable.
En álgebra, hemos usado estas *variables* para expresar qué hace una función y saber su resultado:

$$2y = 2x + 5z + 6$$

- x es una variable que tiene algún valor que no conocemos en este momento
- y es otra variable (porque tiene distinto nombre) y tampoco sabemos su valor ahora, pero sabemos cómo calcularla
- z es otra variable más (porque tiene otro nombre distinto a los dos anteriores)

Variables

Estructuras matemáticas

La forma más simple de guardar **un solo dato** es usando una variable.
En álgebra, hemos usado estas *variables* para expresar qué hace una función y saber su resultado:

$$2y = 2x + 5z + 6$$

- x es una variable que tiene algún valor que no conocemos en este momento
- y es otra variable (porque tiene distinto nombre) y tampoco sabemos su valor ahora, pero sabemos cómo calcularla
- z es otra variable más (porque tiene otro nombre distinto a los dos anteriores)

La forma más simple de guardar **un solo dato** es usando una variable.
En álgebra, hemos usado estas *variables* para expresar qué hace una función y saber su resultado:

$$2y = 2x + 5z + 6$$

- x es una variable que tiene algún valor que no conocemos en este momento
- y es otra variable (porque tiene distinto nombre) y tampoco sabemos su valor ahora, pero sabemos cómo calcularla
- z es otra variable más (porque tiene otro nombre distinto a los dos anteriores)

Variables

Estructuras matemáticas

$$2y = 2x + 5z + 6$$

Si ahora le damos valor a x y z, por ejemplo, $x = 3$ y $z = 2$ ...

- x guarda el valor de 3
- y guarda **2 veces** el resultado de $2(3) + 5(2) + 6 = 2(6 + 10 + 6) = 44$

Variables

Estructuras matemáticas

$$2y = 2x + 5z + 6$$

Si ahora le damos valor a x y z, por ejemplo, $x = 3$ y $z = 2$ ...

- x guarda el valor de 3
- y guarda **2 veces** el resultado de $2(3) + 5(2) + 6 = 2(6 + 10 + 6) = 44$

Variables

Estructuras matemáticas

$$2y = 2x + 5z + 6$$

Si ahora le damos valor a x y z, por ejemplo, $x = 3$ y $z = 2$ ...

- x guarda el valor de 3
- y guarda **2 veces** el resultado de $2(3) + 5(2) + 6 = 2(6 + 10 + 6) = 44$

Arreglos

Estructuras matemáticas

Asumamos que quiero saber las calificaciones de las Tareas 1, 2 y 3 de uno de mis alumnos. Para esto, necesitaría un lugar para guardar esos **3 datos**:

- $t_1 = 90$ será la variable para la Tarea 1
- $t_2 = 75$ será la variable para la Tarea 2
- $t_3 = 87$ será la variable para la Tarea 3

Con esta información, ahora contesta:

- ¿Cuál fue la calificación para la Tarea 2?
- ¿Cuál fue el promedio del alumno?
- ¿Cuál es la tarea en la que mejor le fue?

Arreglos

Estructuras matemáticas

Asumamos que quiero saber las calificaciones de las Tareas 1, 2 y 3 de uno de mis alumnos. Para esto, necesitaría un lugar para guardar esos **3 datos**:

- $t_1 = 90$ será la variable para la Tarea 1
- $t_2 = 75$ será la variable para la Tarea 2
- $t_3 = 87$ será la variable para la Tarea 3

Con esta información, ahora contesta:

- ¿Cuál fue la calificación para la Tarea 2?
- ¿Cuál fue el promedio del alumno?
- ¿Cuál es la tarea en la que mejor le fue?

Arreglos

Estructuras matemáticas

Asumamos que quiero saber las calificaciones de las Tareas 1, 2 y 3 de uno de mis alumnos. Para esto, necesitaría un lugar para guardar esos **3 datos**:

- $t_1 = 90$ será la variable para la Tarea 1
- $t_2 = 75$ será la variable para la Tarea 2
- $t_3 = 87$ será la variable para la Tarea 3

Con esta información, ahora contesta:

- ¿Cuál fue la calificación para la Tarea 2?
- ¿Cuál fue el promedio del alumno?
- ¿Cuál es la tarea en la que mejor le fue?

Arreglos

Estructuras matemáticas

Asumamos que quiero saber las calificaciones de las Tareas 1, 2 y 3 de uno de mis alumnos. Para esto, necesitaría un lugar para guardar esos **3 datos**:

- $t_1 = 90$ será la variable para la Tarea 1
- $t_2 = 75$ será la variable para la Tarea 2
- $t_3 = 87$ será la variable para la Tarea 3

Con esta información, ahora contesta:

- ¿Cuál fue la calificación para la Tarea 2?
- ¿Cuál fue el promedio del alumno?
- ¿Cuál es la tarea en la que mejor le fue?

Arreglos

Estructuras matemáticas

Asumamos que quiero saber las calificaciones de las Tareas 1, 2 y 3 de uno de mis alumnos. Para esto, necesitaría un lugar para guardar esos **3 datos**:

- $t_1 = 90$ será la variable para la Tarea 1
- $t_2 = 75$ será la variable para la Tarea 2
- $t_3 = 87$ será la variable para la Tarea 3

Con esta información, ahora contesta:

- ¿Cuál fue la calificación para la Tarea 2?
- ¿Cuál fue el promedio del alumno?
- ¿Cuál es la tarea en la que mejor le fue?

Arreglos

Estructuras matemáticas

Asumamos que quiero saber las calificaciones de las Tareas 1, 2 y 3 de uno de mis alumnos. Para esto, necesitaría un lugar para guardar esos **3 datos**:

- $t_1 = 90$ será la variable para la Tarea 1
- $t_2 = 75$ será la variable para la Tarea 2
- $t_3 = 87$ será la variable para la Tarea 3

Con esta información, ahora contesta:

- ¿Cuál fue la calificación para la Tarea 2?
- ¿Cuál fue el promedio del alumno?
- ¿Cuál es la tarea en la que mejor le fue?

Arreglos

Estructuras matemáticas

Asumamos que quiero saber las calificaciones de las Tareas 1, 2 y 3 de uno de mis alumnos. Para esto, necesitaría un lugar para guardar esos **3 datos**:

- $t_1 = 90$ será la variable para la Tarea 1
- $t_2 = 75$ será la variable para la Tarea 2
- $t_3 = 87$ será la variable para la Tarea 3

Con esta información, ahora contesta:

- ¿Cuál fue la calificación para la Tarea 2?
- ¿Cuál fue el promedio del alumno?
- ¿Cuál es la tarea en la que mejor le fue?

La pregunta ahora es…¿Realmente necesito **3 variables** para guardar **3 datos**? Podemos *arreglar* los datos de tal manera que **su posición** nos aporte algo más:

$$\mathbf{t} = \langle 90, 75, 87 \rangle$$

La **posición** en esta *lista ordenada* nos indica qué número de tarea fue, y el valor que haya en dicha posición guarda la calificación. Por lo mismo, podemos usar "una sola variable" para guardar de manera ordenada la información requerida, y referirnos sólo a la posición deseada:

$$\mathbf{t}_2 = 75$$

La pregunta ahora es. . . ¿Realmente necesito **3 variables** para guardar **3 datos**? Podemos *arreglar* los datos de tal manera que **su posición** nos aporte algo más:

$$\mathbf{t} = \langle 90, 75, 87 \rangle$$

La **posición** en esta *lista ordenada* nos indica qué número de tarea fue, y el valor que haya en dicha posición guarda la calificación. Por lo mismo, podemos usar "una sola variable" para guardar de manera ordenada la información requerida, y referirnos sólo a la posición deseada:

$$\mathbf{t}_2 = 75$$

La pregunta ahora es. . . ¿Realmente necesito **3 variables** para guardar **3 datos**? Podemos *arreglar* los datos de tal manera que **su posición** nos aporte algo más:

$$\mathbf{t} = \langle 90, 75, 87 \rangle$$

La **posición** en esta *lista ordenada* nos indica qué número de tarea fue, y el valor que haya en dicha posición guarda la calificación. Por lo mismo, podemos usar "una sola variable" para guardar de manera ordenada la información requerida, y referirnos sólo a la posición deseada:

$$\mathbf{t}_2 = 75$$

Cuando *arreglamos* los datos para usarlos de manera lineal, estamos creando una **lista**.

$$\mathbf{x} = \langle 1, 2, 10, 23, 2, -1, 70, 15 \rangle$$

- Usamos negritas para denotar la diferencia entre la variable x que guarda un valor, y la variable $\mathbf{x}$ que guarda **múltiples** valores
- Usamos *angle brackets* (*cuñas* les dicen en español) para delimitar la lista de sus valores
- A diferencia de un conjunto, el orden de los valores **sí importa**

Cuando *arreglamos* los datos para usarlos de manera lineal, estamos creando una **lista**.

$$\mathbf{x} = \langle 1, 2, 10, 23, 2, -1, 70, 15 \rangle$$

- Usamos negritas para denotar la diferencia entre la variable x que guarda un valor, y la variable $\mathbf{x}$ que guarda **múltiples** valores
- Usamos *angle brackets* (*cuñas* les dicen en español) para delimitar la lista de sus valores
- A diferencia de un conjunto, el orden de los valores **sí importa**

Cuando *arreglamos* los datos para usarlos de manera lineal, estamos creando una **lista**.

$$\mathbf{x} = \langle 1, 2, 10, 23, 2, -1, 70, 15 \rangle$$

- Usamos negritas para denotar la diferencia entre la variable x que guarda un valor, y la variable $\mathbf{x}$ que guarda **múltiples** valores
- Usamos *angle brackets* (*cuñas* les dicen en español) para delimitar la lista de sus valores
- A diferencia de un conjunto, el orden de los valores **sí importa**

Cuando *arreglamos* los datos para usarlos de manera lineal, estamos creando una **lista**.

$$\mathbf{x} = \langle 1, 2, 10, 23, 2, -1, 70, 15 \rangle$$

- Usamos negritas para denotar la diferencia entre la variable x que guarda un valor, y la variable $\mathbf{x}$ que guarda **múltiples** valores
- Usamos *angle brackets* (*cuñas* les dicen en español) para delimitar la lista de sus valores
- A diferencia de un conjunto, el orden de los valores **sí importa**

Matrices

Estructuras matemáticas

Supongamos que ahora necesito saber las calificaciones de las tres tareas, pero ahora de **varios alumnos**.

Esto significa que ahora necesitamos **varias listas**, pero en su lugar podemos *arreglar* los datos como **una secuencia de tareas** (*la tarea 1, la tarea 2. . .*). A su vez, cada una de las tareas tiene una **secuencia de alumnos** (*el alumno 1, el alumno 2. . .*)

$$\begin{array}{c c} \begin{array}{c} alumno_1 \\ alumno_2 \\ alumno_3 \\ alumno_4 \end{array} & \begin{bmatrix} 90 & 75 & 87 \\ 100 & 100 & 95 \\ 90 & 70 & 88 \\ 85 & 65 & 50 \end{bmatrix} \end{array}$$

Matrices

Estructuras matemáticas

Supongamos que ahora necesito saber las calificaciones de las tres tareas, pero ahora de **varios alumnos**.
Esto significa que ahora necesitamos **varias listas**, pero en su lugar podemos *arreglar* los datos como **una secuencia de tareas** (*la tarea 1, la tarea 2. . .*).
A su vez, cada una de las tareas tiene una **secuencia de alumnos** (*el alumno 1, el alumno 2. . .*)

$$\begin{array}{c c} \begin{array}{c} alumno_1 \\ alumno_2 \\ alumno_3 \\ alumno_4 \end{array} & \begin{bmatrix} 90 & 75 & 87 \\ 100 & 100 & 95 \\ 90 & 70 & 88 \\ 85 & 65 & 50 \end{bmatrix} \end{array}$$

Supongamos que ahora necesito saber las calificaciones de las tres tareas, pero ahora de **varios alumnos**.
Esto significa que ahora necesitamos **varias listas**, pero en su lugar podemos *arreglar* los datos como **una secuencia de tareas** (*la tarea 1, la tarea 2...*). A su vez, cada una de las tareas tiene una **secuencia de alumnos** (*el alumno 1, el alumno 2...*)

$$\begin{array}{c c} \begin{array}{c} alumno_1 \\ alumno_2 \\ alumno_3 \\ alumno_4 \end{array} & \begin{bmatrix} 90 & 75 & 87 \\ 100 & 100 & 95 \\ 90 & 70 & 88 \\ 85 & 65 & 50 \end{bmatrix} \end{array}$$

Matrices

Estructuras matemáticas

$$A = \begin{bmatrix} 90 & 75 & 87 \\ 100 & 100 & 95 \\ 90 & 70 & 88 \\ 85 & 65 & 50 \end{bmatrix}$$

- Una matriz es una lista de columnas
- Solemos usar mayúsculas para los nombres de las matrices
- En este caso, A tiene 4 renglones y 3 columnas, es decir que es de 4×3
- El renglón A_2 es $\langle 100, 100, 95 \rangle$
- El elemento $A_{3,2}$ es 70

Matrices

Estructuras matemáticas

$$A = \begin{bmatrix} 90 & 75 & 87 \\ 100 & 100 & 95 \\ 90 & 70 & 88 \\ 85 & 65 & 50 \end{bmatrix}$$

- Una matriz es una lista de columnas
- Solemos usar mayúsculas para los nombres de las matrices
- En este caso, A tiene 4 renglones y 3 columnas, es decir que es de 4×3
- El renglón A_2 es $\langle 100, 100, 95 \rangle$
- El elemento $A_{3,2}$ es 70

$$A = \begin{bmatrix} 90 & 75 & 87 \\ 100 & 100 & 95 \\ 90 & 70 & 88 \\ 85 & 65 & 50 \end{bmatrix}$$

- Una matriz es una lista de columnas
- Solemos usar mayúsculas para los nombres de las matrices
- En este caso, A tiene 4 renglones y 3 columnas, es decir que es de 4×3
- El renglón A_2 es $\langle 100, 100, 95 \rangle$
- El elemento $A_{3,2}$ es 70

Matrices

Estructuras matemáticas

$$A = \begin{bmatrix} 90 & 75 & 87 \\ 100 & 100 & 95 \\ 90 & 70 & 88 \\ 85 & 65 & 50 \end{bmatrix}$$

- Una matriz es una lista de columnas
- Solemos usar mayúsculas para los nombres de las matrices
- En este caso, A tiene 4 renglones y 3 columnas, es decir que es de 4×3
- El renglón A_2 es $\langle 100, 100, 95 \rangle$
- El elemento $A_{3,2}$ es 70

Matrices

Estructuras matemáticas

$$A = \begin{bmatrix} 90 & 75 & 87 \\ 100 & 100 & 95 \\ 90 & 70 & 88 \\ 85 & 65 & 50 \end{bmatrix}$$

- Una matriz es una lista de columnas
- Solemos usar mayúsculas para los nombres de las matrices
- En este caso, A tiene 4 renglones y 3 columnas, es decir que es de 4×3
- El renglón A_2 es $\langle 100, 100, 95 \rangle$
- El elemento $A_{3,2}$ es 70

Cuando una matriz es de $1 \times n$ es una matriz renglón:

$$B = \begin{bmatrix} 100 & 95 & 89 & 92 \end{bmatrix}$$

Cuando una matriz es de $n \times 1$ es una matriz columna o vector:

$$\mathbf{c} = \begin{bmatrix} 100 \\ 91 \\ 95 \\ 98 \\ 85 \end{bmatrix}$$

Cuando una matriz es de $1 \times n$ es una matriz renglón:

$$B = \begin{bmatrix} 100 & 95 & 89 & 92 \end{bmatrix}$$

Cuando una matriz es de $n \times 1$ es una matriz columna o vector:

$$\mathbf{c} = \begin{bmatrix} 100 \\ 91 \\ 95 \\ 98 \\ 85 \end{bmatrix}$$

Matrices

Estructuras matemáticas

$$A = \begin{bmatrix} 10 & 15 & 2 & 4 \\ -2 & 8 & 20 & 7 \\ 1 & 5 & 13 & 14 \\ 20 & 4 & 2 & 10 \end{bmatrix}$$

- La diagonal principal hace referencia a cada elemento a_{ii}
- Si todo **abajo** de la diagonal principal es 0, entonces le llamamos matriz triangular superior
- Si todo **arriba** de la diagonal principal es 0, entonces le llamamos matriz triangular inferior

Matrices

Estructuras matemáticas

$$A = \begin{bmatrix} 10 & 15 & 2 & 4 \\ 0 & 8 & 20 & 7 \\ 0 & 0 & 13 & 14 \\ 0 & 0 & 0 & 10 \end{bmatrix}$$

- La diagonal principal hace referencia a cada elemento a_{ii}
- Si todo **abajo** de la diagonal principal es 0, entonces le llamamos matriz triangular superior
- Si todo **arriba** de la diagonal principal es 0, entonces le llamamos matriz triangular inferior

$$A = \begin{bmatrix} 10 & 0 & 0 & 0 \\ -2 & 8 & 0 & 0 \\ 1 & 5 & 13 & 0 \\ 20 & 4 & 2 & 10 \end{bmatrix}$$

- La diagonal principal hace referencia a cada elemento a_{ii}
- Si todo **abajo** de la diagonal principal es 0, entonces le llamamos matriz triangular superior
- Si todo **arriba** de la diagonal principal es 0, entonces le llamamos matriz triangular inferior

¿Y qué puedo hacer con las matrices?

Operaciones matriciales

Podemos sumar dos matrices si tienen **las mismas dimensiones**:

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & 7 & 2 & 9 \\ 2 & 2 & 3 & 1 \end{bmatrix} + \begin{bmatrix} 3 & 7 & 2 & 0 \\ 7 & -2 & 8 & 5 \\ 1 & 3 & -8 & 2 \end{bmatrix} = \begin{bmatrix} 4 & 9 & 5 & 4 \\ 7 & 5 & 10 & 14 \\ 3 & 5 & -5 & 3 \end{bmatrix}$$

Suma de Matrices

Operaciones matriciales

En general ...

$$\begin{bmatrix} a_{11} & \dots & \dots & a_{1j} \\ \vdots & \ddots & \ddots & \vdots \\ a_{i1} & \dots & \dots & a_{ij} \end{bmatrix} + \begin{bmatrix} b_{11} & \dots & \dots & b_{1j} \\ \vdots & \ddots & \ddots & \vdots \\ b_{i1} & \dots & \dots & b_{ij} \end{bmatrix} =$$

$$\begin{bmatrix} a_{11}+b_{11} & \dots & \dots & a_{1j}+b_{1j} \\ \vdots & \ddots & \ddots & \vdots \\ a_{i1}+b_{i1} & \dots & \dots & a_{ij}+b_{ij} \end{bmatrix}$$

Producto por un escalar

Operaciones matriciales

Podemos multiplicar la matriz por **un escalar** (o sea, una cantidad):

$$2\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix} = \begin{bmatrix} 2 & 4 & 6 & 8 \\ 10 & 12 & 14 & 16 \\ 18 & 20 & 22 & 24 \\ 26 & 28 & 30 & 32 \end{bmatrix}$$

En este caso, estamos **escalando** la matriz por 2. ¿Qué pasa si multiplicamos por $\frac{2}{3}$?

Producto por escalar

Operaciones matriciales

En general...

$$c \begin{bmatrix} a_{11} & \dots & \dots & a_{1j} \\ \vdots & \ddots & \ddots & \vdots \\ a_{i1} & \dots & \dots & a_{ij} \end{bmatrix} =$$

$$\begin{bmatrix} ca_{11} & \dots & \dots & ca_{1j} \\ \vdots & \ddots & \ddots & \vdots \\ ca_{i1} & \dots & \dots & ca_{ij} \end{bmatrix}$$

Matriz por vector

yes indeed

De uno en uno...

Operaciones vectorizadas

Los **arreglos** (ya sean vectores o matrices) tienen, por sí solos, una especie de orden. Este orden da pie a pensar en una **secuencia**, y entonces operar usando **ciclos** es *natural*:

```
% sum 10 to each number
x = [1 2 3 4 5];

for i = x
  r(i) = x(i) + 10;
end

disp(r)
```

¿Cuál es el resultado de `r`?

Sin embargo, existen ciertas operaciones que están pensadas para operar directamente sobre **vectores**, y funcionan justo como lo esperaríamos:

Command Window

```
>> x = [1 2 3 4 5];
>> x + 10
ans =
   11  12  13  14  15
```

Estas operaciones son conocidas como operaciones vectorizadas, y trabajan con **cada uno** de los valores al mismo tiempo, en lugar de uno por uno.