

# Github Error Notes

Annie Zhang

01-01-2024

## Índice

1. git error: failed to push some refs to remote	1
2. Tipos de Matrices	2
3. Operaciones con Matrices	4
3.1. Suma o adición . . . . .	4
3.2. Producto por un escalar . . . . .	5
3.3. Producto Matricial . . . . .	6
4. Matrices Binarias y Grafos	8
4.1. Relaciones binarias . . . . .	8
4.2. Grafos . . . . .	8
4.3. Representación de grafos . . . . .	10
5. Matriz Jacobiana	11
5.1. Función escalar . . . . .	11
5.2. Función vectorial . . . . .	11
5.3. Ejemplos . . . . .	12
6. Maxima	12

## Resumen

En este documento se explican qué son las matrices, los distintos tipos que hay, algunas operaciones que se hacen con ellas (y sus respectivas propiedades), además de enseñar cómo realizar estas operaciones en programación. Incluye el concepto de relación binaria, y se explica cómo se pueden representar éstas (si son homogéneas) mediante grafos, y los distintos tipos de éstos últimos. Se recalca la "matriz jacobiana", explicando su función escalar y vectorial

## 1. git error: failed to push some refs to remote

When I use `git push origin master`, I get an error:

Listing 1: Error message example

```
$ git remote -v
origin https://github.com/REDACTED.git (fetch)
origin https://github.com/REDACTED.git (push)

$ git push origin master
Username for 'https://github.com': REDACTED
Password for 'https://REDACTED@github.com':
To https://github.com/REDACTED.git
```

```
! [rejected]      master -> master (non-fast-forward)
error: failed to push some refs to 'https://github.com/REDACTED.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Integrate the remote changes (e.g.
hint: 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

---

Note: starting Oct. 2020, any new repository is created with the default branch main, not master. And you can rename existing repository default branch from master to main

---

Listing 2: commands to fix

---

```
git pull --rebase
git push
```

---

- Cada uno de los números de que consta la matriz se denomina **elemento**. Un elemento se distingue de otro por la posición que ocupa, es decir, la fila y la columna a la que pertenece. Para ello se usa un doble subíndice donde el primero indica la fila y el segundo la columna a la que pertenece.

Una forma simplificada de representar una matriz es  $A := ((a_{ij}))$

- El número de filas y columnas de una matriz se denomina **dimensión de una matriz**. A una matriz con  $n$  filas y  $m$  columnas se le denomina matriz  $n$ -por- $m$  (escrito  $n \times m$ ) donde  $n, m \in \mathbb{N} - \{0\}$ .
- Si la matriz tiene el mismo número de filas que de columnas, y es  $n$ , se dice que es de **orden**  $n$ .
- Dos matrices son **iguales** cuando tienen la misma dimensión y los elementos que ocupan el mismo lugar en ambas, son iguales.
- El conjunto de las matrices de tamaño  $n \times m$  se representa como  $\mathcal{M}_{n \times m}(\mathbb{K})$  donde  $\mathbb{K}$  es el campo al cual pertenecen las entradas. El tamaño de una matriz siempre se da con el número de filas primero y el número de columnas después.

## 2. Tipos de Matrices

**Matriz fila:** Una matriz fila está constituida por una sola fila.

$$[a_{11} \quad a_{12} \quad \cdots \quad a_{1m}]$$

**Matriz columna:** La matriz columna tiene una sola columna.

$$\begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{bmatrix}$$

**Matriz rectangular:** La matriz rectangular tiene distinto número de filas que de columnas, siendo su dimensión  $n \times m$ . (Ver expresión (1)).

**Matriz traspuesta:** Dada una matriz  $A$ , se llama matriz traspuesta de  $A$  a la matriz que se obtiene cambiando ordenadamente las filas por las columnas y se denota por  $A^t$ .

Si  $A = ((a_{ij}))$ , entonces  $A^t = ((a_{ji}))$

Cumplen las siguiente propiedades:

- $(A^t)^t = A$ .

- $(A + B)^t = A^t + B^t$ .
- $(\alpha \cdot)^t = \alpha \cdot A^t$
- $(A \times B)^t = B^t \times A^t$ .

Las operaciones con matrices se definen en el apartado 3.

**Matriz nula:** En una matriz nula todos los elementos son ceros.  $A = ((a_{ij})) = ((0))$

**Matriz cuadrada:** La matriz cuadrada tiene el mismo número de filas que de columnas.

- Los elementos de la forma  $((a_{ii}))$  constituyen **la diagonal principal**.
- La **diagonal secundaria** la forman los elementos con  $i + j = n + 1$ , siendo  $n$  el orden de la matriz.
- Existen los siguientes tipos de matrices cuadradas<sup>1</sup>

*Matriz triangular superior:* En una matriz triangular superior los elementos situados por debajo de la diagonal principal son ceros.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ 0 & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nm} \end{bmatrix}$$

*Matriz triangular inferior:* En una matriz triangular inferior los elementos situados por encima de la diagonal principal son ceros.

$$A = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

*Matriz diagonal:* En una matriz diagonal todos los elementos que no están situados en la diagonal principal son nulos.

$$A = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nm} \end{bmatrix}$$

*Matriz escalar:* Una matriz escalar es una matriz diagonal en la que los elementos de la diagonal principal son iguales.

*Matriz identidad o unidad:* Una matriz identidad es una matriz diagonal en la que los elementos de la diagonal principal son iguales a 1.

*Matriz regular:* Una matriz regular es una matriz cuadrada que tiene inversa.

*Matriz singular:* Una matriz singular no tiene matriz inversa.

*Matriz idempotente:* Una matriz,  $A$ , es idempotente si:  $A^2 = A$ .

*Matriz involutiva:* Una matriz,  $A$ , es involutiva si:  $A^2 = I$ .

*Matriz simétrica:* Una matriz simétrica es una matriz cuadrada que verifica:  $A = A^t$ . *Matriz anti-simétrica o hemisimétrica:* Una matriz antisimétrica o hermisimétrica es una matriz cuadrada que verifica  $A = -A^t$ .

*Matriz ortogonal:* Una matriz es ortogonal si verifica que:  $A \times A^t = I$

---

<sup>1</sup>Las operaciones con matrices se definen en el apartado 3.

### 3. Operaciones con Matrices

Las operaciones que se pueden hacer con matrices provienen de sus aplicaciones, sobre todo de las aplicaciones en álgebra lineal. De ese modo las operaciones, o su forma muy particular de ser implementadas, no son únicas.

#### 3.1. Suma o adición

**git error: failed to push some refs to remote**

Se define la operación de suma o adición de matrices como una operación binaria

$$+ : \mathcal{M}_{n \times m}(\mathbb{K}) \times \mathcal{M}_{n \times m}(\mathbb{K}) \rightarrow \mathcal{M}_{n \times m}(\mathbb{K})$$

tal que  $(A, B) \mapsto C = A + B$  y donde  $c_{ij} = a_{ij} + b_{ij}$  en el que la operación de suma en la última expresión es la operación binaria correspondiente pero en el campo  $\mathbb{K}$ .

#### Ejemplo

Sea  $A, B \in \mathcal{M}_3(\mathbb{R})$

$$\begin{bmatrix} 1 & 3 & 2 \\ 1 & 0 & 0 \\ 1 & 2 & 2 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 5 \\ 7 & 5 & 0 \\ 2 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1+1 & 3+0 & 2+5 \\ 1+7 & 0+5 & 0+0 \\ 1+2 & 2+1 & 2+1 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 7 \\ 8 & 5 & 0 \\ 3 & 3 & 3 \end{bmatrix}$$

La entrada  $c_{12}$  es igual a la suma de los elementos  $a_{12}$  y  $b_{12}$  lo cual es  $a_{12} + b_{12}$ .

#### Propiedades

- *Asociatividad:*  $(A + B) + C = A + (B + C)$
- *Conmutatividad:*  $(A + B) = (B + A)$
- *Existencia del elemento neutro aditivo:* Existe  $0 \in \mathcal{M}_{n \times m}(\mathbb{K})$  tal que  $A + 0 = 0 + A = A$
- *Existencia del inverso aditivo:* Existe  $D \in \mathcal{M}_{n \times m}(\mathbb{K})$  tal que  $A + D = 0$  a esta matriz  $D$  se le denota por  $-A$

#### Estructura algebraica

La operación binaria adición se dice que esta operación es una operación interna por lo que se cumple intrínsecamente la propiedad de que  $\mathcal{M}_{n \times m}(\mathbb{K})$  es cerrado bajo adición. Con estas propiedades se tiene que  $(\mathcal{M}_{n \times m}(\mathbb{K}), +)$  es un grupo abeliano.

En el caso en que el conjunto al que pertenecen las entradas de la matriz sea un anillo  $(A, +_A, \cdot_A)$ , la operación de adición de matrices continúa dotando de estructura de grupo abeliano a  $(\mathcal{M}_{n \times m}(A), +)$ , ya que bajo un anillo  $(A, +_A, \cdot_A)$  se tiene que  $(A, +_A)$  es un grupo abeliano. En el caso de que las entradas estén en un grupo  $(G, +_G)$ , éste necesita ser un grupo abeliano para que la adición de matrices siga dotando de estructura de grupo abeliano a  $(\mathcal{M}_{n \times m}(G), +)$

#### Programación

En pseudocódigo basado en Java, esta operación es:

Listado 1: Suma de Matrices

```
1 void main() {
2     float [] a = creaMatriz(n, m); // Capturamos los valores de la matriz A
3     float [] b = creaMatriz(n, m); // Capturamos los valores de la matriz A
4     float [] suma = suma(a, b); ; // Invocamos a la función que suma matrices.
5 }
7 // La suma de dos matrices es otra matriz
```

```

8 float [][] suma(float[][] matriz1, float [][] matriz2) {
9     // Si no hay matrices no hay nada que hacer.
10    if (matriz1 == null || matriz2 == null) return null;

12    // Asumimos que el primer índice es la fila y el segundo es la columna.
13    // El número de filas y columnas en ambas matrices tienen que ser iguales.
14    if (matriz1.length != matriz2.length) return null; // Filas diferentes.
15    else
16        for (int i=0; i<matriz1.length; i++) {
17            if (matriz1[0].length != matriz1[i].length)
18                return null; // Alguna fila tiene un número diferente de columnas

20            if (matriz1[i].length != matriz2[i].length)
21                return null; // matrices tienen columnas diferentes.
22        }
23    // Ya estamos en condiciones de empezar

25    // Primero creamos la nueva matriz
26    // Tiene tantas filas y columnas, como la primera (o segunda) matriz.
27    float [][] sol = new float[matriz1.length][matriz1[0].length];

29    // Completamos las celdillas de la nueva matriz con el típico bucle anidado.
30    for (int i=0; i<sol.length; i++)
31        for (int j=0; j<sol[i].length; j++)

33        sol[i][j] = matriz1[i][j]+matriz2[i][j]; // En esta celdilla calcularemos la suma.
34    return sol;
35 }

```

## 3.2. Producto por un escalar

### Definición

Sea  $A \in (\mathcal{M}_{n \times m}(\mathbb{K}))$  y  $\lambda \in \mathbb{K}$ . Se define la operación de producto por un escalar como una función

$$\cdot : \mathbb{K} \times \mathcal{M}_{n \times m}(\mathbb{K}) \longrightarrow \mathcal{M}_{n \times m}(\mathbb{K})$$

tal que  $(\lambda, A) \mapsto B = \lambda A$  y donde  $b_{ij} = \lambda a_{ij}$  en donde el producto es la operación binaria correspondiente pero en el campo  $\mathbb{K}$

### Ejemplo

Sea  $A \in \mathcal{M}_{2 \times 3}(\mathbb{R})$  y  $2 \in \mathbb{R}$

$$2 \begin{bmatrix} 1 & 8 & -3 \\ 4 & -2 & 6 \end{bmatrix} = \begin{bmatrix} 2(1) & 2(8) & 2(-3) \\ 2(4) & 2(-2) & 2(6) \end{bmatrix} = \begin{bmatrix} 2 & 16 & -6 \\ 8 & -4 & 12 \end{bmatrix}$$

Por ejemplo, la entrada  $b_{12}$  es igual al producto  $\lambda a_{12}$ .

### Propiedades

Sean  $A, B \in (\mathcal{M}_{n \times m}(\mathbb{K}))$  y  $\lambda, \mu \in \mathbb{K}$ , donde  $\mathbb{K}$  es un campo, entonces se cumplen las siguientes propiedades para la operación producto por un escalar

- *Asociatividad:*  $(\lambda\mu)A = \lambda(\mu A)$
- *Distributividad respecto de la suma de matrices:*  $\lambda(A + B) = \lambda A + \lambda B$
- *Distributividad respecto de la suma en el campo:*  $(\lambda + \mu)A = \lambda A + \mu A$
- *Producto por el neutro multiplicativo del campo:*  $1_{\mathbb{K}}A = A$

### Estructura algebraica

Por como se definió la operación de producto por escalares se dice que  $\mathcal{M}_{n \times m}(\mathbb{K})$  es cerrado bajo producto por escalares. Con estas propiedades y las de la adición se tiene que  $\mathcal{M}_{n \times m}(\mathbb{K})$  es un espacio vectorial con las operaciones de suma y producto por escalares definidas antes.

En el caso de que las entradas y los escalares no estén en un campo sino en un anillo entonces no necesariamente existe el neutro multiplicativo. En caso de que exista, con lo cual el anillo es un anillo con uno, se dice que  $\mathcal{M}_{n \times m}(A)$  es un módulo sobre  $A$ .

## Programación

En pseudocódigo basado en Java, esta operación es:

Listado 2: Suma de Matrices

```

1 void main() {
2     float [][] a = creaMatriz(n, m); // Capturamos los valores de la matriz A
3     float L = leeCte; // Capturamos el valor de la constante.
4     float [][] sol = productoEscalarMatriz (L, a); // Calculamos el producto.
5 }

7 // La suma de dos matrices es otra matriz
8 float [][] suma(float lambda, float [][] matriz) {
9     // Si no hay matrices no hay nada que hacer.
10    if (matriz1 == null) return null;

12    // Primero creamos la nueva matriz
13    // Tiene tantas filas y columnas, como la primera (o segunda) matriz.
14    float [][] sol = new float[matriz.length][matriz[0].length];

16    // Completamos las celdillas de la nueva matriz con el típico bucle anidado.
17    for (int i=0; i<sol.length; i++)
18        for (int j=0; j<sol[i].length; j++)

20        sol[i][j] = lambda * matriz[i][j]; // En esta celdilla calcularemos el producto

22    return sol;
23 }
```

## 3.3. Producto Matricial

Su definición depende del contexto. Cuando proviene de la composición de aplicaciones lineales el tamaño de la matriz corresponde con las dimensiones de los espacios vectoriales entre los cuales se establece la aplicación lineal. De ese modo el producto de matrices representa la composición de aplicaciones lineales.

En efecto, en ciertas bases tenemos que  $f : V \rightarrow W$  se puede representar como  $f(x) = Ax$  donde  $x$  es la representación de un vector de  $V$  en la base que se ha elegido para  $V$  en forma de vector columna. Si tenemos dos aplicaciones lineales  $f : V \rightarrow W$  y  $g : W \rightarrow U$  entonces  $f(x) = g(f(x)) = g(Bx) = AB(x)$  donde  $AB$  es el producto de las representaciones matriciales de  $f, g$ . Nótese que la composición no se puede dar entre cualquier aplicación sino entre aplicaciones que vaya de  $V \rightarrow W \rightarrow U$ , en particular debe de haber una relación entre las dimensiones de los espacios vectoriales. Una vez dicho esto podemos definir el producto de la siguiente manera.

### Definición

Sean  $A \in \mathcal{M}_{n \times m}(K)$  y  $B \in \mathcal{M}_{m \times p}(K)$ . Se define el producto de matrices como una función

$$\cdot : \mathcal{M}_{n \times m}(K) \times \mathcal{M}_{m \times p}(K) \rightarrow \mathcal{M}_{n \times p}(K)$$

tal que  $(A, B) \mapsto C = AB$  y donde  $c_{ij} = \sum_{k=1}^m a_{ik}b_{kj}$  para toda  $i, j$ , es decir

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + a_{i3}b_{3j} + \dots + a_{im}b_{mj}$$

## Ejemplo

Sean  $A \in \mathcal{M}_{2 \times 3}(R)$  y  $B \in \mathcal{M}_{3 \times 2}(R)$ :

$$\begin{bmatrix} 2 & 0 & -2 \\ -1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 2 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1(3) + 0(2) + 2(1) & 1(1) + 0(1) + 2(0) \\ -1(3) + 3(2) + 1(1) & -1(1) + 3(1) + 1(0) \end{bmatrix} = \begin{bmatrix} 5 & 1 \\ 4 & 2 \end{bmatrix}$$

donde la matriz es como habíamos establecido en la definición: una matriz  $Cin \mathcal{M}_{2 \times 2}(R)$ . Por ejemplo, la entrada  $c_{12} = a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} + \dots + a_{1m}b_{m2}$

## Propiedades

- *Asociatividad:*  $A(BC) = (AB)C$
- *Distributividad respecto de la suma de matrices por la izquierda:*  $A(B + C) = AB + AC$

## Estructura algebraica

El producto de matrices no es conmutativo, si lo fuera la composición de funciones lineales sería conmutativa y eso en general no sucede. Obviamente existen casos particulares de algunos tipos de matrices en los que sí hay conmutatividad. En el caso de que tengamos  $\mathcal{M}_n(\mathbb{K})$  tendremos que el producto entre matrices en  $\mathcal{M}_n(\mathbb{K})$  también está en  $\mathcal{M}_n(\mathbb{K})$ . En ese caso,  $\mathcal{M}_n(\mathbb{K})$  además de espacio vectorial es un álgebra sobre un campo. En el caso de que el conjunto al que pertenecen las entradas sea un anillo conmutativo con uno, entonces  $\mathcal{M}_n(A)$  además de módulo es un álgebra sobre un anillo. Más aún  $(\mathcal{M}_n(\mathbb{K}), +, \cdot)$  con  $\cdot$  el producto de matrices es un anillo.

## Programación

En pseudocódigo basado en Java, esta operación es:

Listado 3: Suma de Matrices

```
1 void main() {
2     float [][] a = creaMatriz(n, m); // Capturamos los valores de la matriz A
3     float [][] b = creaMatriz(m, k); // Capturamos los valores de la matriz B
4     float [][] sol = producto(a, b); // Calculamos el producto
5 }

6
7 // El producto de dos matrices es otra matriz
8 float [][] producto(float [][] matriz1, float [][] matriz2) {
9     // Si no hay matrices no hay nada que hacer.
10    if (matriz1==null || matriz2==null) return null;

11
12    // Asumimos que el primer índice es la fila y el segundo es la columna.
13    // El número de columnas de la primera tienen que ser igual al número
14    // de filas de la primera.
15    if (matriz1[0].length != matriz2.length) return null;

16
17    // Aquí habría que poner el código para comprobar que todos los vectores
18    // tienen la misma longitud. Ver el código de cómo sumar (ahí está desarrollado).

19
20    // Ya estamos en condiciones de empezar

21
22    // Primero creamos la nueva matriz
23    // Tiene tantas filas, como filas tiene la primera.
24    // Tiene tantas columnas, como columnas tiene la segunda.
25    // Asumimos que matriz2[i]==matriz2[0]
26    float [][] sol = new float[matriz1.length][matriz2[0].length];

27
28    // Completamos las celdillas de la nueva matriz con el típico bucle anidado.
29    for (int i=0; i<sol.length; i++)
```

```

30     for (int j=0; j<sol[i].length; j++) {
31         // En esta celdilla calcularemos una suma, así que inicializamos a cero.
32         sol[i][j] = 0;

34         // Hacemos tantas sumas como números de columnas tiene la primera.
35         // Para los índices (i, j), recorremos la fila i de la primera
36         // y la columna j de la segunda.
37         for (int k=0; k<matriz1[i].length; k++) {
38             sol[i][j] += matriz1[i][k] * matriz2[k][j];
39         }
40     }

42     return sol;
43 }

```

## 4. Matrices Binarias y Grafos

### 4.1. Relaciones binarias

When I use git push origin master, I get an error:

---

```

// Hello.java
import javax.swing.JApplet;
import java.awt.Graphics;

public class Hello extends JApplet {
    public void paintComponent(Graphics g) {
        g.drawString("Hello, world!", 65, 95);
    }
}

```

---

Que se lee: la relación binaria  $\mathcal{R}$  es el conjunto de pares ordenados  $(a, b)$  pertenecientes al producto cartesiano  $a \times B$ , y para los cuales se cumple la propiedad  $\mathcal{P}$  que los relaciona.

Las propiedades siguientes son correctas para representar la relación binaria  $\mathcal{R}$  entre los elementos  $a$  y  $b$ :

$$a\mathcal{R}b \quad \text{o} \quad \mathcal{R}(a, b) \quad \text{o bien} \quad (a, b) \in \mathcal{R}$$

Una relación binaria homogénea es la que se da entre los elementos de un único conjunto, llamando  $A$  al conjunto, tendríamos:  $R(a, b) : (a, b) \in A^2$ .

También podemos representar una relación binaria homogénea como una correspondencia de  $A$  sobre  $A : R : A \rightarrow A$

Tomando como conjunto inicial al conjunto  $A$  y como final también al conjunto  $A$ , nos permite asociar un elemento inicial a otro final dentro de un mismo conjunto, determinando una operación matemática o función de cálculo y no una estructura interna, teniendo siempre en cuenta, que si bien el conjunto inicial y final son un mismo conjunto, la relación es unidireccional, y si el elemento  $a$  está relacionado con el  $b$  no implica, necesariamente, que el  $b$  lo esté con el  $a$ .

### 4.2. Grafos

Las relaciones binarias homogéneas se pueden representar mediante grafos.

En matemáticas y ciencias de la computación, un grafo (del griego grafo: dibujo, imagen) es un conjunto de objetos llamados vértices o nodos unidos por enlaces llamados aristas o arcos, que permiten representar relaciones binarias entre elementos de un conjunto o un objeto de estudio de la teoría de grafos. Típicamente, un grafo se representa gráficamente como un conjunto de puntos (vértices o nodos) unidos por líneas (aristas).

Desde un punto de vista práctico, los grafos nos permiten estudiar las interrelaciones entre unidades que interactúan unas con otras. Por ejemplo, una red de computadoras puede representarse y estudiarse mediante



un grafo, en el cual los vértices representan terminales y las aristas representan conexiones (las cuales, a su vez, pueden ser cables o conexiones inalámbricas).

Un grafo  $G$  es un par ordenado  $G = (V, E)$ , donde :

- $V$  es un conjunto de vértices o nodos, y
- $E$  es un conjunto de aristas o arcos, que relacionan estos nodos.

Normalmente  $V$  suele ser finito. Muchos resultados importantes sobre grafos no son aplicables para grados infinitos.

- Se llama orden del grafo  $G$  a su número de vértices, .
- El grado de un vértice o nodo  $v \in V$  es igual al número de arcos que lo tienen como extremo.

### Tipos de Grafos

Los grafos pueden ser dirigidos, no dirigidos o mixtos.

- Un grafo dirigido o digrafo es un grafo  $G = (V, E)$  donde:
  - $V \neq \emptyset$
  - $E \subseteq \{(a, b) \in V \times V : a \neq b\}$  es un conjunto de pares ordenados de elementos de  $V$ .

Dada una arista  $(a, b)$ ,  $a$  es un nodo inicial y  $b$  su nodo final.

Un ejemplo es la Figura 2.

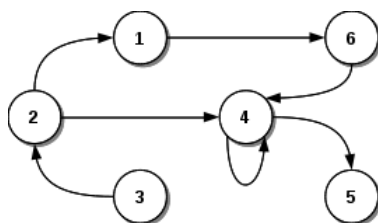


Figura 1: . Un ejemplo de grafo dirigido

- Un grafo no dirigido es un grafo  $G = (V, E)$  donde:
  - $V \neq \emptyset$
  - $E \subseteq \{x \in \mathcal{P}(V) : |x| = 2\}$  es un conjunto de pares ordenados de elementos de  $V$ .

Un par no ordenado es un conjunto de la forma  $(a, b)$ , de manera que  $(a, b) = (b, a)$ . Para los grafos, estos conjuntos pertenecen al conjunto potencia de  $V$ , denotado  $\mathcal{P}(V)$ , y son de cardinalidad 2.

Un ejemplo es la figura 2.

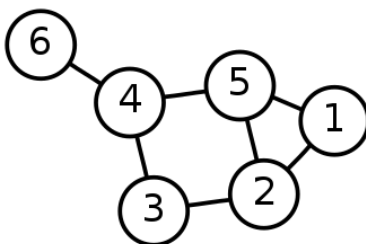


Figura 2: . Un ejemplo de grafo no dirigido

- Un grafo mixto es aquel que se define con la capacidad de poder contener aristas dirigidas y no dirigidas. Tano los grafos dirigidos como los no dirigidos son casos particulares de este.

### 4.3. Representación de grafos

Una matriz lógica, matriz binaria, matriz de relación, matriz booleana o matriz  $(0, 1)$ , es una matriz con entradas del dominio booleano  $B = 0, 1$ . Tal matriz puede ser usada para representar una relación binaria entre un par de conjuntos finitos y en particular a los grafos.

Las dos representaciones principales de grafos son las siguientes:

**Matriz de adyacencia (MA):** Se utiliza una matriz de tamaño  $n \times m$  donde las filas y las columnas hacen referencia a los vértices para almacenar en cada casilla la longitud entre cada par de vértices del grafo. La celda  $M[i, j]$  almacena la longitud entre el vértice  $i$  y el vértice  $j$ . Si su valor es infinito significa que no existe arista entre esos vértices, y  $MA[i, j] = 0$ .

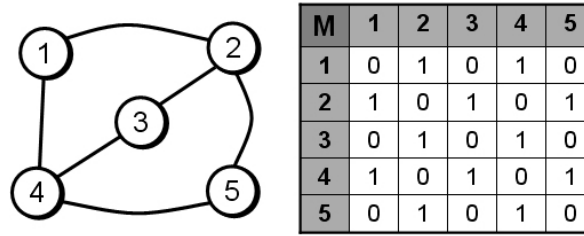


Figura 3: . Un ejemplo de matriz de adyacencia

En pseudocódigo basado en Java, para construir la matriz de adyacencia de la Figura 3 es:

Listado 4: Matriz de Adyacencia

```

1 boolean [][] grafo = new int[5][5]; // Grafo sobre 5 elementos.
3 for (int i=0; i < grafo.length; i++)
4   for (int j=0; j < grafo[i].length; j++) {
5     boolean hayUno = ((i0 || i== 4) && (j==1 || j==3)) ||
6       ((i=1 || i== 3) && (j==0 || j==1 || j==4));
8     if (hayUno) grafo[i][j]=1;
9     else grafo[i][j] = 0;
10  }
11 }

```

**Lista de adyacencia (LA):** Se utiliza un vector de tamaño  $n$  (un elemento por cada vértice) donde  $LA[i]$  almacena la referencia a una lista de los vértices adyacentes a  $i$ . En una red esta lista almacenará también la longitud de la arista que va desde  $i$  al vértice adyacente.

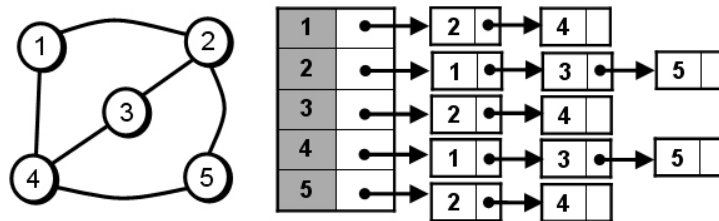


Figura 3: . Un ejemplo de matriz de adyacencia

Podemos simular las listas mediante arrays con filas de longitud variable. En pseudocódigo basado en Java para la Figura 4 es:

Listado 5: Lista de Adyacencia con Arrays

```

1 boolean [][] grafo = new int[5][]; // Grafo sobre 5 elementos.

```

```

3 for (int i=0; i < grafo.length; i++)
4   if (i==0 || i ==2 || i==3) grafo[i] = new int[2];
5   else grafo[i] = new int[3];

7   for(int j=0; j < grafo[i].length; j++) {
8     if (i mod 2 = 0) grafo[i][j] = 2*j + 2;
9     else grafo[i][j] = 2*j + 1;
10  }
11 }

```

## 5. Matriz Jacobiana

En cálculo vectorial, se llama jacobiano o determinante jacobiano al determinante de la matriz jacobiana. Tanto la matriz jacobiana como el determinante jacobiano reciben su nombre en honor al matemático Carl Gustav Jacobi.

La matriz jacobiana es una matriz formada por las derivadas parciales de primer orden de una función. Una de las aplicaciones más interesantes de esta matriz es la posibilidad de aproximar linealmente a la función en un punto. En este sentido, el jacobiano representa la derivada de una función multivariable. Propiamente deberíamos hablar más que de matriz jacobiana, de diferencial jacobiana o aplicación lineal jacobiana ya que la forma de la matriz dependerá de la base o coordenadas elegidas. Es decir, dadas dos bases diferentes la aplicación lineal jacobiana tendrá componentes diferentes aún tratándose del mismo objeto matemático. La propiedad básica de la "matriz" jacobiana es la siguiente, dada una aplicación cualquiera  $\mathbf{F}: \mathbb{R}^n \rightarrow \mathbb{R}^m$  continua, es decir  $\mathbf{F} \in \mathcal{C}^k(\mathbb{R}^n, \mathbb{R}^m)$  se dirá que es diferenciable si existe una aplicación lineal  $\lambda \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$  tal que:

$$\lim_{\|\mathbf{x}-\mathbf{y}\| \rightarrow 0} \frac{\|\mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{y}) - \lambda(\mathbf{x} - \mathbf{y})\|}{\|\mathbf{x} - \mathbf{y}\|} = 0 \quad (1)$$

Una de las aplicaciones más interesantes de esta matriz es la posibilidad de aproximar linealmente a la función en un punto. También se aplica en modelado cinemático de un robot. En dicho modelado se busca las relaciones entre las variables articulares y la posición (expresada normalmente en forma de coordenadas cartesianas) y orientación del extremo del robot. En esta relación no se tienen en cuenta las fuerzas o pares que actúan sobre el robot (actuadores, cargas, fricciones, etc.) y que pueden originar el movimiento del mismo. La matriz jacobiana directa permite conocer las velocidades del extremo del robot a partir de los valores de las velocidades de cada articulación. Por su parte, la matriz Jacobiana inversa permitirá conocer las velocidades determinadas en el extremo del robot.

### 5.1. Función escalar

Empecemos con el caso más sencillo de una función escalar  $F: \mathbb{R}^n \rightarrow \mathbb{R}$ . En este caso la matriz jacobiana será una matriz formada por un vector fila que coincide con el gradiente. Si la función admite derivadas parciales para cada variable puede verse que basta definir la "matriz" jacobiana como:

$$\lambda(\mathbf{x}) := \nabla F(x) = \frac{\partial F(x)}{\partial x_1} \dots \frac{\partial F(x)}{\partial x_n}$$

Ya que entonces se cumplirá la relación (2) automáticamente, por lo que en este caso la "matriz jacobiana" es precisamente el gradiente.

### 5.2. Función vectorial

Supongamos  $\mathbf{F}: \mathbb{R}^n \rightarrow \mathbb{R}^m$  es una función que va del espacio euclídeo  $n$ -dimensional a otro espacio  $m$ -dimensional. Esta función está determinada por  $m$  funciones escalares reales:

$$y_i = F_i(x_1, \dots, x_n), \quad \mathbf{y} = \mathbf{F}(\mathbf{x}) = (F_1(x), \dots, F_m(x))$$

Cuando la función anterior es diferenciable, entonces las derivadas parciales de estas  $m$  funciones pueden ser organizadas en una matriz  $m$  por  $n$ , la matriz jacobiana de  $F$ :

$$\begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}$$

Esta matriz es notada de diversas maneras:

$$\mathcal{J}_{\mathbf{F}}(x_1, \dots, x_n), \quad o \quad \frac{\partial(y_1, \dots, y_m)}{\partial(x_1, \dots, x_n)}, \quad o \quad D\mathbf{F}(x_1, \dots, x_n), \quad o \quad \nabla \mathbf{F}(x_1, \dots, x_n)$$

Nótese que la fila  $i$ -ésima fila coincidirá dada con el gradiente de la función  $y_i$ , para  $i = 1, \dots, m$ .

Si  $p$  es un punto de  $R^n$  y  $F$  es diferenciable en  $p$ , entonces su derivada está dada por  $JF(p)$ . En este caso, la aplicación lineal descrita por  $JF(p)$  es la mejor aproximación lineal de  $F$  cerca del punto  $p$ , de esta manera:

$$\mathbf{F}(\mathbf{x}) \approx \mathbf{F}(\mathbf{p}) + \mathbf{J}_{\mathbf{F}}(\mathbf{p})(\mathbf{x} - \mathbf{p})$$

para  $x$  cerca de  $p$ . O con mayor precisión:

$$\lim_{\|\mathbf{x} - \mathbf{p}\| \rightarrow 0} \frac{\|\mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{p}) - \mathbf{J}_{\mathbf{F}}(\mathbf{p})(\mathbf{x} - \mathbf{p})\|}{\|\mathbf{x} - \mathbf{p}\|} = 0$$

En ciertos espacios vectoriales de dimensión no finita, formados por funciones, puede generalizarse el concepto de matriz jacobiana definiendo una aplicación lineal jacobiana.

### 5.3. Ejemplos

**Ejemplo 1.** La matriz jacobiana de la función  $F: R^3 \rightarrow R^3$  definida como:

$$F(x_1, x_2, x_3) = (x_1, 5x_3, 4x_2^2 - 2x_3)$$

viene dada por

$$J_F(x_1, x_2, x_3) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 5 \\ 0 & 8x_2 & -2 \end{bmatrix}$$

**Ejemplo 2.** No siempre la matriz jacobiana es cuadrada, como en el siguiente ejemplo.

Supóngase la función  $F: R^3 \rightarrow R^4$ , cuyas componentes son:

$$y_1 = 1/x_1, y_2 = 5x_3y_3 = 4x_2^2 - 2x_3y_4 = x_3 \sin(x_1)$$

Aplicando la definición de matriz jacobiana:

$$J_F(x_1, x_2, x_3) = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \frac{\partial y_1}{\partial x_3} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \frac{\partial y_2}{\partial x_3} \\ \frac{\partial y_3}{\partial x_1} & \frac{\partial y_3}{\partial x_2} & \frac{\partial y_3}{\partial x_3} \\ \frac{\partial y_4}{\partial x_1} & \frac{\partial y_4}{\partial x_2} & \frac{\partial y_4}{\partial x_3} \end{bmatrix} = \begin{bmatrix} -1/x_1^2 & 0 & 0 \\ 0 & 0 & 5 \\ 0 & 8x_2 & -2 \\ x_3 \cos(x_1) & 0 & \sin(x_1) \end{bmatrix}$$

## 6. Maxima