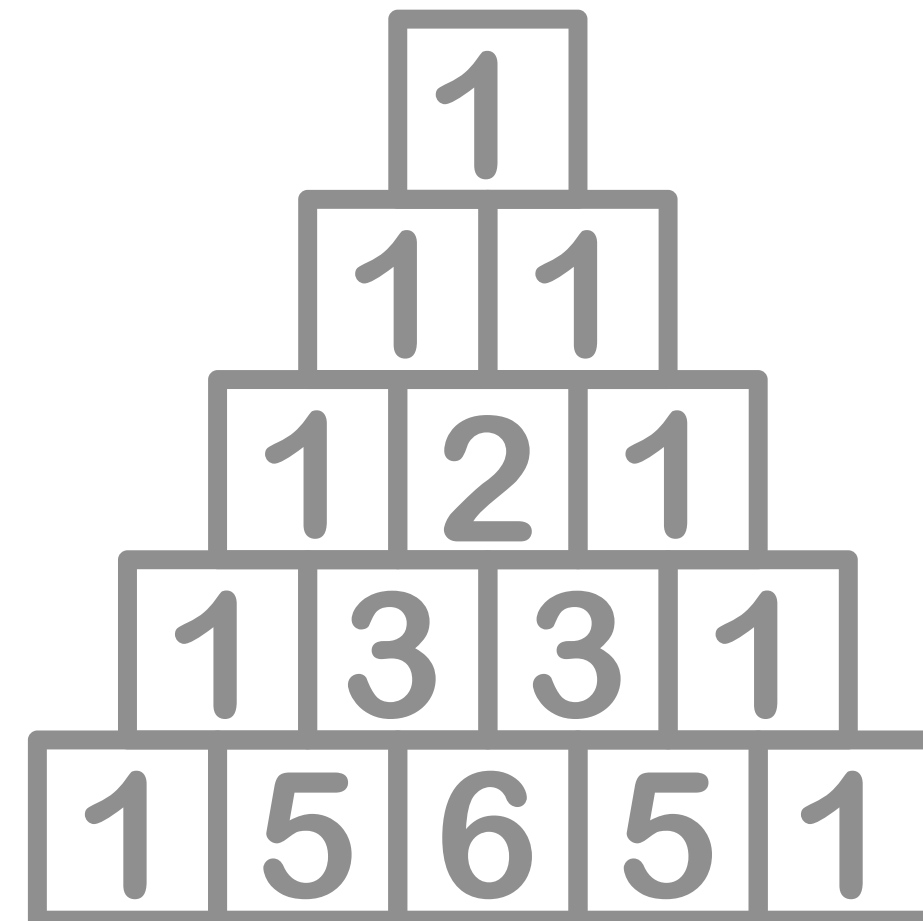


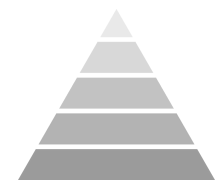
× J Reto #2

Triángulo de Pascal

Equipo 6

Contreras Martínez Alan Gael
Gómez Rosales Roberto Josué.
Guzmán Fernández Andrés Rogelio
Martínez Ruiz Abdiel Barush
Santos Mateos Oswaldo



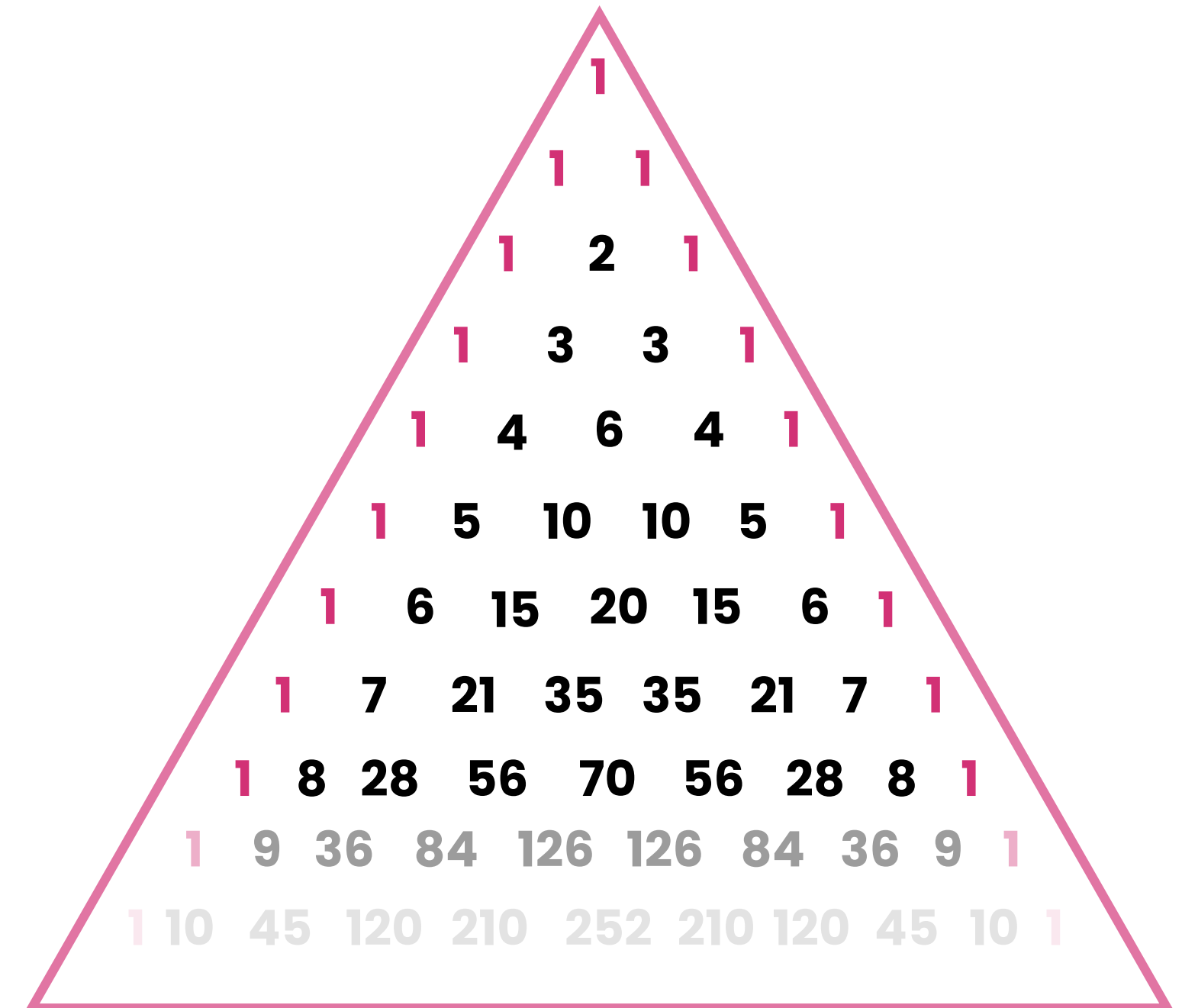


Triángulo de Pascal

El **Triángulo de Pascal** es una representación triangular de números enteros donde **cada número es la suma de los dos que están directamente arriba de él.**

Los bordes del triángulo están formados por unos.

Aunque lleva el nombre del matemático francés **Blaise Pascal**, quien realizó un estudio profundizado del mismo en el siglo XVII, *su conocimiento era conocido por culturas mucho más antiguas.*





Aplicaciones

1-. Coeficientes Binomiales (Combinatoria):

Cada número en la fila "n" y posición "k" (ambas iniciando en 0) corresponde al coeficiente binomial " $C(n, k)$ ". Representa el número de formas de elegir `k` elementos de un conjunto de "n" elementos. Ejemplo: En la fila 4 (1, 4, 6, 4, 1), el valor "6" es " $C(4, 2) = 6$ " (**combinaciones de 2 elementos en un conjunto de 4**).

2-. Expansión Binomial (Teorema del Binomio):

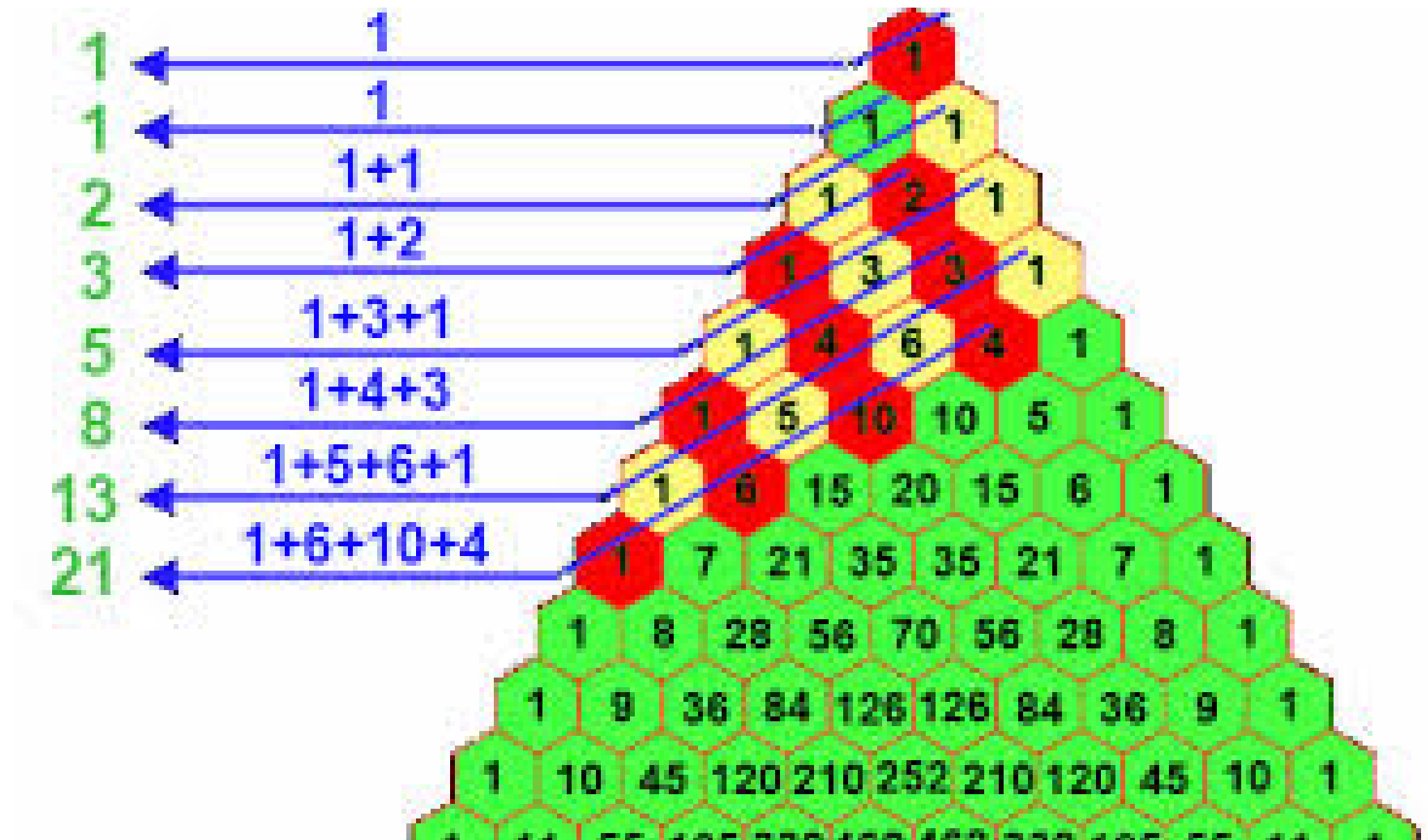
Los coeficientes de la expansión de " $(a + b)^n$ " son idénticos a los valores de la fila "n" del triángulo.

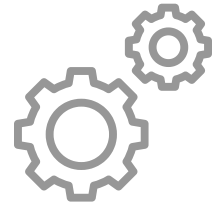
$$(a + b)^3 = 1a^3 + 3a^2b + 3ab^2 + 1b^3$$

Coeficientes: 1, 3, 3, 1 (fila 3).

3-. Serie de Fibonacci:

Sumando los números en diagonales específicas del triángulo se obtienen los términos de la sucesión de Fibonacci (1, 1, 2, 3, 5, 8...).





Triangulo de Pascal (Recursivo)

Pascal(i, k)

Caso Base

Si "k" vale 0 o "k" es igual a "i",
entonces se devuelve 1

Caso Recursivo

$\text{Pascal}(i, k) = \text{Pascal}(i-1, k-1) + \text{Pascal}(i-1, k)$

```
import java.util.Scanner;

public class PascalRecursivo{
    public static void main(String[] args) {
        Scanner e= new Scanner(System.in);
        System.out.println("Hola usuario, este codigo crea el triangulo de Pascal a partir de una entrada");
        System.out.println("Ingresa el numero de filas con el que quieres que se creé el triangulo");
        int filas = e.nextInt();
        System.out.println("");
        System.out.println("El triangulo con " + filas + " filas es: ");
        filas +=1;
        System.out.println("");
    }
}
```

Lo que se imprime

For's

```
for (int i=0; i<filas;i++){ // Crea el triangulo y lo va imprimiendo
    for (int j=0; j<filas-i;j++){ //Imprime los espacios al inicio de cada fila para darle forma al triangulo
        System.out.print(" ");
    }

    for (int k=0; k<=i; k++){ // Imprime los digitos del triangulo con un espacio para separar los terminos
        System.out.print(Pascal(i,k) + " ");
    }

    System.out.println(""); //Salto de linea para la siguiente fila del triangulo
}

}
```

¿Qué es?

Aplicaciones

Código Recursivo

Código Matricial

Método

```
public static int Pascal(int i, int k){ //Función recursiva
    if (k==0 || k==i){ //Caso base
        return 1;
    } else {
        return Pascal(i-1,k-1) + Pascal(i-1, k); //Obtención del dígito a partir de la suma de los dos de la fila anterior
    }
}
}
```



Pruebas de escritorio

i - Iteración del ciclo externo

j - Número de espacios antes de que se imprima el primer dígito

k - Número de valores que se van a imprimir

i=0

j=0,1,2,3,4

k=0

uuuuuu1u

i=1

j=0,1,2,3

k=0,1

uuuuuu1u

uuuu1u1u

i=2

j=0,1,2

k=0,1,2

uuuuuu1u

uuuu1u1u

uuu1u2u1u

i=3

j=0,1

k=0,1,2,3

uuuuuu1u

uuuu1u1u

uuu1u2u1u

uu1u3u3u1u

i=4

j=0

k=0,1,2,3,4

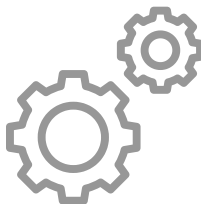
uuuuuu1u

uuuu1u1u

uuu1u2u1u

uu1u3u3u1u

u1u4u6u4u1u



Triangulo de Pascal (Matricial)

filas +1

uuuuu1uuuuu
uuuu1u1uuuu
uuu1u2u1uuu
uu1u3u3u1uu
u1u4u6u4u1u

$(filas + 1) * 2 + 1$

0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	1	0	1	0	0	0	0
0	0	0	1	0	2	0	1	0	0	0
0	0	1	0	3	0	3	0	1	0	0
0	1	0	4	0	6	0	4	0	1	0

```
import java.util.Scanner;

public class PascalMatricial{
    public static void main(String[] args) {
        Scanner e = new Scanner(System.in);
        System.out.println("Hola usuario, este código crea el triangulo de Pascal");
        System.out.println("Ingresa el numero de filas:");
        int filas = e.nextInt();
        System.out.println("");
        System.out.println("El triangulo con " + filas + " filas es: ");
        filas +=1;
        System.out.println("");

        int[][] arreglo = new int[filas][(filas * 2) + 1];
```

Construcción de la matriz

Lo que se imprime

```
// Inicializar todo con ceros
for (int i = 0; i < filas; i++) {
    for (int j = 0; j < (filas * 2 + 1); j++) {
        arreglo[i][j] = 0;
    }
}

arreglo[0][filas] = 1; // punta del triangulo

for (int i = 1; i < filas; i++) {
    // Lógica para calcular los valores
    for (int k = 1; k < (filas * 2); k++) {
        if (arreglo[i-1][k-1] != 0 || arreglo[i-1][k+1] != 0) {
            arreglo[i][k] = arreglo[i-1][k-1] + arreglo[i-1][k+1];
        }
    }
}
```

Impresión de la matriz

```
// Imprimir resultado
for (int m = 0; m < filas; m++) {
    for (int n = 0; n < (filas * 2 + 1); n++) {
        if (arreglo[m][n] == 0) {
            System.out.print(" ");
        } else {
            System.out.print(arreglo[m][n]);
        }
    }
    System.out.println(""); // Siguiendo fila en el triangulo
}
}
```