Reto por token - Equipo 5

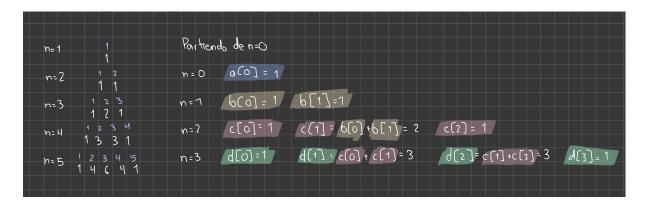
El Triángulo de Pascal es una matriz infinita de números en forma de triángulo, donde cada número es la suma de los dos que tiene encima, y los bordes son siempre 1. Se usa principalmente en álgebra para encontrar los coeficientes del desarrollo de un binomio elevado a cualquier potencia, conocido como el binomio de Newton, y también tiene aplicaciones en combinatoria y probabilidad.

El reto era plasmar en un código de java, su construcción.

Construcción del código:

Análisis del problema:

Se sabe que los bordes del triángulo siempre serán 1, por lo que podemos decir que los casos base son 0 y n, ya en el caso promedio, el elemento tendrá el valor de el arreglo anterior en su posición más el que le precede. Desarrollando lo se puede apreciar de esta forma.



Pseudocódigo:

Se decidió usar dos arreglos, uno en el cual se almacenarán los datos que iremos imprimiendo, y una para guardar los datos que se irán utilizando iteración a iteración.

```
for (i=0, i<n,i+t)

for (j=0, j=i, j+t)

if (j=0 || j=i)

poscol(j)=1

else {

poscol(j)= aux[j-1] + aux[j]

for (j=0, j \in j+t) } Impresion (i) a por (i) a

print pascal(j)

for (K=0, K=1, K+t) } Asignación de volves

aux[k]=pascal[k] } del auxiliar
```

Código en Java

```
import java.util.Scanner;
     public class TrianguloPascal {
       public static void main(String[] args) {
          Scanner e=new Scanner(System.in);
          System.out.print("Ingresa el numero de filas del triangulo de pascal que quieres ver: ");
          int n= e.nextInt();
          int[] pascal = new int[n];
          int[] aux = new int[n];
          for (int i = 0; i < n; i++) {
            //Asignacion valores
            for (int j = 0; j <= i; j++) {
    if (j == 0 || j == i) { //Casos base 0 y n
        pascal[j] = 1;
13
14
              } else { //Caso promedio
                 pascal[j] = aux[j - 1] + aux[j];
            }
20
            for (int j = 0; j \le i; j++) {
               System.out.print(pascal[j] + " ");
23
24
            System.out.println();
25
26
            //Copia de pascal en arreglo auxiliar
27
            for (int k = 0; k \le i; k++) {
28
              aux[k] = pascal[k];
29
         }
31
32
```

Salida