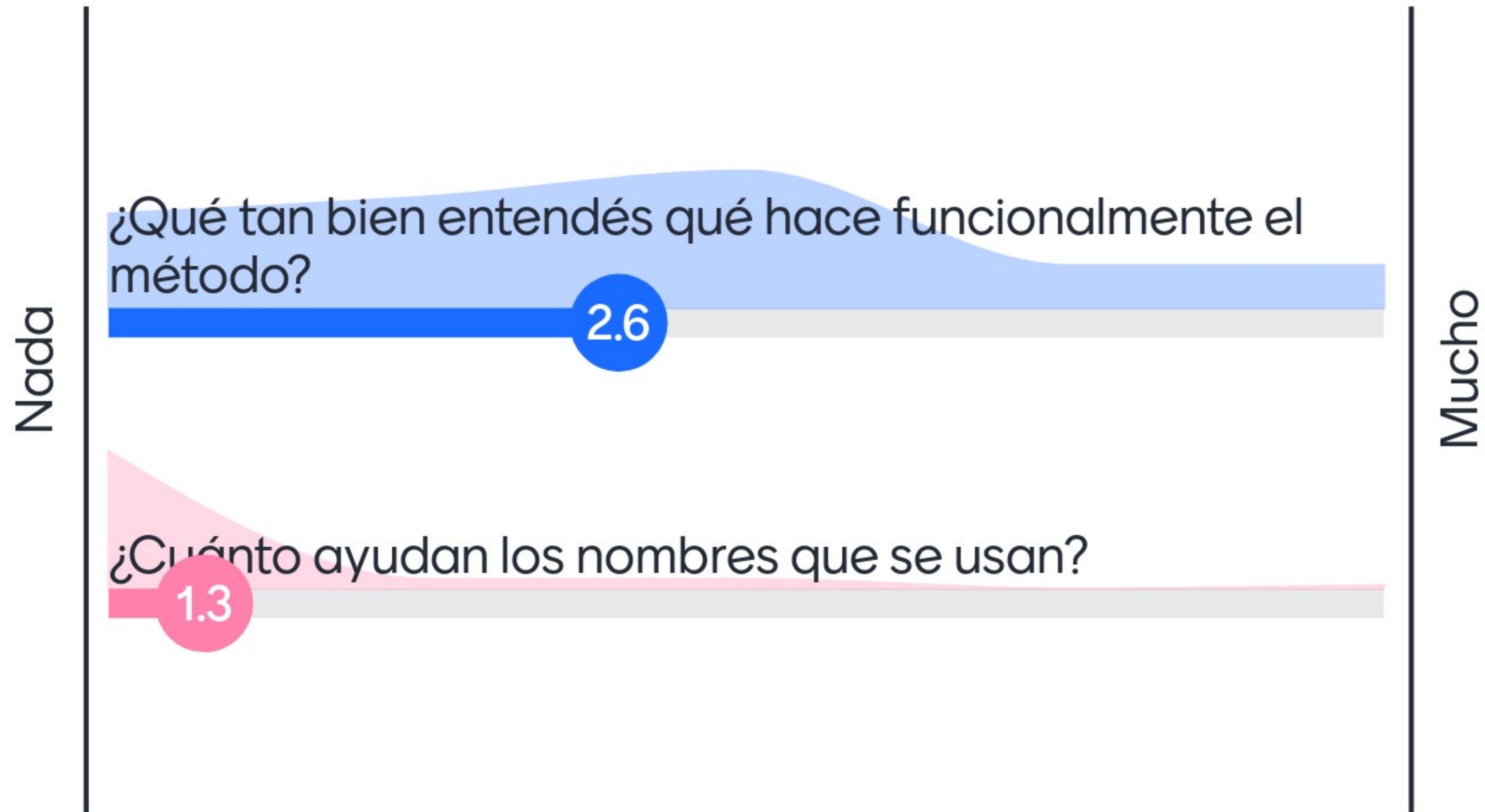


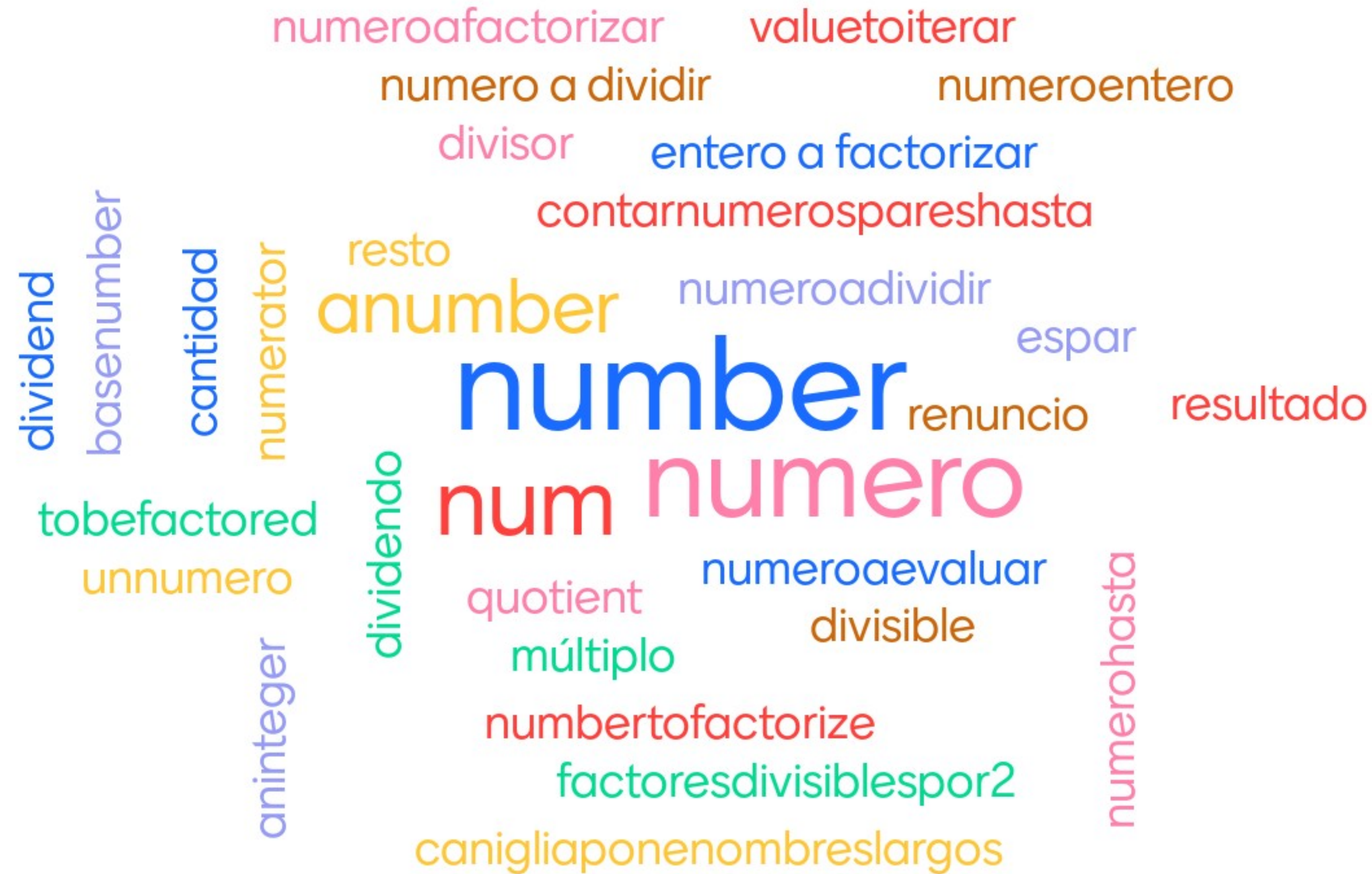
```
public List<Integer> m(int n) {  
    List<Integer> r = new ArrayList<>(); // Crea una lista vacia  
    int i = 2;  
  
    while (n > 1) {  
        while (n % i == 0) { // % es módulo. Si el modulo de n por i es cero  
            r.add(i);        // /= divide n por i y lo asigna a n  
            n /= i;  
        }  
        i++; // incrementa i en uno  
    }  
  
    return r;  
}
```



Entender



¿Cómo llamarías a n?



¿Cómo llamarías a r?



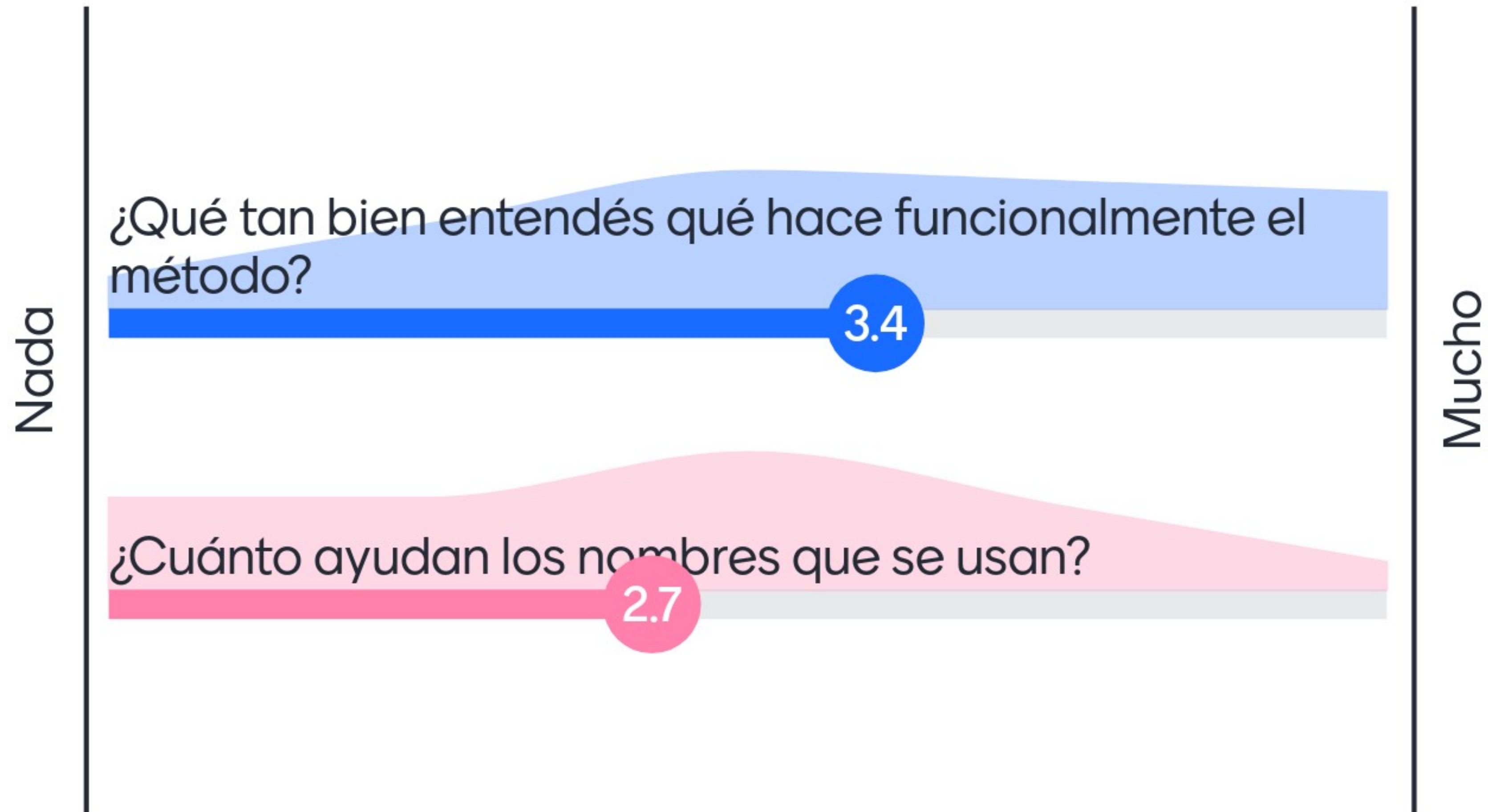
¿Cómo llamarías a i?



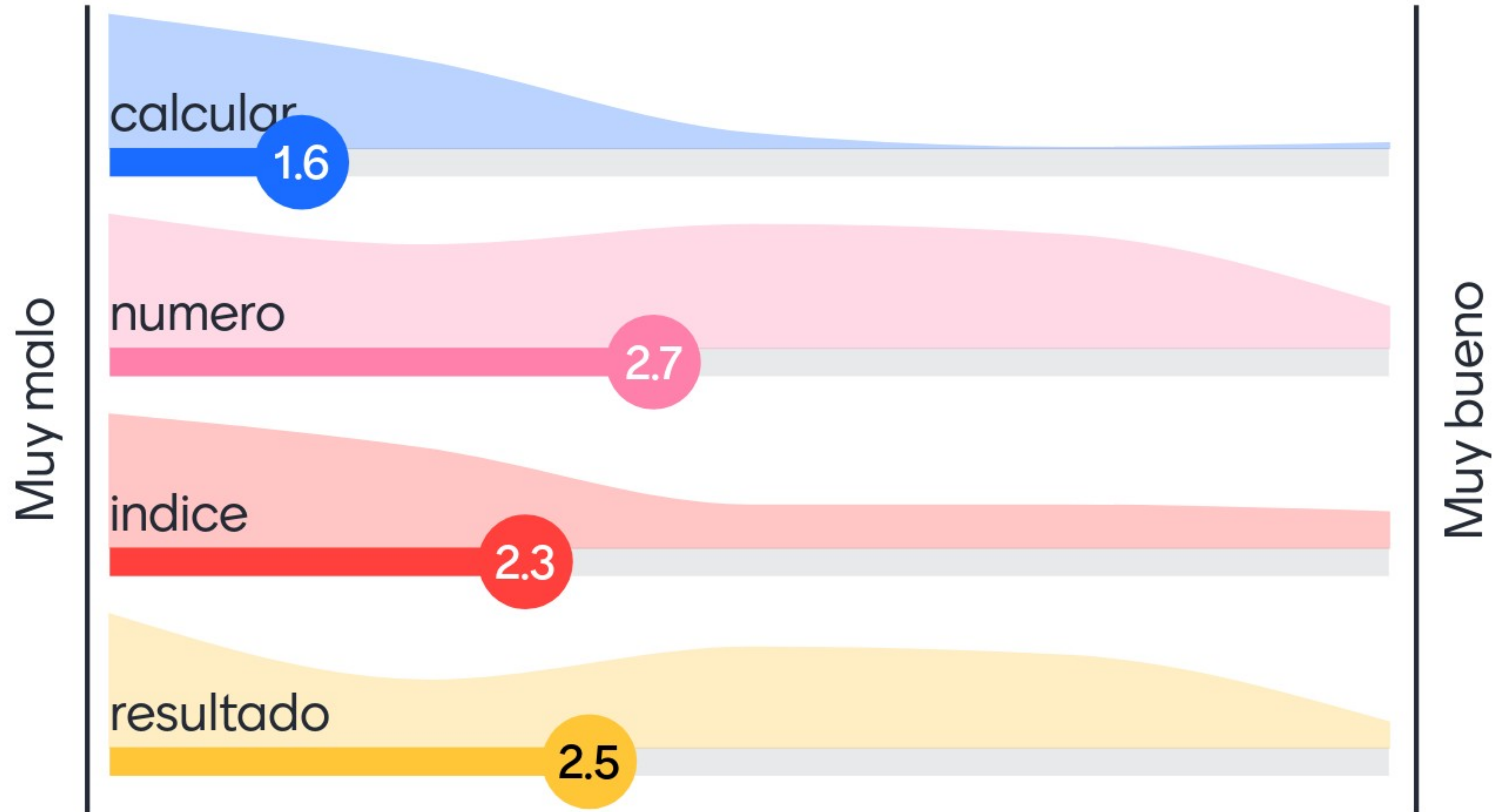
```
public List<Integer> calcular(int numero) {  
    List<Integer> resultado = new ArrayList<>();  
    int indice = 2;  
  
    while (numero > 1) {  
        while (numero % indice == 0) {  
            resultado.add(indice);  
            numero /= indice;  
        }  
        indice++;  
    }  
  
    return resultado;  
}
```



Entender



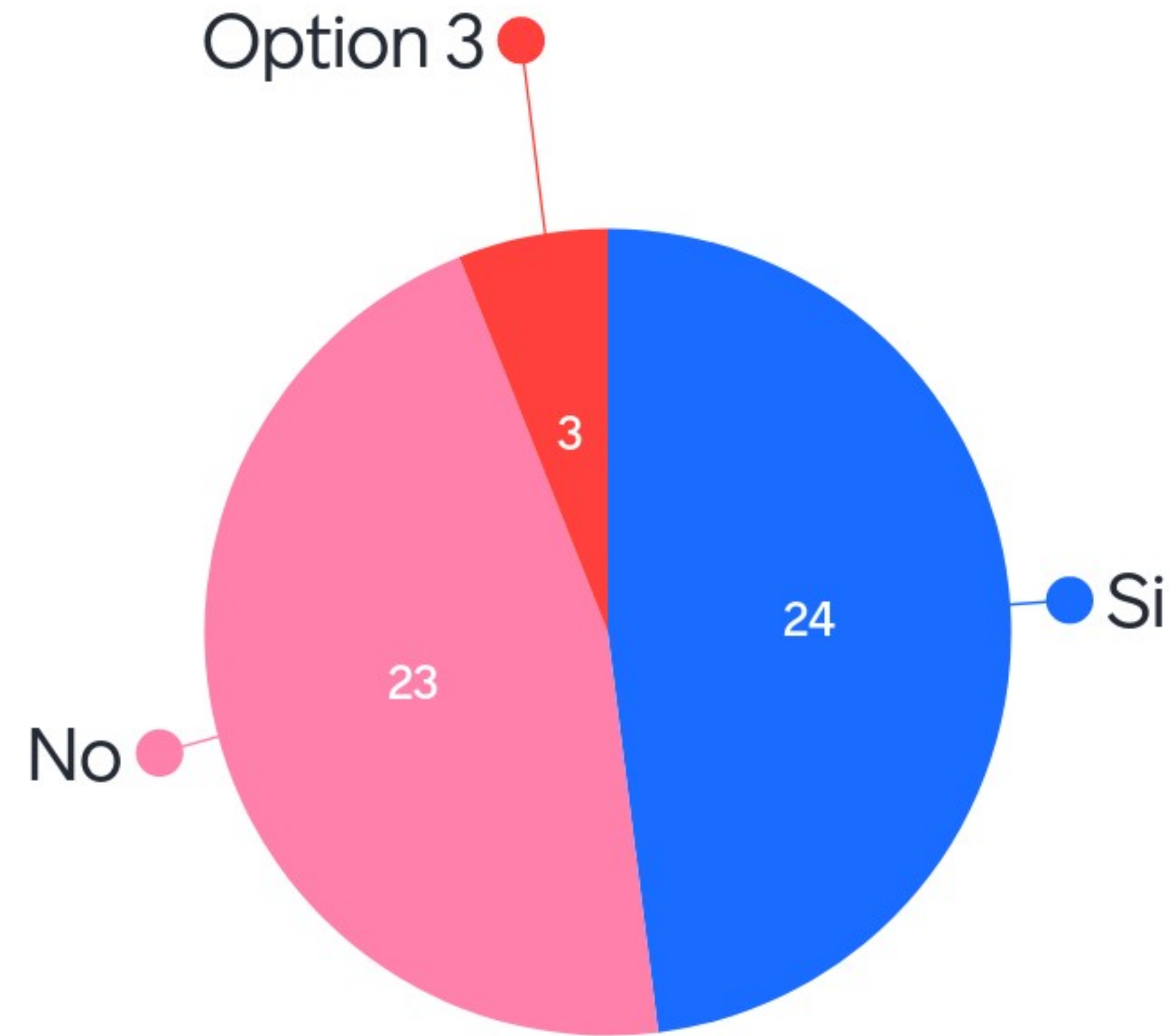
¿Qué tan bueno te parecen estos nombres?



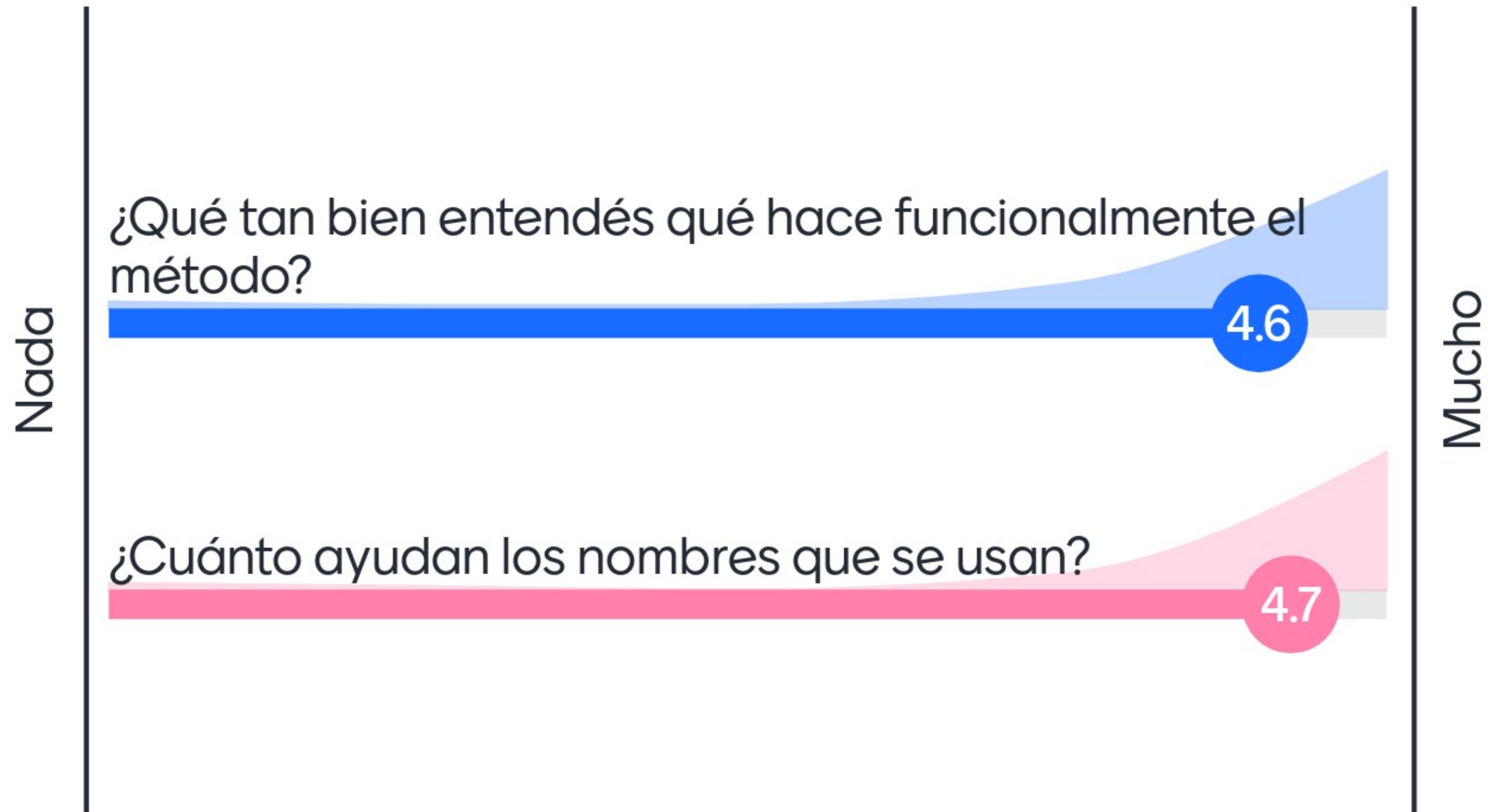

```
public List<Integer> factoresPrimosDe(int numeroAFactorizar) {  
    List<Integer> factoresPrimos = new ArrayList<>();  
    int divisor = 2;  
  
    while (numeroAFactorizar > 1) {  
        while (numeroAFactorizar % divisor == 0) {  
            factoresPrimos.add(divisor);  
            numeroAFactorizar /= divisor;  
        }  
        divisor++;  
    }  
  
    return factoresPrimos;  
}
```



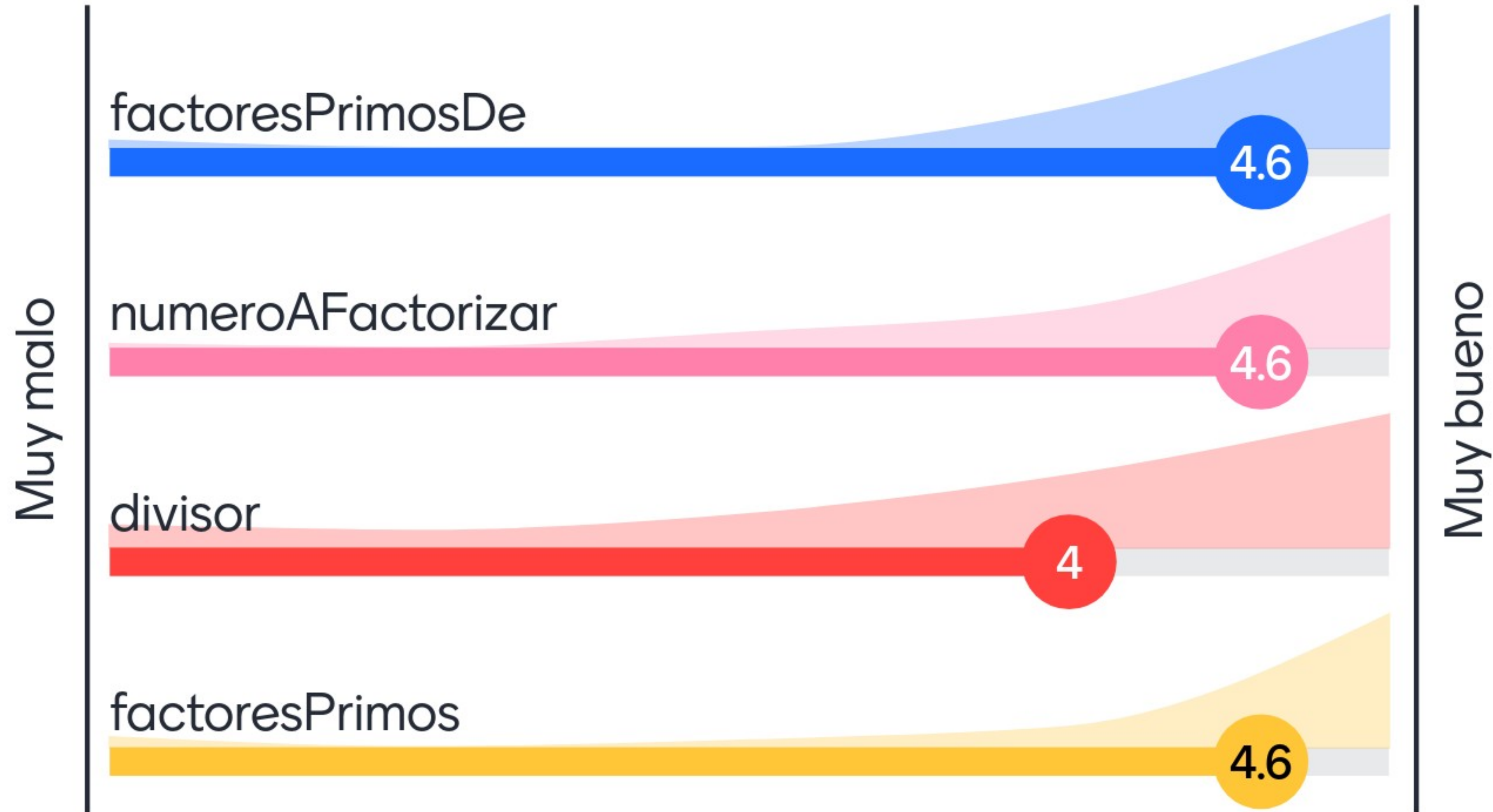
Antes de ver esta versión ¿entendiste que se estaban calculando los factores primos de un número?



Entender



¿Qué tan bueno te parecen estos nombres?




```
class FactoresPrimos {  
    private int numeroAFactorizar;  
    private List<Integer> factoresPrimos;  
    private int divisor;  
  
    public FactoresPrimos(int numeroAFactorizar) { this.numeroAFactorizar = numeroAFactorizar; }  
  
    public List<Integer> factorizar() {  
        inicializar();  
  
        while (sePuedeFactorizar()) {  
            factorizarPorDivisor();  
            calcularProximoDivisor();  
        }  
  
        return factoresPrimos;  
    }  
}
```



Entender

