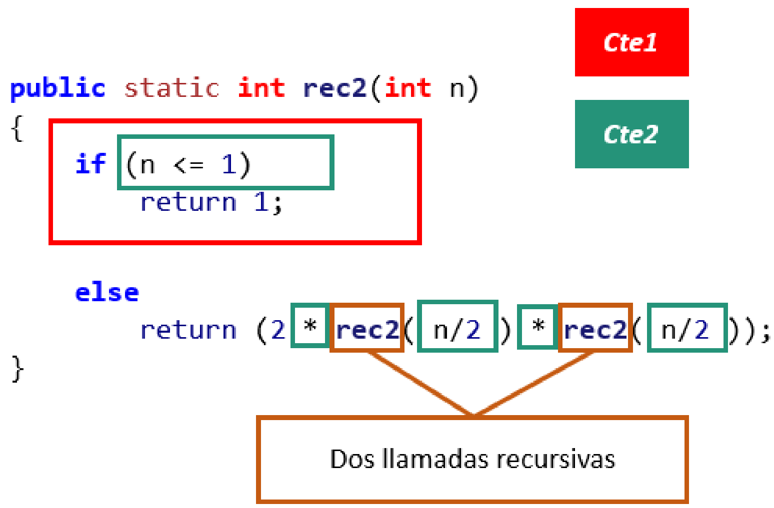
Calcular el T(n) y el big–Oh de:

```java
public static int rec2(int n)
{
    if(n <= 1)
        return 1;
    else
        return (2*rec2(n/2)*rec2(n/2));
}
```

Analizando el algoritmo se llega a lo siguiente:



Por lo tanto, la función de recurrencia es:

$$T(n) = \begin{cases} cte_1\ ; n \leq 1 \\ cte_2 + 2\,T(n/2)\ ; n \geq 2 \end{cases}$$

Para encontrar el patrón de recurrencia se procede de la siguiente manera:

$$T(n) = cte_2 + 2\,T(n/2) \rightarrow Primer\ caso\ recursivo \qquad \boldsymbol{i = 1}$$

$$\boldsymbol{T(n/2) = cte_2 + 2\,T\left(n/2^2\right)\ \ o\ \ T(n/2) = cte_2 + 2\,T(n/4)}$$

$$T(n) = cte_2 + 2[\,cte_2 + 2\,T(n/4)] \Rightarrow T(n) = cte_2 + 2cte_2 + 4\,T(n/4) \Rightarrow$$

$$T(n) = 4\,T(n/4) + 3cte_2\ ;\ \ n \geq 4 \qquad \boldsymbol{i = 2}$$

$$\boldsymbol{T(n/4) = cte_2 + 2\,T\left(n/2^3\right)\ \ o\ \ T(n/2) = cte_2 + 2\,T(n/8)}$$

$$T(n) = 4[cte_2 + 2\,T(n/8)] + 3cte_2 \Rightarrow T(n) = 8\,T(n/8) + 7cte_2\ \ ;\ \ n \geq 8 \qquad \boldsymbol{i = 3}$$

Por lo tanto, el patrón es:

$$\boldsymbol{T(n) = 2^k\, T\left({}^{n}/_{2^k}\right) + (2^k - 1)cte_2 \quad ; \quad n > k\, ; \;\; k = i}$$

Ahora hay que determinar cuántas llamadas recursivas “k” llevan hasta el caso base.

$\frac{n}{2^i} = 1 \Rightarrow n = 2^i \Rightarrow i = \log_2(n)$

Por ultimo hallar el T(n) explícito:

$T(n) = n\, T(1) + (n-1)cte_2 \;\; ; \;\; n \geq 1$

$$\boldsymbol{T(n) = n\, cte_1 + n\, cte_2 - cte_2 \quad ; \quad n \geq 1}$$

Ahora hay que estimar el orden en base al termino con el orden más grande:

- 1° termino: $n\, cte_1$ → es O(n)
- 2° termino: $n\, cte_2$ → es O(n)
- 3° termino: cte_2 → es O(1)

Para decir que $T(n) = n\, cte_1 + n\, cte_2 - cte_2$ es O(n) se tiene que poder demostrar que:

- Existen constantes c_0 y n_0 tales que:
  - $T(n) \leq c_0\, n\; para\; todo\; n \geq n_0$

En este caso ya se sabe que la $f(n) = n$, entonces:

- $cte_1 n \leq cte_1 n \;\; \forall\, n \geq 1$
- $cte_2 n \leq cte_2 n \;\; \forall\, n \geq 1$
- $cte_2 \leq cte_2 n \;\; \forall\, n \geq 1$

Entonces:

$$\begin{array}{rl} & cte_1 n \leq cte_1\, n \;\; \forall\, n \geq 1 \\ & cte_2 n \leq cte_2\, n \;\; \forall\, n \geq 1 \\ + & cte_2 \leq cte_2\, n \;\; \forall\, n \geq 1 \\ \hline & T(n) \leq (cte_1 + cte_2 + cte_2)n \;\; \forall\, n \geq 1 \end{array}$$

Como $\boldsymbol{T(n) \leq c_0\, n\, \forall\, n \geq n_0 \;\therefore\; T(n)\; es\; O(n)}$