🕐 Cuando esperas a que cargue una aplicación web o cuando una pestaña de un navegador ocupa mucho espacio, estamos consumiendo recursos como tiempo o espacio.

📈 Los algoritmos que se ejecutan al realizar acciones pueden ser medibles en la notación Big-O.

👷 Para calcular la notación Big-O aplicamos una serie de reglas a través de nuestro código.

📏 Reglas para estructuras

Las reglas para Big-O en tiempo son:

Reglas de Big-O en tiempo

`let a = true`	O(1)

`for() {`
 	
`}`
	O(n)

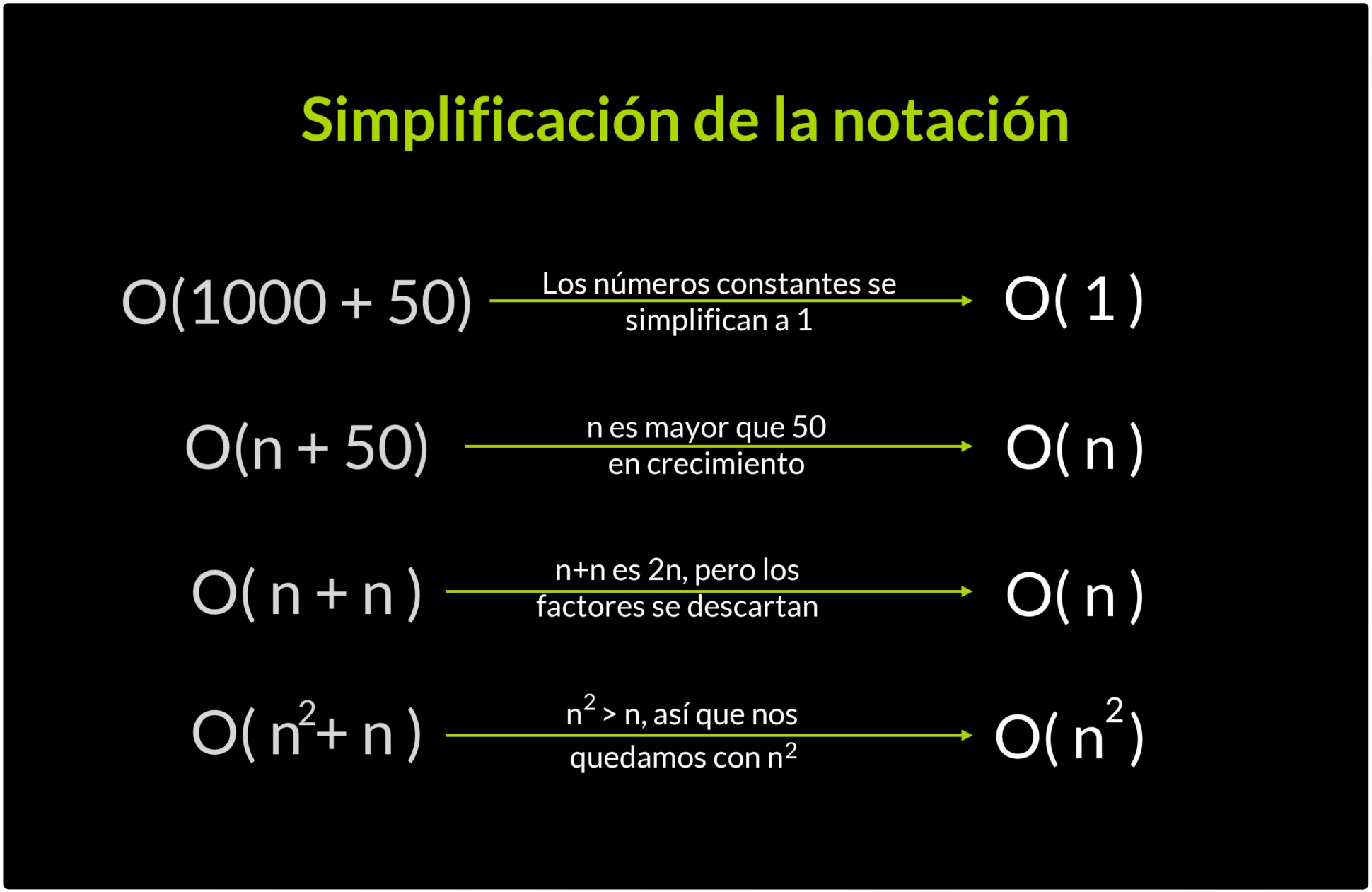
`for() {`
`  for() {}`
`}`
	$O(n^2)$

Las reglas para Big-O en espacio son:

Reglas de Big-O en espacio

`let a = true`	O(1)
`let b = [1, 2, 3, 4, ...]`	O(n)
`let c = [[1, 2], [3, 4]]`	$O(n^2)$

Y siempre debemos simplificar la notación:



Hasta aquí tenemos todas las conclusiones del curso. Pero recuerda que también ahondamos en el por qué de cada concepto.

¿Por qué necesitamos Big-O? ¿Por qué una notación?

Un algoritmo o un programa podrían ejecutarse en cinco o diez horas, incluso si hablamos de una o varias computadoras. Big-O viene a poner orden todo eso, dándonos una forma fácil de leer en la que podemos determinar la eficiencia de un algoritmo.

¿Por qué asignamos Big-O a cada estructura?

Big-O medirá el recurso generado respecto a la entrada del algoritmo. Y las estructuras son un aspecto sencillo de convertir en medición de recursos.

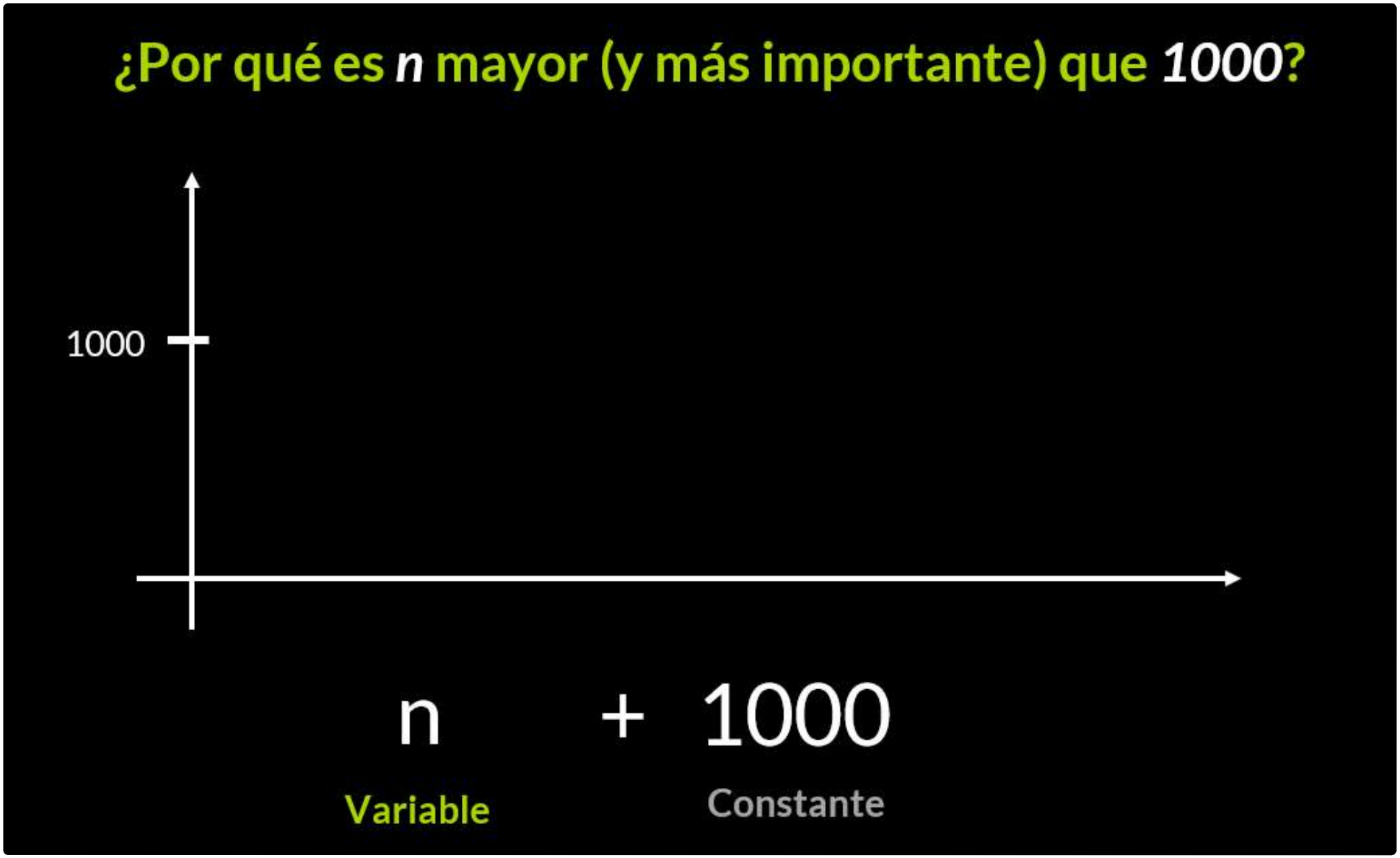
Por ejemplo:

- Con los bucles es sencillo, los bucles repiten instrucciones, y al repetir toman más tiempo en la computadora.
- Con los arreglos es igual, repiten una serie de elementos, y al repetir esos elementos toman más espacio de la computadora.

¿Por qué nos quedamos con el grado mayor al simplificar Big-O?

En Big-O queremos comprender qué tanto recurso (como tiempo o espacio) nos gasta un algoritmo cuándo aumentamos los datos. Y cada grado aumenta a un ritmo totalmente distinto.

Por ejemplo n crece más que 1000:



No es necesario quedarnos con los grados pequeños: Podemos simplificar y quedarnos con lo importante.

¿Solo hay Big-O para espacio y tiempo?

No, la complejidad es el estudio de los recursos que utilizan los algoritmos. Estos recursos pueden ser cualquier concepto de hardware y software. Como acceso a la memoria, comparaciones de condiciones, o lo que se necesite *limitar*.

La buena noticia de esto, es que no se requiere inventar nuevas notaciones. Big-O es suficiente para que personas como tú, desarrolladores/as de software o científicos/as de computación trabajen sobre esto.