



En la ciencia de la computación, currificar es la técnica inventada por Moses Schönfinkel y Gottlob Frege que consiste en transformar una función que utiliza múltiples argumentos (o más específicamente una n-tupla como argumento) en una función que utiliza un único argumento.

El nombre «currificar», acuñado por Christopher Strachey en 1967, es una referencia al lógico Haskell Curry. Un nombre alternativo, *Schönfinkelisation*, ha sido propuesto.

Dada una función  $f$  del tipo  $\alpha \rightarrow \beta$ , currificándola sería una función del tipo  $\alpha \rightarrow (\beta \rightarrow \gamma)$ . En otras palabras,  $f$  toma un argumento del tipo  $\alpha$  y retorna una función del tipo  $\beta \rightarrow \gamma$ . Descurrificar es la transformación inversa.

Intuitivamente, la currificación expone que «Si ajustas algunos argumentos, tendrás una función de los argumentos restantes». Por ejemplo, si la función *div* significa la versión currificada de la operación  $x / y$ , entonces *div* con el parámetro  $x$  ajustado en 1 es otra función: igual que la función *inv* que devuelve la inversa multiplicativa de sus argumentos, definida por  $inv(y) = 1 / y$ .

La motivación práctica para currificar es que en ocasiones, muy seguidas, las funciones obtenidas al utilizar algunos, pero no todos, los argumentos en una función currificada pueden resultar útiles; por ejemplo, muchos lenguajes tienen una función o un operador similar a `plus_one`. Currificar hace fácil definir dichas funciones.