**Ejemplos de funciones con sus parámetros y argumentos:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Función** | **Parámetros** | **Descripción** | **Por qué es un buen nombre** |
| calcularImpuestoTotal(precioBase, tasaImpuesto) | precioBase: number, tasaImpuesto: number | Calcula el impuesto total sobre un precio base dado una tasa impositiva. | Claro, conciso y describe la acción exacta. Utiliza términos comunes como "precio" e "impuesto". |
| validarFormatoEmail(email) | email: string | Verifica si una cadena de texto tiene el formato de un correo electrónico válido. | Describe la acción y el tipo de dato. Utiliza un verbo fuerte ("validar") y un término específico ("formato de email"). |
| guardarUsuario(usuario) | usuario: Usuario | Guarda un objeto de usuario en la base de datos. | Describe la acción y el objeto involucrado. Utiliza un término genérico ("usuario") que es fácilmente comprensible. |
| enviarNotificacion(usuario, mensaje) | usuario: Usuario, mensaje: string | Envía una notificación a un usuario específico con un mensaje determinado. | Describe la acción y los destinatarios del mensaje. Utiliza términos comunes como "usuario" y "mensaje". |
| obtenerProductosEnOferta(categoria) | categoria: string | Obtiene una lista de productos en oferta de una categoría específica. | Describe la acción y el criterio de búsqueda. Utiliza términos comunes como "producto", "oferta" y "categoría". |
| convertirMoneda(cantidad, monedaOrigen, monedaDestino) | cantidad: number, monedaOrigen: string, monedaDestino: string | Convierte una cantidad de una moneda a otra. | Describe la acción y las unidades involucradas. Utiliza términos específicos como "moneda origen" y "moneda destino". |
| generarReporteVentas(fechaInicio, fechaFin) | fechaInicio: Date, fechaFin: Date | Genera un reporte de ventas en un rango de fechas especificado. | Describe la acción y los parámetros de entrada. Utiliza términos comunes como "reporte" y "ventas". |

Export to Sheets

**Consideraciones adicionales:**

* **Verbos fuertes:** Utiliza verbos que describan claramente la acción que realiza la función (calcular, validar, guardar, enviar, obtener, convertir, generar, etc.).
* **Nombres de parámetros descriptivos:** Los nombres de los parámetros deben indicar el tipo de dato que esperan y su propósito en la función.
* **Evita abreviaturas:** A menos que sean muy conocidas y utilizadas en el contexto, evita abreviaturas para mejorar la legibilidad.
* **Sé consistente:** Utiliza un estilo de nombramiento consistente en todo tu código.
* **Piensa en el lector:** Imagina que alguien que no está familiarizado con tu código está leyendo tu función. ¿El nombre y los parámetros son lo suficientemente claros para entender su propósito?

**Ejemplo con comentarios explicativos:**

JavaScript

function calcularPrecioTotalConDescuento(precioOriginal, descuentoPorcentaje) { // Calcula el precio total de un producto después de aplicar un descuento. // // Parámetros: // - precioOriginal: El precio original del producto. // - descuentoPorcentaje: El porcentaje de descuento a aplicar. // // Retorna: // - El precio total después del descuento. const descuento = precioOriginal \* (descuentoPorcentaje / 100); const precioFinal = precioOriginal - descuento; return precioFinal; }

Use code [with caution.](file:///C:/faq#coding)

**Beneficios de utilizar nombres claros para funciones:**

* **Mayor legibilidad:** Facilita la comprensión del código.
* **Menor curva de aprendizaje:** Nuevos desarrolladores pueden entender el código más rápidamente.
* **Menos errores:** Al ser claros los nombres, se reducen las posibilidades de errores al llamar a las funciones.
* **Mantenimiento más fácil:** Al ser más fácil de entender, el código es más fácil de modificar y ampliar.

Al seguir estos consejos, podrás crear funciones con nombres claros y descriptivos que faciliten la comprensión y el mantenimiento de tu código a largo plazo.