# C Obtener ayuda



Todo el mundo se atasca en algún momento cuando está aprendiendo a programar. Por eso, una de las habilidades más importantes que debes aprender como programador es cómo desatascarte eficazmente. Este apéndice describe varias formas de ayudarte a volver a ponerte en marcha cuando la programación se vuelve confusa.

## Primeros pasos

Cuando estés atascado, tu primer paso debe ser evaluar tu situación. Antes de pedir ayuda a nadie, responde con claridad a las tres preguntas siguientes:

* ¿Qué intentas hacer?
* ¿Qué has intentado hasta ahora?
* ¿Qué resultados has obtenido?

Haz que tus respuestas sean lo más concretas posible. Para la primera pregunta, afirmaciones explícitas como "Estoy intentando instalar la última versión de Python en mi nuevo portátil con Windows" son lo suficientemente detalladas como para que otros miembros de la comunidad Python puedan ayudarte. Afirmaciones como "Estoy intentando instalar Python" no proporcionan suficiente información para que otros puedan ofrecer mucha ayuda.

Tu respuesta a la segunda pregunta debe proporcionar suficientes detalles para que no te aconsejen repetir lo que ya has intentado: "Fui a <https://python.org/downloads> y pulsé el botón Descargar para mi sistema. Luego ejecuté el instalador" es más útil que "Fui al sitio web de Python y descargué algo".

Para la tercera pregunta, es útil conocer los mensajes de error exactos que recibiste, de modo que puedas utilizarlos para buscar una solución en Internet o proporcionarlos cuando pidas ayuda.

A veces, el mero hecho de responder a estas tres preguntas antes de pedir ayuda a los demás te permite ver algo que te falta, y te ayuda a desatascarte sin tener que ir más lejos. Los programadores incluso tienen un nombre para esto: *rubber duck debugging*. La idea es que si explicas claramente tu situación a un patito de goma (o a cualquier objeto inanimado) y le haces una pregunta concreta, a menudo serás capaz de responder a tu propia pregunta. Algunos equipos de programación incluso tienen cerca un pato de goma de verdad para animar a la gente a "hablar con el pato".

### Inténtalo de nuevo

Volver al principio e intentarlo de nuevo puede ser suficiente para resolver muchos problemas. Digamos que estás intentando escribir un bucle for basándote en un ejemplo de este libro. Puede que sólo hayas pasado por alto algo sencillo, como dos puntos al final de la línea for. Repasar los pasos de nuevo podría ayudarte a evitar repetir el mismo error.

### Tómate un descanso

Si llevas tiempo trabajando en el mismo problema, tomarte un descanso es una de las mejores tácticas que puedes probar. Cuando trabajamos en la misma tarea durante largos periodos de tiempo, nuestro cerebro empieza a centrarse en una única solución. Perdemos de vista las suposiciones que hemos hecho, y hacer una pausa nos ayuda a tener una nueva perspectiva del problema. No hace falta que sea una pausa larga, sólo algo que te saque de tu mentalidad actual. Si llevas mucho tiempo sentado, haz algo físico: da un breve paseo, sal un rato al exterior, o tal vez bebe un vaso de agua o come un tentempié ligero.

Si te sientes frustrado, quizá merezca la pena dejar el trabajo a un lado por hoy. Una buena noche de sueño casi siempre hace que un problema sea más abordable.

### Consulta los recursos de este libro

Los recursos en línea de este libro, disponibles en [https://ehmatthes.github.io/pcc\_3e,](https://ehmatthes.github.io/pcc_3e) incluyen una serie de secciones útiles sobre cómo configurar tu sistema y trabajar en cada capítulo. Si aún no lo has hecho, echa un vistazo a estos recursos y comprueba si hay algo que te ayude en tu situación.

## Búsqueda en Internet

Es muy probable que alguien haya tenido el mismo problema que tú y haya escrito sobre ello en Internet. Unas buenas habilidades de búsqueda e indagaciones específicas te ayudarán a encontrar recursos existentes para resolver el problema al que te enfrentas. Por ejemplo, si tienes problemas para instalar la última versión de Python en un nuevo sistema Windows, buscar *install python windows* y limitar los resultados a recursos del último año podría dirigirte a una respuesta clara.

Buscar el mensaje de error exacto también puede ser muy útil. Por ejemplo, supongamos que recibes el siguiente error cuando intentas ejecutar un programa Python desde un terminal en un nuevo sistema Windows:

> python hello\_world.py  
Python was not found; run without arguments to install from the Microsoft  
 Store...

Buscar la frase completa, "No se encontró Python; ejecutar sin argumentos para instalar desde la Tienda Microsoft", probablemente te dará algún buen consejo.

Cuando empieces a buscar temas relacionados con la programación, aparecerán repetidamente algunos sitios. Describiré brevemente algunos de estos sitios, para que sepas hasta qué punto pueden serte útiles.

### Stack Overflow

*Stack Overflow* [(https://stackoverflow.com)](https://stackoverflow.com) es uno de los sitios más populares de preguntas y respuestas para programadores, y a menudo aparecerá en la primera página de resultados de las búsquedas relacionadas con Python. Los miembros publican preguntas cuando están atascados, y otros miembros intentan dar respuestas útiles. Los usuarios pueden votar las respuestas que consideran más útiles, por lo que las mejores respuestas suelen ser las primeras que encontrarás.

Muchas preguntas básicas sobre Python tienen respuestas muy claras en Stack Overflow, porque la comunidad las ha perfeccionado con el tiempo. También se anima a los usuarios a publicar actualizaciones, por lo que las respuestas tienden a mantenerse relativamente actualizadas. En el momento de escribir esto, se han respondido casi dos millones de preguntas relacionadas con Python en Stack Overflow.

Hay una expectativa que debes tener en cuenta antes de publicar en Stack Overflow. Las preguntas deben ser el ejemplo más breve del tipo de problema al que te enfrentas. Si publicas entre 5 y 20 líneas de código que generen el error al que te enfrentas, y si respondes a las preguntas mencionadas en "Primeros pasos" en la página 477 de este apéndice, es probable que alguien te ayude. Si compartes un enlace a un proyecto con varios archivos grandes, es muy poco probable que la gente te ayude. Hay una guía estupenda para redactar una buena pregunta en [https://stackoverflow.com/help/how-to-ask.](https://stackoverflow.com/help/how-to-ask) Las sugerencias de esta guía son aplicables a la obtención de ayuda en cualquier comunidad de programadores.

### La documentación oficial de Python

La documentación oficial de Python[(https://docs.python.org)](https://docs.python.org) es un poco más imprecisa para los principiantes, porque su propósito es más documentar el lenguaje que proporcionar explicaciones. Los ejemplos de la documentación oficial deberían funcionar, pero es posible que no entiendas todo lo que se muestra. Aún así, es un buen recurso para consultar cuando aparezca en tus búsquedas, y te resultará más útil a medida que sigas ampliando tus conocimientos de Python.

### Documentación oficial de bibliotecas

Si estás utilizando una biblioteca específica, como Pygame, Matplotlib o Django, los enlaces a la documentación oficial de la misma aparecerán a menudo en las búsquedas. Por ejemplo, <https://docs.djangoproject.com> es muy útil cuando se trabaja con Django. Si piensas trabajar con alguna de estas bibliotecas, es una buena idea que te familiarices con su documentación oficial.

### r/learnpython

Reddit se compone de una serie de subforos denominados *subreddits*. El subreddit *r/learnpython* [(https://reddit.com/r/learnpython)](https://reddit.com/r/learnpython) es muy activo y solidario. Puedes leer las preguntas de los demás y publicar también las tuyas. A menudo obtendrás múltiples perspectivas sobre las preguntas que plantees, lo que puede ser realmente útil para comprender mejor el tema en el que estás trabajando.

### Publicaciones en blogs

Muchos programadores mantienen blogs y comparten posts sobre las partes del lenguaje con las que están trabajando. Deberías buscar una fecha en las entradas de blog que encuentres, para ver hasta qué punto es probable que la información sea aplicable a la versión de Python que estás utilizando.

## Discord

*Discord* es un entorno de chat en línea con una comunidad Python donde puedes pedir ayuda y seguir discusiones relacionadas con Python.

Para comprobarlo, dirígete a <https://pythondiscord.com> y haz clic en el enlace **Discord** de la parte superior derecha. Si ya tienes una cuenta de Discord, puedes conectarte con ella. Si no tienes una cuenta, introduce un nombre de usuario y sigue las indicaciones para completar tu registro en Discord.

Si es la primera vez que visitas el Discord de Python, tendrás que aceptar las normas de la comunidad antes de participar plenamente. Una vez hecho esto, puedes unirte a cualquiera de los canales que te interesen. Si buscas ayuda, asegúrate de publicar en uno de los canales de Ayuda de Python.

## Slack

*Slack* es otro entorno de chat online. Suele utilizarse para las comunicaciones internas de la empresa, pero también hay muchos grupos públicos a los que puedes unirte. Si quieres echar un vistazo a los grupos Slack de Python, empieza por [https://pyslackers.com.](https://pyslackers.com) Haz clic en el enlace **Slack** de la parte superior de la página y, a continuación, introduce tu dirección de correo electrónico para obtener una invitación.

Una vez que estés en el espacio de trabajo Desarrolladores Python, verás una lista de canales. Haz clic en **Canales** y elige los temas que te interesen. Quizá quieras empezar por los canales *#help* y *#django*.