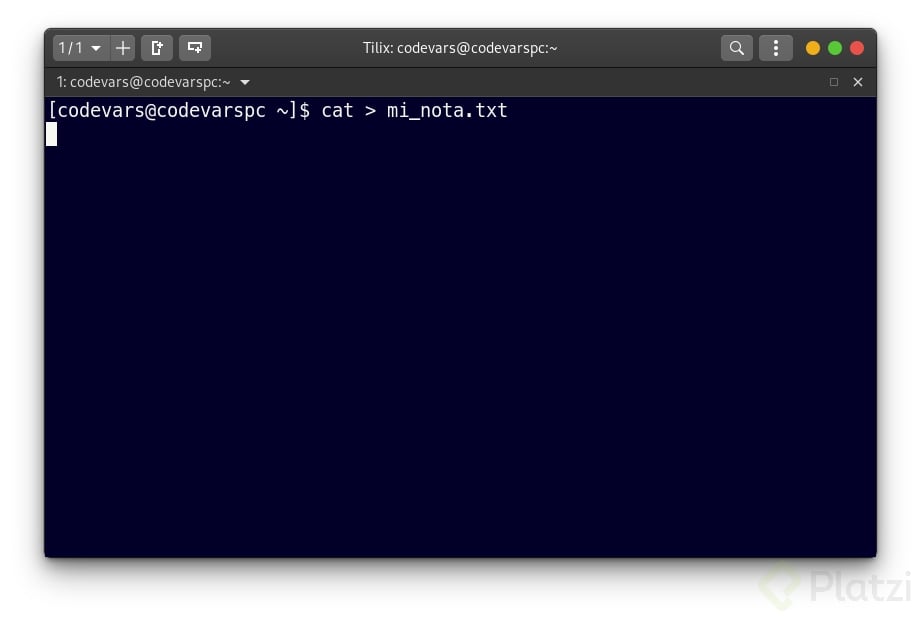
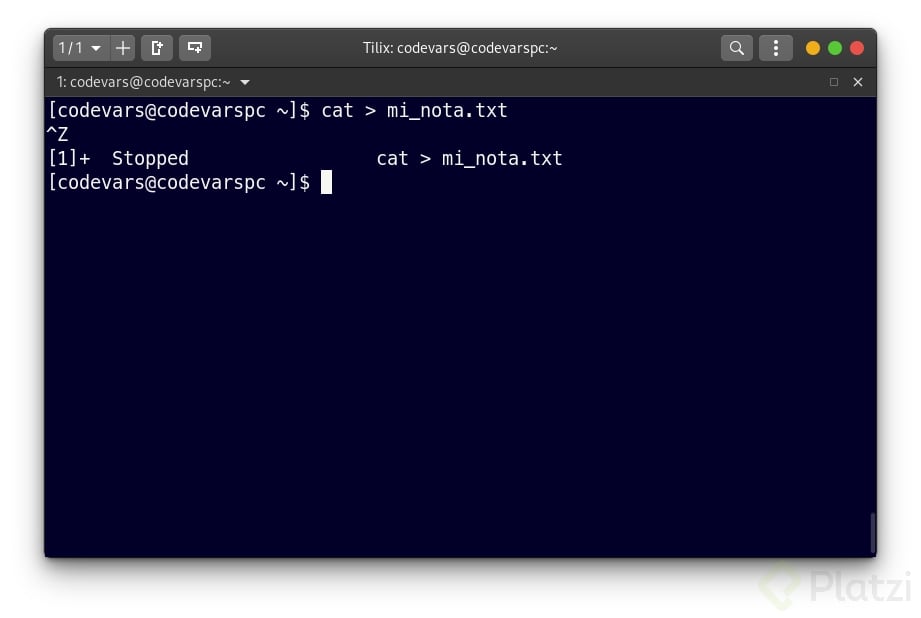
Como viste en la clase de procesos podemos correr de manera asíncrona comandos, y si estos no se completan quedarán activos dentro de los procesos de la terminal.

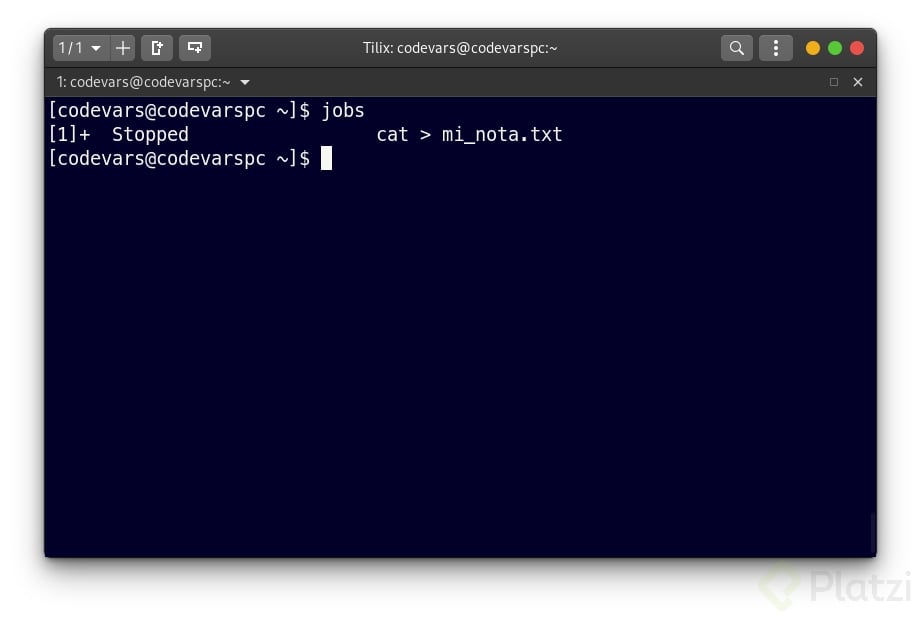
Cuando un proceso está en ejecución sin que sea mostrado en la terminal se dice que se está ejecutando en el background. Si se muestra la ejecución del comando dentro de la terminal se dice que está en el foreground. En esta clase aprenderás a cómo mover los procesos del background al foreground a tu voluntad, incluso a cómo suspenderlos.

¿Te acuerdas del truco que aprendimos para tener un editor de texto supersencillo en la terminal? Lo usaremos en esta ocasión. Imagina que queremos una nota desde la terminal y para eso usamos:

cat > mi\_nota.txt

Nuestra terminal se verá de la siguiente manera, con el prompt esperando a que ingresemos texto.  


Podemos escribir algo y después terminar el input del texto con CTRL+D, pero en esta ocasión no haremos eso. Lo que queremos hacer será suspender el proceso, esto lo podemos hacer con CTRL+Z. El resultado que nos mostrará la terminal deberá ser uno donde nos indique la suspensión del comando cat.  


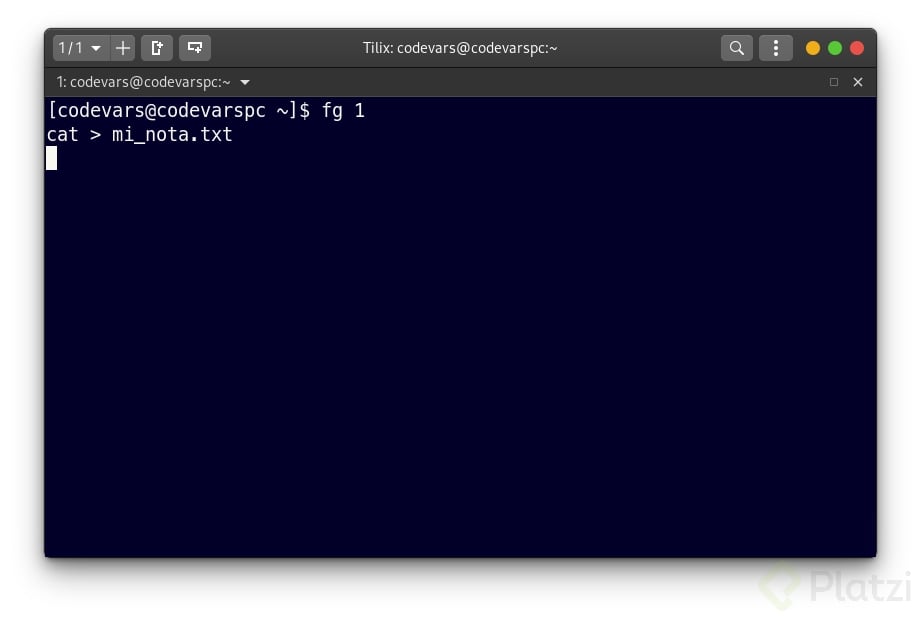
Ahora hemos movido nuestro comando exitosamente al background de la terminal. Para consultar todos los procesos que tenemos en background podemos hacerlo con el comando jobs.  


A la izquierda aparece el número del trabajo ( ⚠ ️ cuidado que no es lo mismo que el process ID). Si queremos traer la ejecución de nuevo a la terminal, es decir, al foreground; debemos usar el comando fg y especificar qué número de trabajo queremos continuar. Para nuestro caso será el 1.

fg 1

En caso de que estés usando ZSH como shell el formato para llamar el trabajo sería con un porcentaje. ZSH tiende a interpretar algunas cosas incluyendo las wildcards de manera diferente.

fg %1

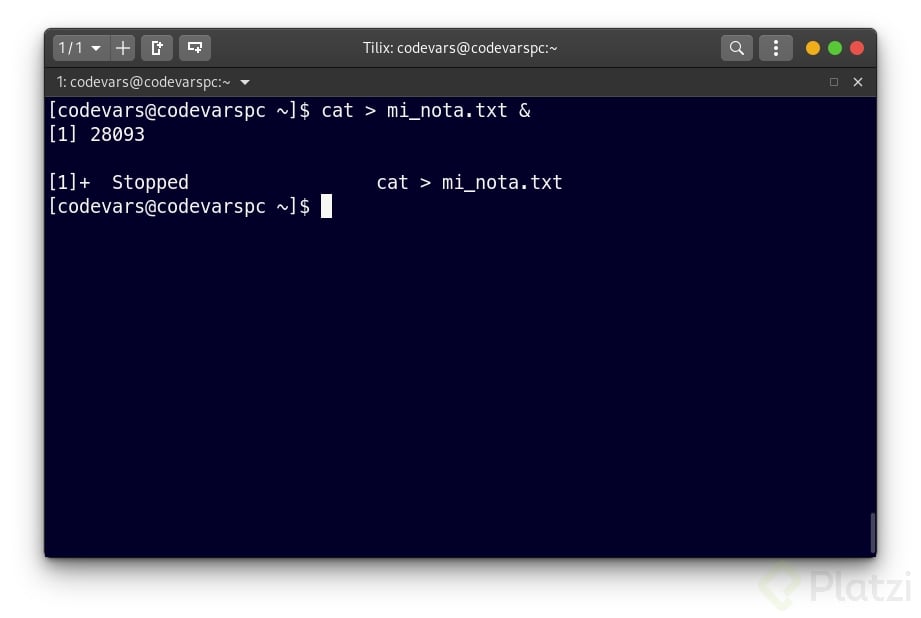
Una vez enviado al foreground veremos como se activa la ejecución del comando en la terminal y podremos seguir escribiendo nuestra nota. Recuerda que una vez terminemos de escribir presionamos CTRL+D para terminar el input y guardar.  


Cuando se guarda nuestra nota nos daremos cuenta de que el proceso por fin termina y si usamos jobs no nos mostrará ningún trabajo en background.

**Otras formas de enviar al background**

Existen otras formas de enviar comandos al background. La primera es usando el operador de control & al final de un comando. Este operador nos permite enviar de manera directa un proceso al background una vez ejecutado. Por ejemplo:

cat > mi\_nota.txt &



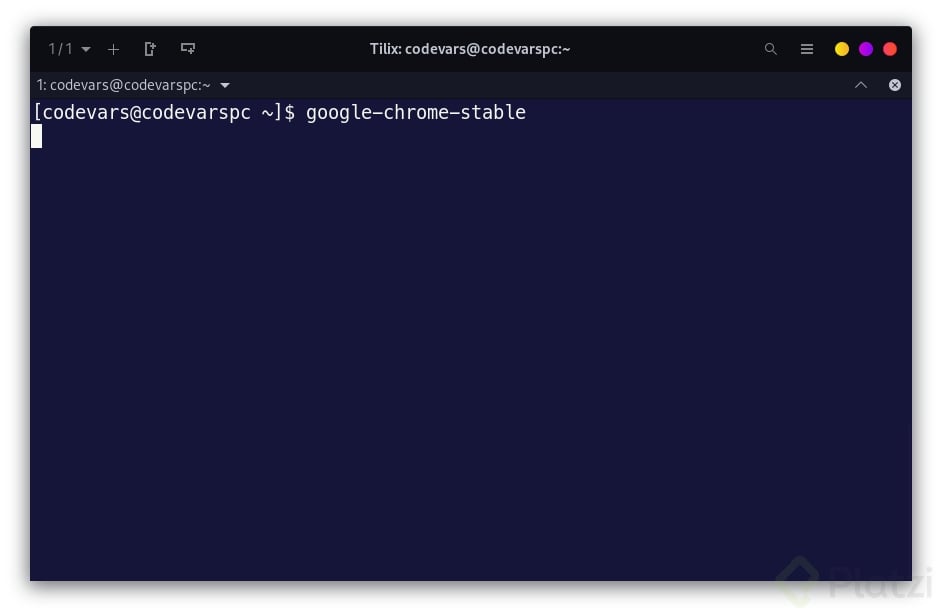
La segunda forma es con el comando bg. Este sirve de manera similar que fg solo que en vez de traerlo al foreground este lleva un trabajo al background. Por ejemplo:

**bg** 1

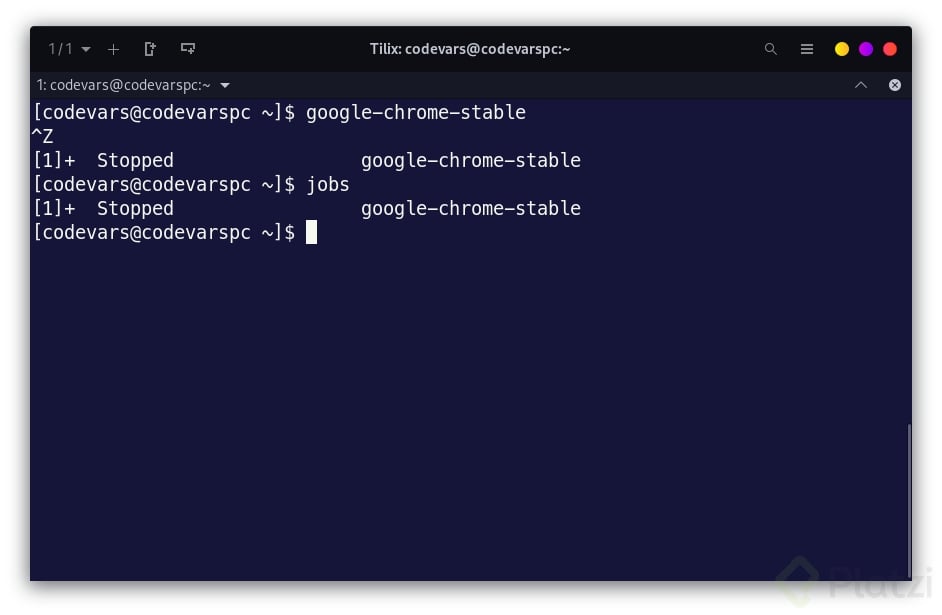
Bien, la pregunta ahora es ¿Cómo usamos bg? Imagina que abrimos algún programa de interfaz gráfica desde la terminal. En mi caso abriré el navegador Google Chrome. Para hacerlo desde la terminal solo ejecuta:

google-chrome-stable

Y verás como se ejecuta pero no nos deja hacer ninguna otra tarea ya que la ventana del navegador está abierta:



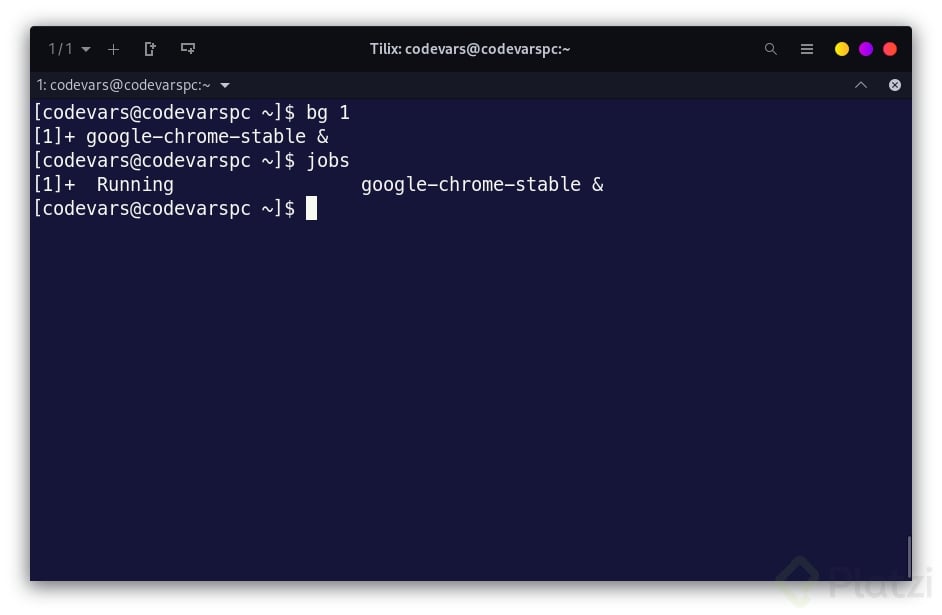
Para suspender el proceso como ya sabes lo hacemos con CTRL+Z y si revisamos con jobs veremos como el proceso se encuentra en pausa. En este caso la ventana del navegador que se abrió no nos dejará interactuar ni escribir en ella.



Como se ve en la imagen el navegador tiene el número de trabajo 1. Para dejar nuestro navegador corriendo y al mismo tiempo seguir trabajando en la terminal tenemos que reactivar este proceso y a la vez mandarlo al background. Para ello ejecutamos:

**bg** 1

Con esto podremos ver como nuestro proceso de Google Chrome sigue corriendo en el background dejando la terminal disponible para nosotros.



¡Genial! Con esto ya sabes cómo mover procesos dentro de la terminal del foreground al background. Esto es muy útil cuando solo tenemos una terminal y necesitamos ejecutar varios comandos en paralelo. ¡Te espero en la siguiente clase!