

# Precompilado de paquetes

Hemos visto que cuando un ambiente independiente no cuenta con un módulo o paquete, el gestor de paquetes se encarga de descargar la versión especificada en los archivos TOML, o cuando es un ambiente vacío, descarga la versión más reciente y lo agrega a los archivos TOML. Cuando en la máquina en la que trabajamos el paquete había sido previamente descargado (para otro ambiente, por ejemplo), Julia no lo vuelve a descargar, pero sí lleva tiempo cargarlo (usando `using` o `import`).

Además, cada vez que abrimos una nueva sesión de Julia, la primera vez que mandamos a llamar a una función, Julia la vuelve a compilar. Esto puede llevar un tiempo considerable (por ejemplo, para el paquete `Plots`).

Podemos tomar el tiempo de carga de este paquete en una sesión de Julia nueva con la macro `@time`. En un directorio vacío llamado `Prueba`, iniciamos el gestor de paquetes, activamos el ambiente y agregamos el paquete `Plots`.

```
(@v1.5) pkg> activate .  
Activating new environment at `D:\Google Drive\0000CIC\CORE2020\Taller_Julia\projects\Prueba\Project.toml`  
  
(Prueba) pkg> add Plots
```

Después, podemos cargar el módulo en el REPL tomándole el tiempo con la macro `@time`:

```
julia> @time using Plots  
17.344243 seconds (13.24 M allocations: 783.438 MiB, 2.58% gc time)  
  
julia> @time using Plots  
0.045678 seconds (38.93 k allocations: 1.966 MiB)
```

Vemos que le lleva más de 17 segundos a Julia cargar `Plots` en este ambiente. Además, necesita más de 13 millones de *allocations* de memoria y casi 800 megabytes. Si lo hacemos nuevamente, el tiempo y los recursos de memoria son solo una fracción.

## PackageCompiler.jl

El paquete [PackageCompiler.jl](#) nos permite tener una imagen del sistema de esta compilación para un ambiente dado. De esta forma, se puede pasar la indicación al momento de arrancar un REPL de Julia que precargue esta imagen y los tiempos de carga de los paquetes de nuestro ambiente se reducirá drásticamente.

Pueden ver la documentación del paquete (al momento de escribir esto aún está en etapa de desarrollo) y puede ser un poco complicado el proceso. Cabe señalar, que

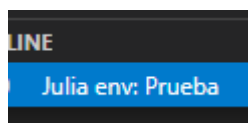
en caso de que algún paquete del ambiente de actualice o se instale después de la creación de la imagen, es necesario volver a crear la imagen.

Además de lo anterior, este paquete permite generar archivos binarios de alguna aplicación de Julia para ejecutarla sin necesitar abrir una sesión de Julia.

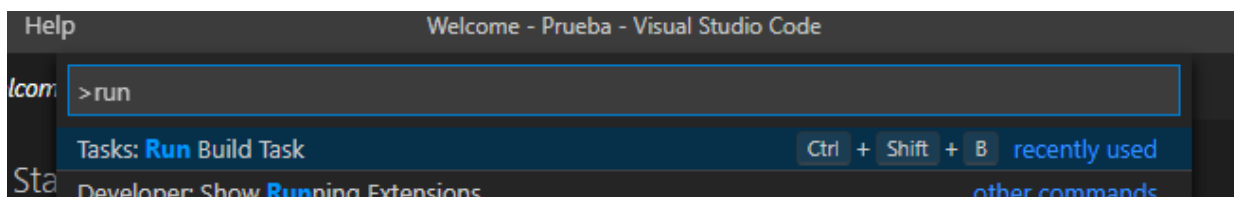
## La forma fácil en VS Code

El *plug-in* de Julia para Visual Studio Code cuenta con una funcionalidad experimental (al momento de escribir esto) para generar la imagen del sistema para un ambiente dado.

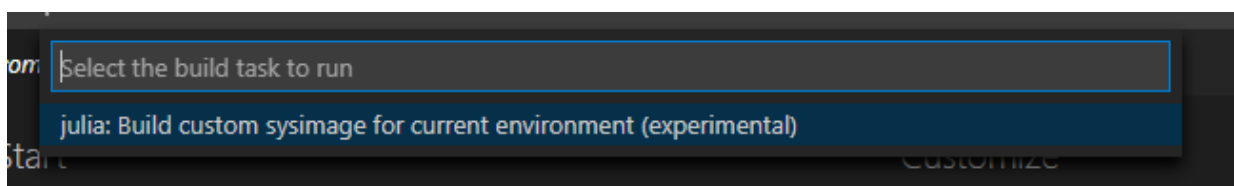
**NOTA:** Asegúrate de seleccionar el ambiente en la barra de estado inferior de VS Code



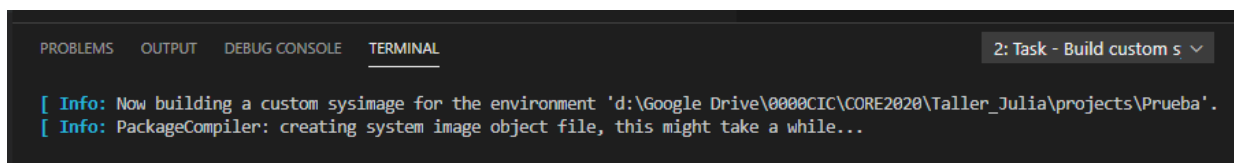
1. En la paleta de comandos de VS Code (*ctrl + shift + p*), seleccionamos "Tasks:Run Build Task"



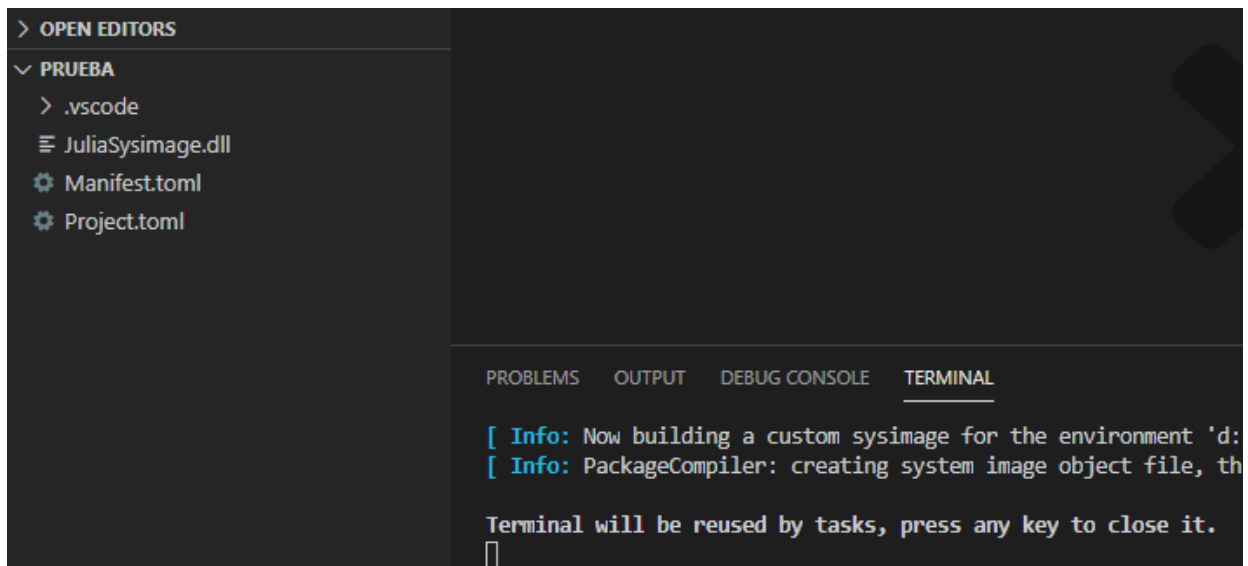
2. Aparecerá en la parte superior la opción "julia: Build custom sysimage for current environment (experimental)". Selecciona esta opción.



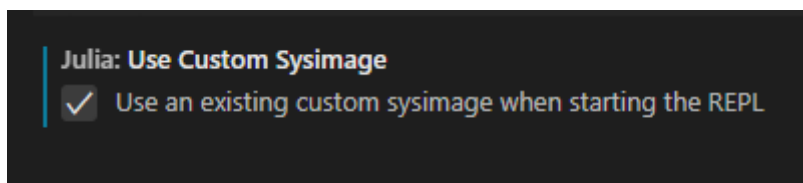
3. En la terminal aparecerá el mensaje que nos dice que se está construyendo la imagen y nos da el directorio del ambiente que está contruyendo y que esté proceso puede demorar.



4. Al término del proceso, se habrá generado un archivo DLL en el directorio del ambiente y en l terminal habrá un mensaje que nos dice que presionemos cualquier tecla para cerrar esa terminal.

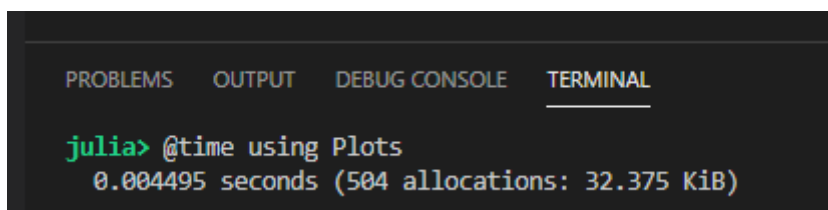


5. En la configuración de Julia (engrane → *settings* ) cerciórate que la siguiente opción está seleccionada:



Esta opción hace que VS Code busque un archivo de imagen del sistema (el DLL que acabamos de generar) y lo cargue al iniciar el REPL de Julia.

6. Ahora, podemos abrir una nueva sesión del REPL de Julia y automáticamente cargará la DLL.



¡Es casi instantáneo cargar ahora **Plots** !

**NOTA:** Toma en cuenta que una nueva imagen del sistema debe generarse si actualizas algún paquete del ambiente o si agregas paquetes así como el tamaño del archivo de imagen.