

Archivo de Configuración para Proyecto Embebido

Este documento describe la configuración JSON proporcionada para un proyecto embebido, específicamente para la plataforma Win32.

Estructura del Archivo

```
{
  "configurations": [
    {
      "name": "Win32",
      "includePath": [
        "${workspaceFolder}/**"
      ],
      "defines": [
        "_DEBUG",
        "UNICODE",
        "_UNICODE"
      ]
    }
  ],
```

"version he JSON snippet you provided is a configuration setting for Visual Studio Code (VS Code), specifically for the C/C++ extension. This setting disables the error squiggles (red underlines) that appear in the editor when there are syntax or semantic errors in your C/C++ code.

Explanation:

```
{
  "C_Cpp.errorSquiggles": "disabled"
}
```

- "C_Cpp.errorSquiggles": This is the setting key for controlling error squiggles in the C/C++ extension.
- "disabled": This value turns off the error squiggles. Other possible values are:

- "enabled" (default): Shows error squiggles.
- "disabledIfIncludesResolve": Disables squiggles only if include files are resolved correctly.

How to Use:

1. Open VS Code settings:
 - **GUI:** Go to File > Preferences > Settings (or Code > Preferences > Settings on macOS), search for "errorSquiggles," and modify the setting.
 - **JSON:** Open the settings.json file (Ctrl + , → click the "Open Settings (JSON)" icon) and add/update this entry.

When to Use This:

- You might disable error squiggles if:
 - You're working with a codebase that has known issues but compiles fine.
 - You're using a custom build system, and the IntelliSense engine reports false positives.
 - You find the squiggles distracting during development.

However, disabling this feature means you won't get real-time feedback on potential errors, so use it cautiously

": 4

}

Explicación de Campos

1. configurations (**Configuraciones**)
 - Array que contiene una o más configuraciones de compilación.
2. name: "**Win32**"
 - Nombre identificador de esta configuración específica.
 - Indica que esta configuración es para el sistema operativo Windows de 32 bits.
3. includePath (**Rutas de Inclusión**)
 - Especifica los directorios donde el compilador buscará los archivos de cabecera (.h).

- "\${workspaceFolder}/**":
 - \${workspaceFolder} es una variable que representa la ruta raíz del proyecto.
 - /** indica que se deben incluir todos los subdirectorios recursivamente.

4. defines (**Definiciones de Preprocesador**)

- Lista de macros que se definen para el preprocesador:
 - "_DEBUG": Habilita el modo de depuración.
 - "UNICODE" y "_UNICODE": Habilitan el soporte para caracteres Unicode en Windows.