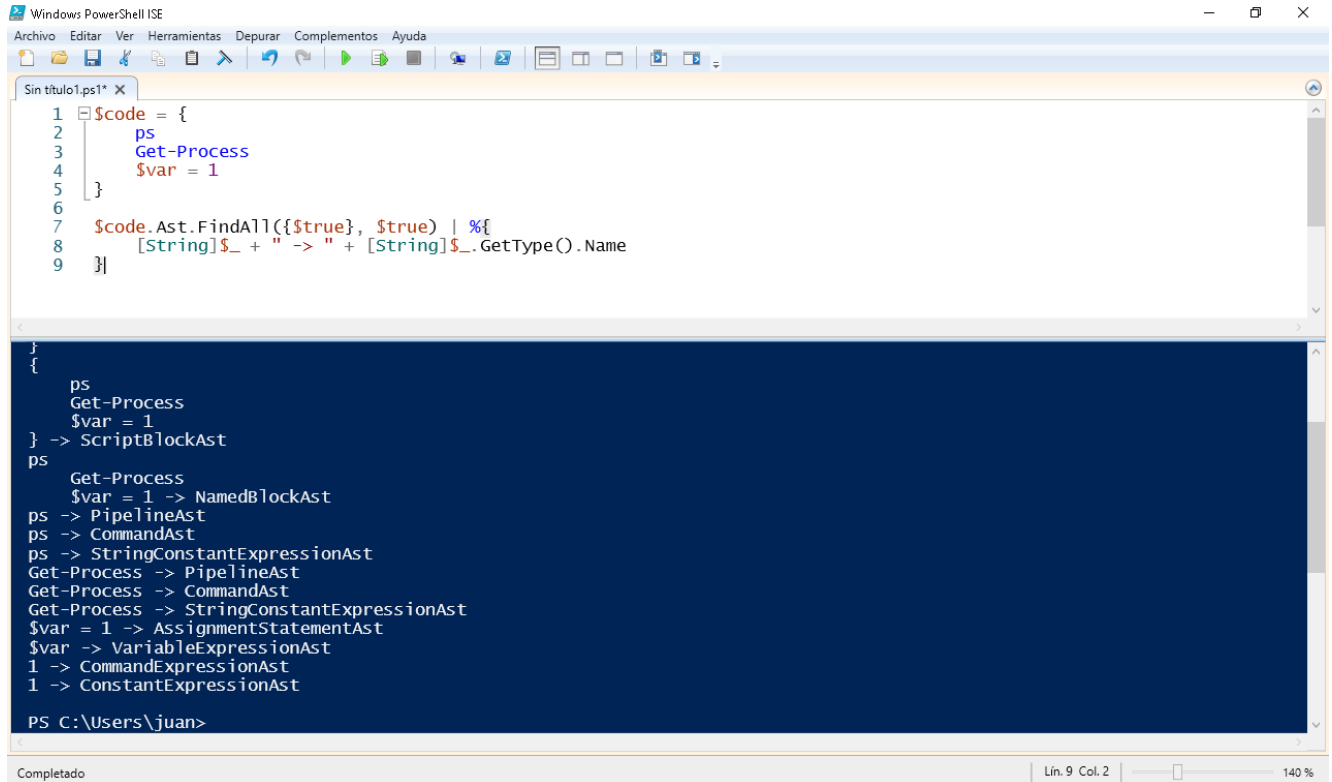


Analizar un bloque de código en PowerShell (cmdlets, variables, asignaciones)

[crayon-6a9d676b9d4a2254744002/]



The screenshot shows the Windows PowerShell ISE interface. The top pane displays a PowerShell script named 'Sin titulo1.ps1'. The script defines a function '\$code' which contains a 'ps' block with 'Get-Process' and '\$var = 1'. Below this, it uses '\$code.Ast.FindAll' to find all AST nodes of type 'true' and formats them into a string. The bottom pane shows the output of this command, which is a detailed list of AST nodes and their types, such as 'ScriptBlockAst', 'NamedBlockAst', 'PipelineAst', 'CommandAst', 'StringConstantExpressionAst', 'AssignmentStatementAst', 'VariableExpressionAst', 'CommandExpressionAst', and 'ConstantExpressionAst'. The status bar at the bottom indicates 'Completado' and 'Lín. 9 Col. 2'.

```
1 $code = {  
2     ps  
3     Get-Process  
4     $var = 1  
5 }  
6  
7 $code.Ast.FindAll({$true}, $true) | %{  
8     [String]$_ + " -> " + [String]$.GetType().Name  
9 }
```

```
}  
{  
    ps  
    Get-Process  
    $var = 1  
} -> ScriptBlockAst  
ps  
    Get-Process  
    $var = 1 -> NamedBlockAst  
ps -> PipelineAst  
ps -> CommandAst  
ps -> StringConstantExpressionAst  
Get-Process -> PipelineAst  
Get-Process -> CommandAst  
Get-Process -> StringConstantExpressionAst  
$var = 1 -> AssignmentStatementAst  
$var -> VariableExpressionAst  
1 -> CommandExpressionAst  
1 -> ConstantExpressionAst  
  
PS C:\Users\juan>
```

Completado | Lín. 9 Col. 2 | 140 %