

Ejercicio 11. Productividad en Visual Studio Code



En este ejercicio expondremos algunas herramientas de productividad que nos permiten hacer nuestro trabajo como desarrolladores de Business Central de forma más sencilla, rápida y controlada.

Podemos agrupar las distintas herramientas de productividad en los siguientes grupos:

- Visual Studio Code
- Extensiones para Visual Studio Code
- Control de código fuente con GIT
- AL Language
- Test Driven Development (TDD)
- PowerShell

A continuación detallaremos algunos de los aspectos de cada uno de los grupos.

Visual Studio Code

1. Renombrar un símbolo (variable, procedimiento, objeto o control)

Comando vscode *Rename Symbol* (F2)

Se puede renombrar desde la definición del símbolo, o desde cualquier sitio en el que se use el símbolo.

2. Replicar una línea

Alt + Shift + FlechaAbajo

Alt + Shift + FlechaArriba

3. Seleccionar múltiples columnas

Alt + Shift

4. Cambiar el *Ajuste de línea*

Alt + Shift

5. Copiar o cortar una línea entera, sin necesidad de seleccionar

Ctrl + C

Ctrl + X

6. Mover Línea

Alt + FlechaArriba

Alt + FlechaAbajo

7. Seleccionar Ocurrencias

Ctrl + D

8. Buscar y reemplazar en múltiples ficheros

Ctrl + Shift + F

Ctrl + Shift + H

Con las expresiones regulares tenemos mucha potencia de búsqueda y reemplazo.

9. Ocultar / Mostrar el menu lateral

Ctrl + B

10. Abrir el explorador de archivos

Ctrl + Shift + E

11. Abrir el área de GIT

Ctrl + Shift + G

12. Abrir el Terminal

Ctrl + Ñ

Ctrl + Shift + Ñ (crear nuevo)

13. Abrir el área de problemas

Ctrl + Shift + M

14. Dividir el editor

**View: Split Editor **

Ctrl + °

View: Move Editor Group

15. Poner foco en otro editor

Ctrl + 2

16. Buscar ficheros por nombre

Ctrl + P

F1 y borrar el símbolo >

17. Ir a la última posición del cursor

Alt + Flechalzquierda

Alt + FlechaDerecha

18. Ir a la definición

F12

19. Encontrar todas las referencias

Shift + Alt + F12

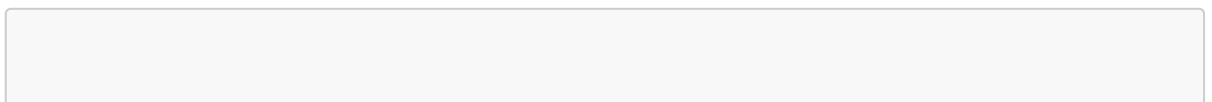
20. Comentar / Descomentar una línea

Ctrl + Ç

Extensiones para Visual Studio Code

1. Waldo's CRS AL Language Extension

1. *CRS: Rename File*



```
"CRS.OnSaveAlFileAction": "Rename",
"CRS.RemovePrefixFromFilename": true,
"CRS.FileNamePattern": "<ObjectNameShort>."
<ObjectTypeShortPascalCase>.al",
"CRS.FileNamePatternExtensions": "<BaseNameShort>."
<ObjectTypeShortPascalCase>.al",
"CRS.ObjectNamePrefix": "Clip"
```

2. CRS: Search Microsoft Docs

2. AL Variable Helper

1. Give Variable Names

vFull

vShort

vTag

2. GoTo Locals

Ctrl + L

3. GoTo Globals

Ctrl + G

4. AL-Var: Sort Variables

3. AL CodeActions

1. Create Procedure

2. Extract to Procedure

4. Todo Tree

1. Ver tags en forma de árbol

2. Personalizar tags

5. Error Lens

1. Resaltar errores

Control de código fuente con GIT

1. Integración nativa con vscode

2. Ejecutar comandos desde el terminal

3. Validar cambios antes de hacer un commit

4. Trabajar en una rama aislada

5. Deshacer cambios antes de hacer un commit

6. Montar un nuevo entorno muy fácilmente

AL Language

1. Analizadores de código

2. Snippets
3. Mostrar errores sin compilar
4. Nuevas posibilidades de desarrollo:
 1. Sobrecarga de funciones
 2. Interficies
 3. Enum
 4. Suscribirse a eventos
 5. Lanzar eventos
 6. Patrón Handled
 7. Un procedimiento > Una responsabilidad

Test Driven Development TDD

1. No depender de los datos
2. No depender de las configuraciones
3. Testear muchos escenarios en muy poco tiempo
4. Confianza en ampliar la capacidad de la arquitectura
5. Confianza en los refactoring
6. Confianza en no romper código antiguo con los nuevos requisitos
7. Feedback muy rápido
8. Asegurar la calidad del código
9. Montar un entorno nuevo muy rápido con una Cronus
10. Hacer un test específico por cada bug

PowerShell

1. Cada vez se pueden gestionar más escenarios desde PowerShell
2. Crear scripts para automatizar tareas
3. Tener una colección de instrucciones habituales