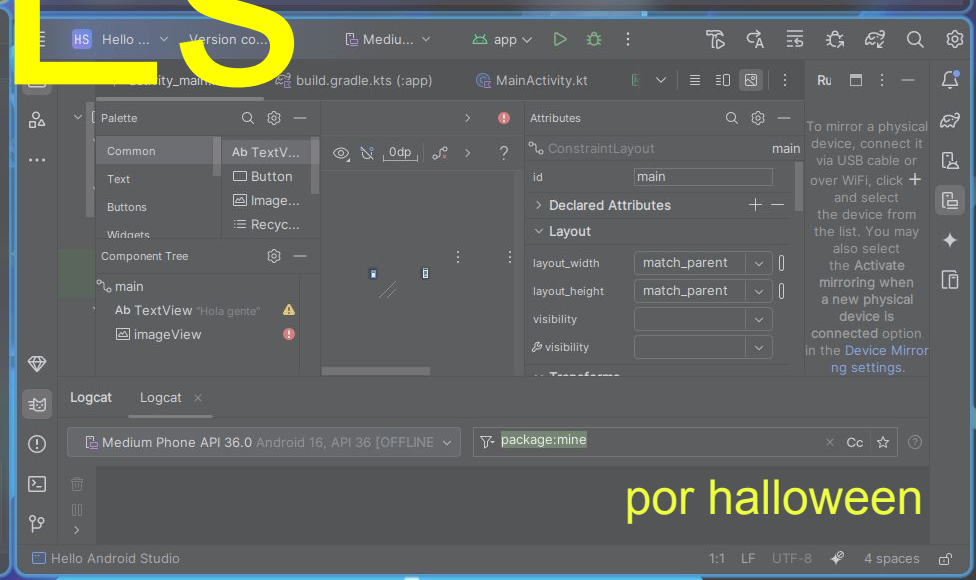
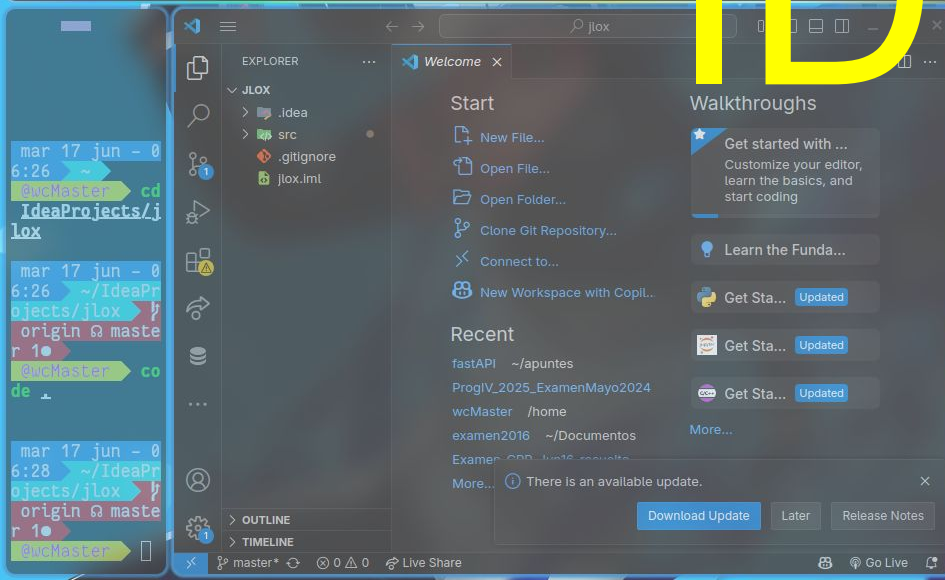


IDES



ÍNDICE

- > IDEs y programar en terminal
- > Vi
- > Vim
- > Configuración fácil de neoVim



IDE

->I

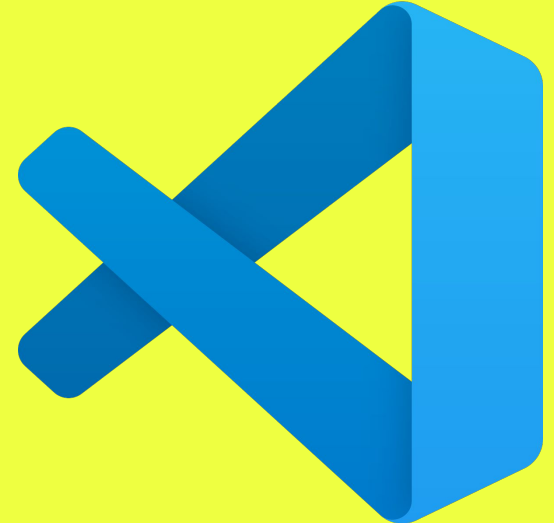
->D

->E

eclipse->



android
studio->



IDE

->Entorno

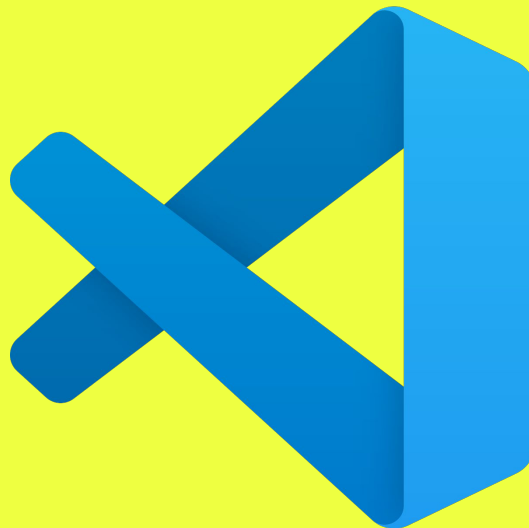
->de Desarrollo

->Integrado

JetBrains->



vscode->



Porque programar a terminal

- > Velocidad (usar el ratón es obviamente lento)
- > Menor consumo de recursos

Programmers 1960s



With this software we shall
fly to the moon and back

Programmers 2020



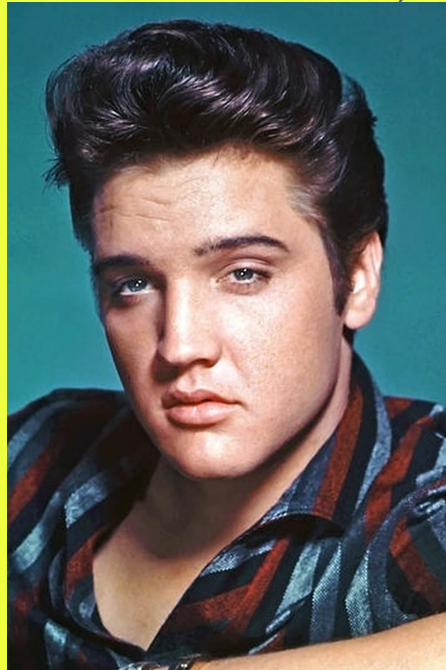
Halp me pliz, I can't
exit vim.



VI

-> para Unix

1976 -> Bill Joy



Comandos de VI

Comandos de Salida					
zz, :x, :wq	escribe (guarda) el fichero y lo cierra [los 3 casos]	:q	cierra el fichero	:e #	editar fichero opcional
:w	grabar fichero	:q!	cerrar el fichero desechando los cambios no guardados	:vi	invocar el editor vi desde ex
:w!	guardar el fichero sin hacer caso de la protección contra escritura	Q	salir de vi y arrancar ex	:	usar un comando ex en el editor vi
:30,60w nuevofichero	guardar de la línea 30 a la 60 como nuevo fichero	:n	editar el siguiente fichero	#	alternate filename (substitutes into ex command)
:30,60w>> fichero	añadir las líneas 30 a 60 a un fichero	:e fichero2	editar fichero2 sin salir de vi	%	current filename (substitutes into ex command)
:w %nuevo	escribe el contenido del búfer de edición a un fichero llamado fichero.nuevo	:e!	devuelve el fichero al estado en que se encontraba cuando se guardó por última vez		

Comandos de vi	
h, j, k, l	izquierda, abajo, arriba, derecha
w, W, b, B	adelante y atrás palabra por palabra
e, E	ir a final de palabra
), (inicio de la frase siguiente, de la frase anterior
}, {	inicio del párrafo siguiente, del párrafo anterior
]], [[inicio de la siguiente sección, de la sección anterior
o, \$	primera y última posición de la línea donde está el cursor
^	primer carácter alfanumérico de la línea
+, -	primer carácter alfanumérico de la línea siguiente, de la línea anterior
n	columna n de la línea donde está el cursor
H, M, L	Primera línea, Línea central, Última línea de la pantalla
nH, nL	n número de líneas tras la Primera o antes de la Última línea

Números de Línea en vi	
ctrl G	Muestra el número de línea donde está el cursor
nG	ir la la línea número n
G	ir la la última línea del fichero
:n	ir a la línea n del fichero

Inserción	
i, a	Insertar texto Antes, Después del cursor
I, A	insertar texto Antes del principio, Tras el final de la línea
o, O	crear una nueva línea Encima, Debajo del cursor

Copiar (yank)	
yw	copiar palabra
yy	copiar la línea donde está el cursor
"ayy	copiar la línea donde está el cursor al búfer a (a-z). los nombres en mayúsculas añaden texto
ymovimiento	copia el texto entre el cursor y el destino del movimiento
p, P	pegar el texto copiado Antes, Después del cursor
"aP	pegar texto del búfer a (a-z) Antes del cursor

https://tikismikis.org/vim/vi_cheats.html

VIM





Documentarse de VIM

Guia → https://victorhck.gitbook.io/aprende-vim/cap01_empezando_en_vim

Documentacion oficial -> <https://www.vim.org/docs.php>

```
~  
NORMAL file1.js  
:buffers  
 1 %a "file1.js" line 1  
 2 "file2.js" line 0  
Press ENTER or type command to continue
```

comando de buffers mostrando 2 buffers

Hay varias formas de poder desplazarse por los *buffers*:

- `:bnext` o de manera abreviada puedes utilizar `:bn` para ir al *buffer* siguiente (`:bprevious` o `:bp` para ir al *buffer* previo).
- `:buffer` + nombre_de_archivo. De manera abreviada puedes utilizar `:b`. Vim puede autocompletar el nombre de archivo con `Tab`.
- `:buffer` + `n`, donde `n` es el número del *buffer*. También puedes reemplazar `:buffer` por `:b`. Por ejemplo, escribiendo: `:buffer 2` o `:b 2` te llevará al *buffer* número 2.
- Salta a la antigua posición en la lista de saltos con `Ctrl-O` y a la nueva posición con `Ctrl-I`. Estos no son métodos específicos para *buffers*, pero pueden ser utilizados para saltar entre diferentes *buffers*. Explicaré más detalles sobre los saltos en el capítulo 5.
- Ir al *buffer* previamente editado con `Ctrl-^`.

```
vim -o2
```

Para abrir Vim con 5 ventanas horizontales, ejecuta:

```
vim -o5
```

Para abrir Vim con 5 ventanas horizontales y ocupar las dos primeras con los archivos `hola1.txt` y `hola2.txt`, ejecuta:

```
vim -o5 hola1.txt hola2.txt
```

Para abrir Vim con dos ventanas verticales, 5 ventanas verticales y 5 ventanas verticales con 2 archivos:

```
vim -O2  
vim -O5  
vim -O5 hola1.txt hola2.txt
```

NVIM y NVCHAD

