

Sumario

i3 Guía del Usuario.....	4
1. Asignaciones de teclas predeterminadas.....	4
2. Usando i3.....	5
2.1. Abrir terminales y moverse.....	5
2.2. Cambiar el diseño del contenedor.....	6
2.3. Alternar el modo de pantalla completa para una ventana.....	7
2.4. Abrir otras aplicaciones.....	7
2.5. Cerrar ventanas.....	7
2.6. Usando espacios de trabajo.....	7
2.7. Mover ventanas a espacios de trabajo.....	8
2.8. Redimensionando.....	8
2.9. Reiniciar i3 en el lugar.....	8
2.10. Salir de i3.....	8
2.11. Flotante.....	8
3. Árbol.....	9
3.1. El árbol consta de contenedores.....	9
3.2. Orientación y contenedores divididos.....	9
3.3. Padre de enfoque.....	10
3.4. Contenedores implícitos.....	11
4. Configurando i3.....	11
4.1. Comentarios.....	12
4.2. Fuentes.....	12
4.3. Enlaces de teclado.....	13
4.4. Fijaciones de mouse.....	14
4.5. Modos de enlace.....	14
4.6. El modificador flotante.....	15
4.7. Restricción del tamaño de la ventana flotante.....	15
4.8. Orientación para nuevos espacios de trabajo.....	16
4.9. Modo de diseño para nuevos contenedores.....	16
4.10. Alineación de título de ventana.....	16
4.11. Estilo de borde predeterminado para ventanas nuevas.....	16
4.12. Ocultar bordes adyacentes a los bordes de la pantalla.....	17
4.13. Comandos arbitrarios para ventanas específicas (for_window).....	17
4.14. No enfoque la ventana al abrir.....	18
4.15. Variables.....	18
4.16. X recursos.....	18
4.17. Poner automáticamente a los clientes en espacios de trabajo específicos.....	19
4.18. Inicio automático de aplicaciones al iniciar i3.....	21
4.19. Poner automáticamente espacios de trabajo en pantallas específicas.....	22
4.20. Cambio de colores.....	22
4.21. Comunicación entre procesos.....	23
4.22. El foco sigue al mouse.....	24
4.23. Deformación del ratón.....	24
4.24. Popups durante el modo de pantalla completa.....	24
4.25. Enfoque de envoltura.....	25
4.26. Forzando Xinerama.....	25
4.27. Avance y retroceso automático al cambiar al espacio de trabajo actual.....	26
4.28. Retrasar el restablecimiento de la sugerencia de urgencia en el cambio del espacio de trabajo.....	26
4.29. Centrarse en la activación de la ventana.....	26

4.30. Dibujar marcas en la decoración de ventanas.....	27
4.31. Continuación de línea.....	27
5. Configurando i3bar.....	27
5.1. Comando i3bar.....	28
5.2. Comando statusline.....	28
5.3. Modo de visualización.....	28
5.4. Comandos del botón del mouse.....	29
5.5. ID de barra.....	30
5.6. Posición.....	30
5.7. Salida (s).....	31
5.8. Salida de la bandeja.....	32
5.9. Bandeja acolchada.....	33
5.10. Fuente.....	33
5.11. Símbolo separador personalizado.....	33
5.12. Botones del espacio de trabajo.....	33
5.13. Franja números / nombre del espacio de trabajo.....	34
5.14. Indicador de modo de enlace.....	34
5.15. Indicador de modo de enlaces.....	35
5.16. Colores.....	35
5.17. Transparencia.....	37
6. Lista de comandos.....	37
6.1. Ejecutando aplicaciones (exec).....	39
6.2. División de contenedores.....	40
6.3. Diseño de manipulación.....	41
6.4. Contenedores de enfoque.....	42
6.5. Mover contenedores.....	43
6.6. Intercambiar contenedores.....	44
6.7. Ventanas flotantes adhesivas.....	45
6.8. Cambio de espacios de trabajo (con nombre) / traslado a espacios de trabajo.....	45
6.8.1. Espacios de Trabajos Nombrados.....	46
6.8.2. Renombrar espacios de trabajo.....	47
6.9. Mover espacios de trabajo a una pantalla diferente.....	47
6.10. Mover contenedores/espacios de trabajo a salidas RandR.....	48
6.11. Mover contenedores / ventanas a marcas.....	48
6.12. Cambio de tamaño de contenedores / ventanas.....	49
6.13. Saltar a ventanas específicas.....	49
6.14. Marcas similares a VIM (mark/goto).....	50
6.15. Formato de título de ventana.....	51
6.16. Cambiar estilo de borde.....	52
6.17. Habilitar el registro de memoria compartida.....	52
6.18. Habilitar el registro de depuración.....	53
6.19. Recarga / reinicio / salida.....	53
6.20. Scratchpad.....	53
6.21. Nop.....	54
6.22. control i3bar.....	54
7. Múltiples monitores.....	55
7.1. Configurando sus monitores.....	55
7.2. Configuración interesante para entornos de múltiples monitores.....	57
8. i3 y el resto de su mundo de software.....	57
8.1. Mostrar una línea de estado.....	57
8.2. Dar presentaciones (multimonitor).....	58
1. 8.2.1. Caso 1: todos obtienen la misma salida.....	58

2. 8.2.2. Caso 2: puedes ver más que tu audiencia.....	58
8.3. Pantallas de alta resolución (también conocidas como pantallas HIDPI).....	58
9. Configuración por defecto.....	59
10. La situación de múltiples monitores.....	63
10.1. La solución rápida.....	63
10.2. La explicación.....	64
11. Ver también.....	64

i3 Guía del Usuario

Michael Stapelberg
<michael@i3wm.org>

Este documento contiene toda la información que necesita para configurar y usar el administrador de ventanas i3. Si no es así, primero verifique reddit.com/r/i3wm/, luego contáctenos en IRC (preferido) o publique sus preguntas en la lista de correo.

1. Asignaciones de teclas predeterminadas

Para las personas "demasiado largo; no leí", aquí hay una descripción general de las combinaciones de teclas predeterminadas:

Claves para usar con **\$mod** (**Alt**):

~	!	@	#	\$	%	^	&	*	()	-	+	←	
`	1	2	3	4	5	6	7	8	9	0	_	=	Backspace	
Tab	↔	Q	tabbed layout	default layout	resize mode	T	Y	U	I	O	P	{	}	
↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔
Caps Lock	focus parent	stacked layout	dmenu	full screen	G	split horiz.	left	down	up	right	"	,	open terminal	
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
Shift	Z	X	C	split vert.	B	N	M	<	>	?	/	↕	Shift	
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
Ctrl	Win Key	Mod1	focus floating/tiling							Alt	Win Key	Menu	Ctrl	

Claves para usar con **Shift+\$mod**:

~	!	@	#	\$	%	^	&	*	()	-	+	←	
`	1	2	3	4	5	6	7	8	9	0	_	=	Backspace	
Tab	kill window	W	exit i3	restart i3	T	Y	U	I	O	P	{	}		
↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔
Caps Lock	A	S	D	F	G	H	move left	move down	move up	move right	"	,	Enter	
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
Shift	Z	X	C	V	B	N	M	<	>	?	/	↕	Shift	
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
Ctrl	Win Key	Mod1	toggle tiling/floating							Alt	Win Key	Menu	Ctrl	

Las **teclas rojas** son los modificadores que necesita presionar (por defecto), las **teclas azules** son las teclas que pertenecen a la línea central del teclado.

Tenga en cuenta que al iniciar i3 sin un archivo de configuración, **i3-config-wizard** le ofrecerá crear un archivo de configuración en el que las posiciones de las teclas coincidan con lo que ve en la imagen de arriba,

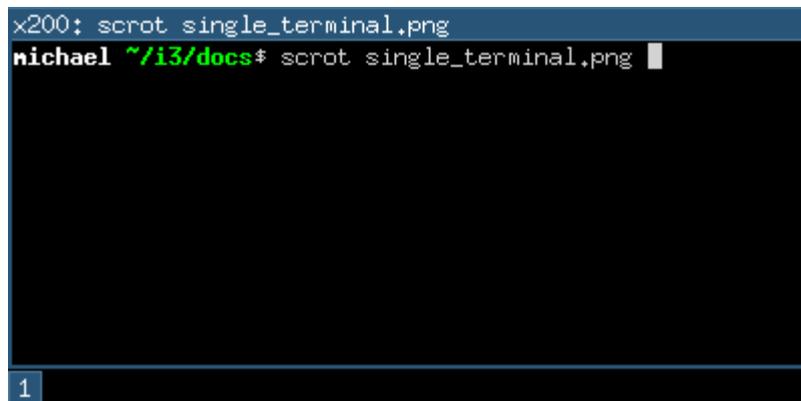
independientemente de la distribución del teclado que esté utilizando. Si prefiere usar un archivo de configuración donde las letras clave coinciden con lo que está viendo arriba, simplemente rechace el ofrecimiento de `i3-config-wizard` y base su configuración en `/etc/i3/config`.

2. Usando i3

A lo largo de esta guía, la palabra clave `$mod` se usará para referirse al modificador configurado. Esta es la tecla `Alt` (`Mod1`) de forma predeterminada, siendo la tecla `Windows` (`Mod4`) una alternativa popular que evita en gran medida los conflictos con los accesos directos definidos por la aplicación.

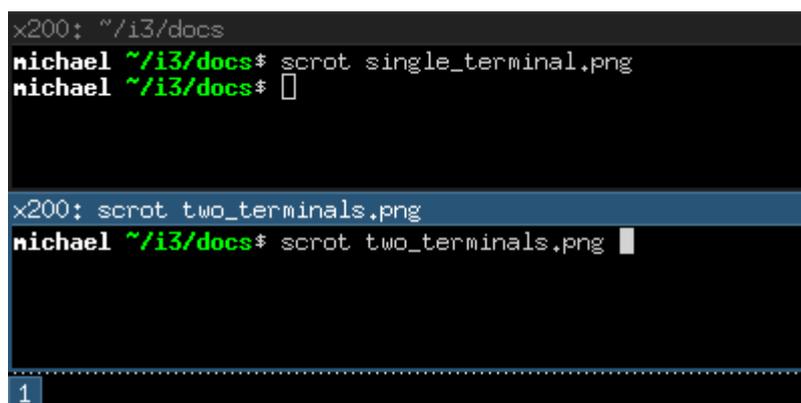
2.1. Abrir terminales y moverse

Una operación muy básica es abrir una nueva terminal. Por defecto, la combinación de teclas para esto es `$mod+Enter`, es decir `Alt+Enter` (`Mod1+Enter`) en la configuración predeterminada. Al presionar `$mod+Enter`, se abrirá una nueva terminal. Ocupará todo el espacio disponible en su pantalla.



```
x200: scrot single_terminal.png
michael ~/i3/docs$ scrot single_terminal.png
```

Si ahora abre otro terminal, `i3` lo colocará al lado del actual, dividiendo el tamaño de la pantalla a la mitad. Dependiendo de su monitor, `i3` colocará la ventana creada al lado de la ventana existente (en pantallas anchas) o debajo de la ventana existente (pantallas giradas).



```
x200: ~/i3/docs
michael ~/i3/docs$ scrot single_terminal.png
michael ~/i3/docs$

x200: scrot two_terminals.png
michael ~/i3/docs$ scrot two_terminals.png
```

Para mover el foco entre los dos terminales, puede usar las teclas de dirección que puede conocer del editor `vi`. Sin embargo, en `i3`, su línea de teclas central (la línea de teclas que se encuentra en el medio de el teclado) se usa para estas teclas (en `vi`, las teclas se desplazan hacia la izquierda por una para compatibilidad con la mayoría de los diseños de teclado). Por lo tanto, `$mod+j`: mueve el foco hacia la izquierda, `$mod+k`: mueve

el foco hacia la derecha, `$mod+l` : mueve el foco hacia arriba y `$mod+;` lo mueve hacia abajo. Entonces, para cambiar entre las terminales, use `$mod+k` o `$mod+l` . Por supuesto, también puede usar las teclas de flecha.

Por el momento, su espacio de trabajo está dividido (contiene dos terminales) en una dirección específica (horizontal por defecto). Cada ventana se puede dividir horizontal o verticalmente de nuevo, al igual que el espacio de trabajo. La terminología es "ventana" para un contenedor que realmente contiene una ventana [X11](#) (como un terminal o navegador) y "contenedor dividido" para contenedores que consisten en una o más ventanas.

TODO: foto del árbol

Para dividir una ventana verticalmente, presione `$mod+v` antes de crear la nueva ventana. Para dividirlo horizontalmente, presione `$mod+h` .

2.2. Cambiar el diseño del contenedor

Un contenedor dividido puede tener uno de los siguientes diseños:

splith/splitv

Las ventanas están dimensionadas para que cada ventana tenga la misma cantidad de espacio en el contenedor. `splith` distribuye las ventanas horizontalmente (las ventanas están una al lado de la otra), `splitv` las distribuye verticalmente (las ventanas están una encima de la otra).

stacking (apilado)

Solo se muestra la ventana enfocada en el contenedor. Obtiene una lista de ventanas en la parte superior del contenedor.

tabbed (con pestañas)

El mismo principio que el apilamiento, pero la lista de ventanas en la parte superior es solo una línea que se divide verticalmente.

Para cambiar de modo, presione `$mod+e` para `splith/splitv` (se alterna), `$mod+s` para apilar y `$mod+w` para pestañas.

```

x200: ~/i3
michael ~/i3$

x200: ~/i3
michael ~/i3$

x200: /tmp
michael /tmp$

x200: ~/i3/docs
michael ~/i3/docs$ scrot mo
des.png
  
```

2.3. Alternar el modo de pantalla completa para una ventana

Para mostrar una ventana en modo de pantalla completa o para salir nuevamente del modo de pantalla completa, presione `$mod+f`.

También hay un modo de pantalla completa global en `i3` en el que el cliente abarcará todas las salidas disponibles (el comando es `fullscreen toggle global`).

2.4. Abrir otras aplicaciones

Además de abrir aplicaciones desde una terminal, también puede usar el práctico menú que se abre presionando `$mod+d` por defecto. Simplemente escriba el nombre (o una parte de él) de la aplicación que desea abrir. La aplicación correspondiente debe estar en su `$PATH` para que esto funcione.

Además, si tiene aplicaciones que abre con mucha frecuencia, puede crear una combinación de teclas para iniciar la aplicación directamente. Consulte la sección [Configurando] `i3` para más detalles.

2.5. Cerrar ventanas

Si una aplicación no proporciona un mecanismo para cerrar (la mayoría de las aplicaciones proporcionan un menú, la tecla de escape o un acceso directo como `Control+w` para cerrar), puede presionar `$mod+Shift+q` para cerrar una ventana. Para las aplicaciones que admiten el protocolo `WM_DELETE`, esto cerrará correctamente la aplicación (guardando cualquier modificación o haciendo otra limpieza). Si la aplicación no admite el protocolo `WM_DELETE`, su servidor `X` cerrará la ventana y el comportamiento dependerá de la aplicación.

2.6. Usando espacios de trabajo

Los espacios de trabajo son una manera fácil de agrupar un conjunto de ventanas. Por defecto, está en el primer espacio de trabajo, como lo indica la barra en la parte inferior izquierda. Para cambiar a otro espacio de trabajo, presione `$mod+num` donde `num` es el número del espacio de trabajo que desea usar. Si el espacio de trabajo aún no existe, se creará.

Un paradigma común es colocar el navegador web en un espacio de trabajo, las aplicaciones de comunicación (`mutt`, `irssi`, ...) en otro y las aplicaciones con las que trabaja, en el tercero. Por supuesto, no hay necesidad de seguir este enfoque.

Si tiene varias pantallas, se creará un espacio de trabajo en cada pantalla al inicio. Si abre un nuevo espacio de trabajo, estará vinculado a la pantalla en la que lo creó. Cuando cambie a un espacio de trabajo en otra pantalla, `i3` establecerá el foco en esa pantalla.

2.7. Mover ventanas a espacios de trabajo

Para mover una ventana a otro `workspace` (espacio de trabajo), simplemente presione `$mod+Shift+num` donde `num` es el número del espacio de trabajo de destino (como cuando se cambian espacios de trabajo). De manera similar al cambio de espacios de trabajo, el espacio de trabajo de destino se creará si aún no existe.

2.8. Redimensionando

La forma más fácil de cambiar el tamaño de un contenedor es usando el mouse: agarre el borde y muévelo al tamaño deseado.

También puede usar [Modos de enlace] para definir un modo para cambiar el tamaño a través del teclado.

Para ver un ejemplo de esto, mire la [Configuración por defecto] proporcionada por i3.

2.9. Reiniciar i3 en el lugar

Para reiniciar i3 en su lugar (y así pasar a un estado limpio si hay un error, o para actualizar a una versión más nueva de i3) puede usar `$mod+Shift+r`.

2.10. Salir de i3

Para salir limpiamente de i3 sin matar su servidor X, puede usar `$mod+Shift+e`. Por defecto, un cuadro de diálogo le pedirá que confirme si realmente desea salir.

2.11. Flotante

El modo flotante es lo opuesto al modo de mosaico. i3 no gestiona automáticamente la posición y el tamaño de una ventana, sino usted lo hace de forma manual. El uso de este modo viola el paradigma de mosaico, pero puede ser útil para algunos casos de esquina como ventanas de diálogo "Guardar como" o ventanas de barra de herramientas (GIMP o similar). Esas ventanas generalmente establecen la sugerencia apropiada y se abren en modo flotante de forma predeterminada.

Puede alternar el modo flotante para una ventana presionando `$mod+Shift+Space`. Al arrastrar la barra de título de la ventana con el mouse, puede mover la ventana. Al agarrar los bordes y moverlos, puede cambiar el tamaño de la ventana. También puede hacerlo usando [El modificador flotante]. Otra forma de cambiar el tamaño de las ventanas flotantes con el mouse es hacer clic derecho en la barra de título y arrastrar.

Para cambiar el tamaño de las ventanas flotantes con su teclado, consulte el modo de enlace de cambio de tamaño proporcionado por la [Configuración por defecto] de i3.

Las ventanas flotantes siempre están encima de las ventanas de mosaico.

3. Árbol

i3 almacena toda la información sobre las salidas X11, los espacios de trabajo y el diseño de las ventanas en un árbol. El nodo raíz es la ventana raíz X11, seguido de las salidas X11, luego las áreas de acoplamiento y un contenedor de contenido, luego espacios de trabajo y finalmente las ventanas mismas. En versiones anteriores de i3 teníamos múltiples listas (de salidas, espacios de trabajo) y una tabla para cada espacio de trabajo. Ese enfoque resultó ser complicado de usar (ajustar), comprender e implementar.

3.1. El árbol consta de contenedores.

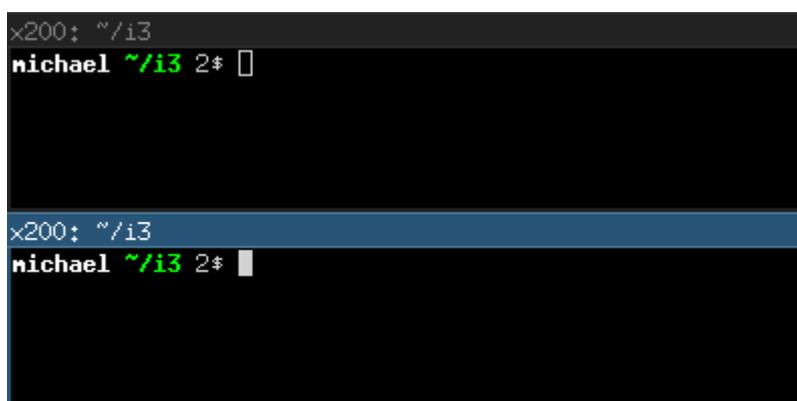
Los bloques de construcción de nuestro árbol son los llamados contenedores. Un contenedor puede alojar una ventana (es decir, una ventana `X11`, una que realmente puede ver y usar, como un navegador). Alternativamente, podría contener uno o más Contenedores. Un ejemplo simple es el espacio de trabajo: cuando inicia `i3` con un solo monitor, un solo espacio de trabajo y abre dos ventanas de terminal, terminará con un árbol como este:



Dos terminales en el espacio de trabajo estándar

3.2. Orientación y contenedores divididos

Es natural usar los llamados `split containers` (contenedores divididos) para construir un diseño cuando se usa un árbol como estructura de datos. En `i3`, cada contenedor tiene una orientación (horizontal, vertical o no especificada) y la orientación depende del diseño en el que se encuentra el contenedor (`splitv` para dividir y apilar, `splith` para horizontal y con pestañas). Entonces, en nuestro ejemplo con el espacio de trabajo, el diseño predeterminado del contenedor del espacio de trabajo es dividido `splith` o sea, en forma horizontal (, (la mayoría de los monitores son de pantalla panorámica hoy en día). Si cambia el diseño a `splitv` (`$mod+v` en la configuración predeterminada) y luego abre dos terminales, `i3` configurará sus ventanas de esta manera:



Orientación vertical de el espacio de trabajo.

Una nueva característica interesante de i3 desde la versión 4 es la capacidad de dividir cualquier cosa: supongamos que tiene dos terminales en un espacio de trabajo (con diseño dividido, que es orientación horizontal), el foco está en el terminal derecho. Ahora desea abrir otra ventana de terminal debajo de la actual. Si solo abriera una nueva ventana de terminal, se mostraría a la derecha debido al diseño horizontal (`splith`). En su lugar, presione `$mod+v` para dividir el contenedor con el diseño de `splittv` (para abrir un Contenedor de división horizontal, use `$mod+h`). Ahora puede abrir una nueva terminal y se abrirá debajo de la actual:

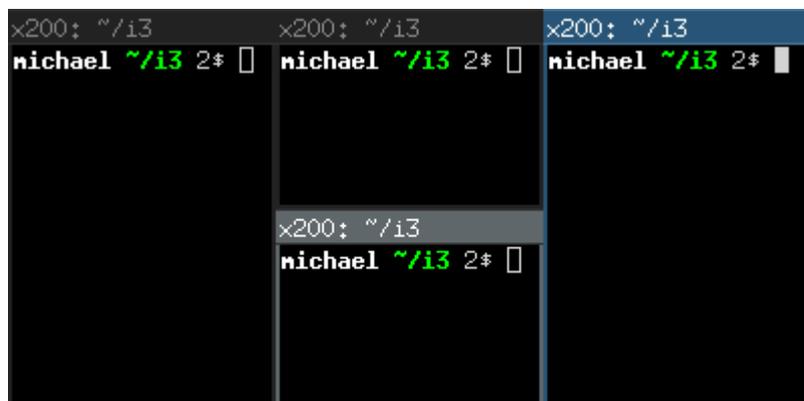


Contenedor vertical dividido

3.3. Padre de enfoque

Sigamos con nuestro ejemplo de arriba. Tenemos un terminal a la izquierda y dos terminales divididos verticalmente a la derecha, el foco está en el inferior derecho. Cuando abre una nueva terminal, se abrirá debajo de la actual.

Entonces, ¿cómo puede abrir una nueva ventana de terminal a la derecha de la actual? La solución es usar *focus parent*, que enfocará el contenedor principal del contenedor actual. En la configuración predeterminada, use `$mod+a` para navegar un Contenedor hacia arriba en el árbol (puede repetir esto varias veces hasta llegar al Contenedor del área de trabajo). En este caso, enfocaría el Contenedor de división vertical que está dentro del espacio de trabajo orientado horizontalmente. Por lo tanto, ahora se abrirán nuevas ventanas a la derecha del Contenedor de división vertical:



Enfoque padre, luego nueva terminal abierta.

3.4. Contenedores implícitos

En algunos casos, **i3** necesita crear implícitamente un contenedor para cumplir con su comando.

Un ejemplo es el siguiente escenario: inicia **i3** con un solo monitor y un único espacio de trabajo en el que abre tres ventanas de terminal. Todas estas ventanas de terminal están conectadas directamente a un nodo dentro del árbol de diseño de **i3**, el nodo del espacio de trabajo. De forma predeterminada, la orientación del nodo del espacio de trabajo es horizontal.

Ahora mueve uno de estos terminales hacia abajo (`$mod+Shift+k` por defecto). La orientación del nodo del espacio de trabajo se cambiará a vertical. La ventana de terminal que movió hacia abajo está directamente conectada al espacio de trabajo y aparece en la parte inferior de la pantalla. Se creó un nuevo contenedor (horizontal) para acomodar las otras dos ventanas de terminal. Notará esto cuando cambie al modo de pestañas (por ejemplo). Terminaría teniendo una pestaña con una representación del contenedor dividido (por ejemplo, "H [urxvt firefox]") y la otra es la ventana de terminal que movió hacia abajo.

4. Configurando i3

Aquí es donde comienza la verdadera diversión. La mayoría de las cosas dependen mucho de su entorno de trabajo ideal, por lo que no podemos establecer valores predeterminados razonables para ellos.

Si bien no utiliza un lenguaje de programación para la configuración, **i3** se mantiene bastante flexible con respecto a las cosas que generalmente desea que haga su administrador de ventanas.

Por ejemplo, puede configurar enlaces para saltar a ventanas específicas, puede configurar aplicaciones específicas para que comiencen en espacios de trabajo específicos, puede iniciar aplicaciones automáticamente, puede cambiar los colores de **i3** y puede vincular sus teclas para hacer cosas útiles.

Para cambiar la configuración de **i3**, copie `/etc/i3/config` a `~/.i3/config` (o `~/.config/i3/config` si le gusta el esquema de directorio `XDG`) y edítelo con un editor de texto.

En el primer inicio (y en todos los siguientes inicios, a menos que tenga un archivo de configuración), **i3** le ofrecerá crear un archivo de configuración. Puede decirle al asistente que use tecla `Alt (Mod1)` o tecla `Windows (Mod4)` como modificador en el archivo de configuración. Además, el archivo de configuración creado usará los símbolos clave de la distribución actual del teclado. Para iniciar el asistente, use el comando `i3-config-wizard`. Tenga en cuenta que no debe tener `~/.i3/config`; de lo contrario, el asistente se cerrará.

Desde **i3 v4.0**, se utiliza un nuevo formato de configuración. **i3** intentará detectar automáticamente la versión de formato de un archivo de configuración en función de algunas palabras clave diferentes, pero si desea asegurarse de que su configuración se lea con el nuevo formato, incluya la siguiente línea en su archivo de configuración:

```
# i3 config file (v4)
```

4.1. Comentarios

Es posible y recomendado utilizar comentarios en su archivo de configuración para documentar adecuadamente su configuración para referencia posterior. Los comentarios comienzan con un `#` y solo se pueden usar al comienzo de una línea:

```
# This is a comment
```

4.2. Fuentes

i3 tiene soporte para fuentes `X core` y `FreeType` (a través de `Pango`) para renderizar títulos de ventanas.

Para generar una descripción de fuente `X core`, puede usar `xfontsel (1)`. Para ver caracteres especiales (`Unicode`), debe usar una fuente que admita la codificación `ISO-10646`.

Una descripción de fuente `FreeType` está compuesta por una familia de fuentes, un estilo, un peso, una variante, un tramo y un tamaño. Las fuentes `FreeType` admiten el renderizado de derecha a izquierda y a menudo contienen más glifos `Unicode` que las fuentes `X core`.

Si i3 no puede abrir la fuente configurada, generará un error en el archivo de registro y volverá a una fuente que funcione.

Sintaxis:

```
font <X core font description>
font pango:<family list> [<style options>] <size>
```

Ejemplo:

```
font -misc-fixed-medium-r-normal--13-120-75-75-C-70-iso10646-1
font pango:DejaVu Sans Mono 10
font pango:DejaVu Sans Mono, Terminus Bold Semi-Condensed 11
font pango:Terminus 11px
```

4.3. Enlaces de teclado

Un enlace de teclado hace que i3 ejecute un comando (ver más abajo) al presionar una tecla específica. i3 le permite enlazar en códigos clave o en símbolos de teclas (también puede mezclar sus enlaces, aunque i3 no lo protegerá de la superposición).

- Un `keysym` (símbolo clave) es una descripción de un símbolo específico, como `"a"` o `"b"`, pero también otros más extraños como `"underscore" instead of "_"` ("subrayado" en lugar de `"_"`). Estos son los que usa en `Xmodmap` para reasignar sus claves. Para obtener la asignación actual de sus claves, use `xmodmap -pke`. Para ingresar una clave de manera interactiva y ver en qué `keysym` está configurado, use `xev`.
- Los códigos de tecla no necesitan tener un símbolo asignado (útil para las teclas de acceso rápido de proveedores personalizados en algunos portátiles) y no cambiarán su significado cuando cambie a una distribución de teclado diferente (cuando use `xmodmap`).

Mi recomendación es: si a menudo cambia la distribución del teclado pero desea mantener sus enlaces en la misma ubicación física en el teclado, use códigos de teclas. Si no cambia de diseño y desea un archivo de configuración limpio y simple, use `keysyms`.

Es posible que algunas herramientas (como importar o `xdotool`) no puedan ejecutarse en un evento `KeyPress`, porque el puntero/teclado todavía está ligado. Para estas situaciones, se puede usar el indicador `--release`, que ejecutará el comando una vez que se hayan liberado las teclas.

Sintáxis:

```
bindsym [--release] [<Group>+][<Modifiers>+]<keysym> command
bindcode [--release] [<Group>+][<Modifiers>+]<keycode> command
```

Ejemplo:

```
# Fullscreen
bindsym $mod+f fullscreen toggle

# Restart
bindsym $mod+Shift+r restart

# Notebook-specific hotkeys
bindcode 214 exec --no-startup-id /home/michael/toggle_beamer.sh

# Simulate ctrl+v upon pressing $mod+x
bindsym --release $mod+x exec --no-startup-id xdotool key --clearmodifiers
ctrl+v

# Take a screenshot upon pressing $mod+x (select an area)
bindsym --release $mod+x exec --no-startup-id import /tmp/latest-
screenshot.png
```

Modificadores disponibles: `Mod1-Mod5`, `Shift`, `Control`

Modificadores estándar, ver `xmodmap(1)`

`Group1`, `Group2`, `Group3`, `Group4`

Al usar múltiples diseños de teclado (por ejemplo, con `setxkbmap -layout us, ru`), puede especificar en qué grupo `XKB` (también llamado "diseño") debe estar activa una combinación de teclas. Por defecto, las combinaciones de teclas se traducen en `Group1` y están activas en todos los grupos. Si desea anular las combinaciones de teclas en uno de sus diseños, especifique el grupo correspondiente. Para compatibilidad con versiones anteriores, el grupo `"Mode_switch"` es un alias para `Group2`.

4.4. Fijaciones de mouse

Un enlace del mouse hace que `i3` ejecute un comando al presionar un botón específico del mouse en el alcance del contenedor en el que se hizo clic (ver criterios de comando). Puede configurar los enlaces del mouse de manera similar a los enlaces de teclas.

4.5. Modos de enlace

Puede tener múltiples conjuntos de enlaces utilizando diferentes modos de enlace. Cuando cambia a otro modo de enlace, todos los enlaces del modo actual se liberan y solo los enlaces definidos en el nuevo modo son válidos mientras permanezca en ese modo de enlace. El único modo de enlace predefinido es el predeterminado (`default`), que es el modo con el que comienza `i3` y al que pertenecen todos los enlaces no definidos en un modo de enlace específico.

Trabajar con modos de enlace consta de dos partes: definir un modo de enlace y cambiar a él. Para estos propósitos, hay una directiva de configuración y un comando, ambos llamados `mode`. La directiva se utiliza para definir los enlaces que pertenecen a un determinado modo de enlace, mientras que el comando cambiará al modo especificado.

Se recomienda utilizar modos de enlace en combinación con [Variables] para facilitar el mantenimiento. A continuación se muestra un ejemplo de cómo usar un modo de enlace.

Tenga en cuenta que es aconsejable definir enlaces para volver al modo predeterminado.

Tenga en cuenta que es posible usar [Pango markup] para los modos de enlace, pero debe habilitarlo explícitamente pasando el indicador `--pango_markup` a la definición del modo.

Sintaxis:

```
# config directive
mode [--pango_markup] <name>
```

```
# command
mode <name>
```

Ejemplo:

```
# Press $mod+o followed by either f, t, Escape or Return to launch firefox,
# thunderbird or return to the default mode, respectively.
set $mode_launcher Launch: [f]firefox [t]thunderbird
bindsym $mod+o mode "$mode_launcher"

mode "$mode_launcher" {
    bindsym f exec firefox
    bindsym t exec thunderbird

    bindsym Escape mode "default"
    bindsym Return mode "default"
}
```

4.6. El modificador flotante

Para mover ventanas flotantes con el mouse, puede tomar su barra de título o configurar el llamado modificador flotante que luego puede presionar y hacer clic en cualquier lugar de la ventana para moverlo. La configuración más común es usar la misma clave que usa para administrar ventanas (`Mod1`, por ejemplo). Luego puede presionar `Mod1`, hacer clic en una ventana con el botón izquierdo del mouse y arrastrarlo a la posición que desee.

Al mantener presionado el modificador flotante, puede cambiar el tamaño de una ventana flotante presionando el botón derecho del mouse sobre ella y moviéndose mientras la mantiene presionada. Si mantiene presionado el botón de desplazamiento también, el cambio de tamaño será proporcional (se conservará la relación de aspecto).

Sintaxis:

```
floating_modifier <Modifier>
```

Ejemplo:

```
floating_modifier Mod1
```

4.7. Restricción del tamaño de la ventana flotante

Se pueden especificar las dimensiones máximas y mínimas de las ventanas flotantes. Si cualquiera de las dimensiones de `floating_maximum_size` se especifica como `-1`, esa dimensión no tendrá restricciones con respecto a su valor máximo. Si cualquiera de las dimensiones de `floating_maximum_size` no está definida o se especifica como `0`, `i3` usará un valor predeterminado para restringir el tamaño máximo. `floating_minimum_size` se trata de manera análoga a `floating_maximum_size`.

Sintaxis:

```
floating_minimum_size <width> x <height>
floating_maximum_size <width> x <height>
```

Ejemplo:

```
floating_minimum_size 75 x 50
floating_maximum_size -1 x -1
```

4.8. Orientación para nuevos espacios de trabajo

Los nuevos espacios de trabajo obtienen una orientación predeterminada razonable: los monitores de pantalla ancha (cualquier cosa más ancha que alta) obtienen orientación horizontal, los monitores rotados (cualquier cosa más alta que ancha) obtienen orientación vertical.

Con la directiva de configuración `default_orientation`, puede anular ese comportamiento.

Sintaxis: `default_orientation horizontal|vertical|auto`

Ejemplo: `default_orientation vertical`

4.9. Modo de diseño para nuevos contenedores

Esta opción determina en qué modo comenzarán los nuevos contenedores en el nivel del espacio de trabajo.

Sintaxis: `workspace_layout default|stacking|tabbed`

Ejemplo: `workspace_layout tabbed`

4.10. Alineación de título de ventana

Esta opción determina la alineación del texto del título de la ventana. Por defecto es `left`.

Sintaxis: `title_align left|center|right`

4.11. Estilo de borde predeterminado para ventanas nuevas

Esta opción determina qué estilo de borde tendrán las nuevas ventanas. El valor predeterminado es normal. Tenga en cuenta que `default_floating_border` se aplica solo a las ventanas que comienzan como ventanas flotantes, por ejemplo, ventanas de diálogo, pero no ventanas que serán lanzadas como flotantes más adelante.

Establecer el estilo del borde en píxeles elimina las barras de título. El estilo de borde normal le permite ajustar el ancho del borde manteniendo su barra de título.

Sintaxis:

```
default_border normal|none|pixel
default_border normal|pixel <px>
default_floating_border normal|none|pixel
default_floating_border normal|pixel <px>
```

Tenga en cuenta que `new_window` y `new_float` han quedado en desuso en favor de las opciones anteriores y se eliminarán en una versión futura. En su lugar, recomendamos utilizar las nuevas opciones.

Ejemplo: `default_border pixel`

Los estilos de borde "normal" y "píxel" admiten un ancho de borde opcional en píxeles:

Ejemplo:

```
# The same as default_border none
default_border pixel 0

# A 3 px border
default_border pixel 3
```

4.12. Ocultar bordes adyacentes a los bordes de la pantalla

Puede ocultar los bordes del contenedor adyacentes a los bordes de la pantalla usando `hide_edge_borders`. Esto es útil si está utilizando barras de desplazamiento o no desea desperdiciar ni siquiera dos píxeles en el espacio de visualización. La configuración "smart" (inteligente) oculta los bordes de los espacios de trabajo con una sola ventana visible, pero los mantiene en los espacios de trabajo con múltiples ventanas visibles. El valor predeterminado es `none` (ninguno).

Sintaxis: `hide_edge_borders none|vertical|horizontal|both|smart`

Ejemplo: `hide_edge_borders vertical`

4.13. Comandos arbitrarios para ventanas específicas (`for_window`)

Con la directiva `for_window`, puede permitir que `i3` ejecute cualquier comando cuando encuentre una ventana específica. Esto se puede usar para configurar ventanas como flotantes o para cambiar su estilo de borde, por ejemplo.

Sintaxis: `for_window <criteria> <command>`

Ejemplo:

```
# enable floating mode for all XTerm windows
for_window [class="XTerm"] floating enable

# Make all urxvts use a 1-pixel border:
for_window [class="urxvt"] border pixel 1

# A less useful, but rather funny example:
# makes the window floating as soon as I change
# directory to ~/work
for_window [title="x200: ~/work"] floating enable
```

Los criterios válidos son los mismos que para los comandos, consulte [criterios de comando]. Solo las directivas de configuración con un comando equivalente se pueden ejecutar en tiempo de ejecución, consulte [Lista de comandos].

4.14. No enfoque la ventana al abrir

Cuando aparece una nueva ventana, se enfocará. La directiva `no_focus` permite evitar que esto suceda y debe usarse en combinación con [criterios de comando].

Tenga en cuenta que esto no se aplica a todos los casos, por ejemplo, cuando se alimentan datos en una aplicación en ejecución, lo que hace es solicitar que se centren. Para configurar el comportamiento en tales casos, consulte [Centrarse en la activación de la ventana].

`no_focus` también se ignorará para la primera ventana en un espacio de trabajo, ya que no debería haber una razón para no enfocar la ventana en este caso. Esto permite una mejor usabilidad en combinación con `workspace_layout`.

Sintaxis: `no_focus <criteria>`

Ejemplo: `no_focus [window_role="pop-up"]`

4.15. Variables

Como aprendió en la sección sobre enlaces de teclado, tendrá que configurar muchos enlaces que contengan teclas modificadoras. Si desea ahorrarse algo de escritura y poder cambiar el modificador que usa más adelante, las variables pueden ser útiles.

Sintaxis: `set $<name> <value>`

Ejemplo:

```
set $m Mod1
bindsym $m+Shift+r restart
```

Las variables se reemplazan directamente en el archivo al analizar. La expansión de variables no es recursiva, por lo que no es posible definir una variable con un valor que contenga otra variable. No hay un manejo sofisticado y no hay absolutamente ningún plan para cambiar esto. Si necesita una configuración más dinámica, debe crear un pequeño script que genere un archivo de configuración y ejecutarlo antes de iniciar i3 (por ejemplo, en su archivo `~/.xsession`).

También vea [X recursos] para aprender cómo crear variables basadas en los recursos cargados desde la base de datos de recursos X.

4.16. X recursos

[Variables] también se pueden crear utilizando un valor configurado en la base de datos de recursos X. Esto es útil, por ejemplo, para evitar configurar valores de color dentro de la configuración de i3. En cambio, los valores se pueden configurar, una vez, en la base de datos de recursos X para lograr un tema de color coherente y fácil de mantener en muchas aplicaciones X.

La definición de un recurso cargará este recurso desde la base de datos de recursos y asignará su valor a la variable especificada. Esto se hace literalmente y, por lo tanto, el valor debe estar en el formato que utiliza i3. Se debe especificar una reserva en caso de que el recurso no se pueda cargar desde la base de datos.

Sintaxis:

```
set_from_resource $<name> <resource_name> <fallback>
```

Ejemplo:

```
# The ~/.Xresources should contain a line such as
# *color0: #121212
# and must be loaded properly, e.g., by using
# xrdp ~/.Xresources
# This value is picked up on by other applications (e.g., the URxvt terminal
# emulator) and can be used in i3 like this:
set_from_resource $black i3wm.color0 #000000
```

4.17. Poner automáticamente a los clientes en espacios de trabajo específicos

Para que aparezca automáticamente una ventana específica en un espacio de trabajo específico, puede usar una asignación (`assignment`). Puede hacer coincidir ventanas utilizando cualquier criterio, consulte [criterios de comando]. La diferencia entre `assign` y `for_window <criteria> move to workspace` es que el primero solo se ejecutará cuando la aplicación mapee la ventana (el mapeo significa realmente mostrarlo en la pantalla) pero el segundo se ejecutará cada vez que una ventana cambie sus propiedades a algo que coincide con los criterios especificados.

Por lo tanto, se recomienda que coincida con las clases de ventana (y las instancias, cuando corresponda) en lugar de los títulos de las ventanas siempre que sea posible porque algunas aplicaciones primero crean su ventana y luego se preocupan por establecer el título correcto. Me viene a la mente Firefox con Vimperator. La ventana comienza a llamarse Firefox, y solo cuando se carga Vimperator cambia el título. Como `i3` obtendrá el título tan pronto como la aplicación mapee la ventana, en este caso necesitaría tener que coincidir con Firefox. Otro problema conocido es con Spotify, que no establece las sugerencias de clase al asignar la ventana, lo que significa que tendrá que usar una regla `for_window` para asignar Spotify a un espacio de trabajo específico. Finalmente, el uso de `assing [tiling]` y `assign [floating]` no es compatible.

También puede asignar una ventana para que aparezca en una salida específica. Puede usar nombres `RandR` como `VGA1` o nombres relativos a la salida con el espacio de trabajo enfocado actualmente como izquierda y abajo.

Las asignaciones son procesadas por `i3` en el orden en que aparecen en el archivo de configuración. Gana el primero que coincida con la ventana y las tareas posteriores no se consideran.

Sintáxis:

```
assign <criteria> [-] [workspace] [number] <workspace>
assign <criteria> [-] output left|right|up|down|primary|<output>
```

Ejemplo:

```
# Assign URxvt terminals to workspace 2
assign [class="URxvt"] 2

# Same thing, but more precise (exact match instead of substring)
assign [class="^URxvt$"] 2

# Same thing, but with a beautiful arrow :)
assign [class="^URxvt$"] → 2

# Assignment to a named workspace
assign [class="^URxvt$"] → work

# Assign to the workspace with number 2, regardless of name
assign [class="^URxvt$"] → number 2
```

```
# You can also specify a number + name. If the workspace with number 2 exists,
# assign will skip the text part.
assign [class="^URxvt$"] → number "2: work"

# Start urxvt -name irssi
assign [class="^URxvt$" instance="^irssi$"] → 3

# Assign urxvt to the output right of the current one
assign [class="^URxvt$"] → output right

# Assign urxvt to the primary output
assign [class="^URxvt$"] → output primary
```

Tenga en cuenta que es posible que todavía no tenga una salida principal configurada. Para hacerlo, ejecuta:

```
xrandr --output <output> --primary
```

Además, la flecha no es necesaria, simplemente se ve bien. Si decide usarlo, debe ser una flecha codificada en UTF-8, no → o algo así.

Para obtener la clase y la instancia, puede usar `xprop`. Después de hacer clic en la ventana, verá el siguiente resultado:

```
xprop
WM_CLASS(STRING) = "irssi", "URxvt"
```

La primera parte de `WM_CLASS` es la instancia ("irssi" en este ejemplo), la segunda parte es la clase ("URxvt" en este ejemplo).

Si tiene algún problema con las tareas, asegúrese de verificar primero el archivo de registro `i3` (consulte <https://i3wm.org/docs/debugging.html>). Incluye más detalles sobre el proceso de coincidencia y la clase, la instancia y el título reales de la ventana al iniciar.

Tenga en cuenta que si desea iniciar una aplicación solo una vez en un espacio de trabajo específico, pero no desea asignar todas las instancias de forma permanente, puede utilizar el soporte de notificación de inicio de `i3` (consulte `[exec]`) en su archivo de configuración de la siguiente manera:

Inicio iceweasel en workspace 3 (una vez):

```
# Start iceweasel on workspace 3, then switch back to workspace 1
# (Being a command-line utility, i3-msg does not support startup notifications,
# hence the exec --no-startup-id.)
# (Starting iceweasel with i3's exec command is important in order to make i3
# create a startup notification context, without which the iceweasel window(s)
# cannot be matched onto the workspace on which the command was started.)
exec --no-startup-id i3-msg 'workspace 3; exec iceweasel; workspace 1'
```

4.18. Inicio automático de aplicaciones al iniciar i3

Al usar la palabra clave `exec` fuera de una combinación de teclas, puede configurar qué comandos ejecutará `i3` en al iniciar. Los comandos `exec` no se ejecutarán al reiniciar `i3`, si necesita un comando para ejecutar también al reiniciar `i3`, debe usar la palabra clave `exec_always`. Estos comandos se ejecutarán en orden.

Ver [encadenar] para detalles sobre el significado especial de `;` (punto y coma) y `,` (coma): encadenan los comandos en `i3`, por lo que debe usar cadenas entre comillas (como se muestra en [comando a ejecutar]) si aparecen en su comando.

Sintáxis:

```
exec [--no-startup-id] <command>
exec_always [--no-startup-id] <command>
```

Ejemplo:

```
exec chromium
exec_always ~/my_script.sh

# Execute the terminal emulator urxvt, which is not yet startup-notification aware.
exec --no-startup-id urxvt
```

La bandera `--no-startup-id` está explicado en [exec].

4.19. Poner automáticamente espacios de trabajo en pantallas específicas

Si asigna clientes a espacios de trabajo, puede ser útil colocar los espacios de trabajo en pantallas específicas. Además, la asignación de espacios de trabajo a las pantallas determinará qué espacio de trabajo `i3` usa para una nueva pantalla al agregar pantallas o al iniciar (por ejemplo, por defecto usará 1 para la primera pantalla, 2 para la segunda pantalla, etc.).

Sintáxis:

```
workspace <workspace> output <output1> [output2]...
```

La salida es el nombre de la salida `RandR` a la que adjunta su pantalla. En una computadora portátil, puede tener `VGA1` y `LVDS1` como nombres de salida. Puede ver las salidas disponibles ejecutando `xrandr --current`.

Si su servidor `X` es compatible con `RandR 1.5` o posterior, `i3` usará objetos de monitor `RandR` en lugar de objetos de salida. Ejecute `xrandr --listmonitors` para ver una lista. Por lo general, un objeto de monitor contiene exactamente una salida y tiene el mismo nombre que esta; pero si ese no fuera el caso, puede especificar el nombre del monitor o la salida en la configuración de `i3`. Por ejemplo, el `Dell UP2414Q` usa dos escaladores internamente, por lo que sus nombres de salida pueden ser `"DP1"` y `"DP2"`, pero el nombre del monitor es `"Dell UP2414Q"`.

(Tenga en cuenta que incluso si especifica el nombre de una salida que no abarca todo el monitor, `i3` seguirá utilizando toda el área del monitor que lo contiene en lugar de solo el de la salida).

Puede especificar múltiples salidas. Se utilizará el primero disponible.

Si usa espacios de trabajo con nombre, deben citarse:

Ejemplo:

```
workspace 1 output LVDS1
workspace 2 output primary
workspace 5 output VGA1 LVDS1
workspace "2: vim" output VGA1
```

4.20. Cambio de colores

Puede cambiar todos los colores que `i3` usa para dibujar las decoraciones de las ventanas.

Sintaxis:

```
<colorclass> <border> <background> <text> <indicator> <child_border>
```

Donde `colorclass` puede ser uno de:

`client.focused` Un cliente que actualmente tiene el foco.

`client.focused_inactive` Un cliente que es el enfocado de su contenedor, pero no tiene el foco en este momento.

`client.unfocused` Un cliente que no es el enfocado de su contenedor.

`client.urgent` Un cliente que tiene activada su sugerencia de urgencia.

`client.placeholder` El color de fondo y de texto se utiliza para dibujar el contenido de la ventana del marcador de posición (al restaurar diseños). El borde y el indicador se ignoran.

`client.background` Color de fondo que se utilizará para pintar el fondo de la ventana del cliente sobre el cual se representará el cliente. Solo los clientes que no cubren toda el área de esta ventana exponen el color. Tenga en cuenta que esta clase de color solo toma un solo color.

Los colores están en formato hexadecimal HTML (`#rrggbb`), consulte el siguiente ejemplo:

Ejemplos (colores predeterminados):

```
# class          border  backgr. text  indicator  child_border
client.focused   #4c7899 #285577 #ffffff #2e9ef4   #285577
client.focused_inactive #333333 #5f676a #ffffff #484e50   #5f676a
client.unfocused #333333 #222222 #888888 #292d2e   #222222
client.urgent    #2f343a #900000 #ffffff #900000   #900000
client.placeholder #000000 #0c0c0c #ffffff #000000   #0c0c0c

client.background #ffffff
```

Tenga en cuenta que para las decoraciones de la ventana, el color alrededor de la ventana secundaria es el "child_border", y el color "border" es solo las dos líneas finas alrededor de la barra de título.

El color del indicador se utiliza para indicar dónde se abrirá una nueva ventana. Para contenedores divididos horizontales, el borde derecho se pintará en color indicador, para contenedores divididos verticales, el borde inferior. Esto solo se aplica a ventanas individuales dentro de un contenedor dividido, porque de otra manera no se pueden distinguir de ventanas individuales fuera de un contenedor dividido.

4.21. Comunicación entre procesos

i3 usa sockets Unix para proporcionar una interfaz IPC. Esto permite que los programas de terceros obtengan información de i3, como los espacios de trabajo actuales (para mostrar una barra de espacio de trabajo), y para controlar i3.

El socket IPC está habilitado de manera predeterminada y se creará en `/tmp/i3-%u.XXXXXX/ipc-socket.%p`, donde %u es su nombre de usuario UNIX, %p es el PID de i3 y XXXXXX es una cadena aleatoria del conjunto de caracteres portátil del nombre de archivo (consulte `mkdtemp (3)`).

Puede anular la ruta predeterminada a través de la variable de entorno `I3SOCK` o especificando la directiva `ipc-socket`. Sin embargo, esto se desaconseja, ya que i3 hace lo correcto de manera predeterminada. Si decide cambiarlo, se recomienda encarecidamente configurarlo en una ubicación en su directorio de inicio para que ningún otro usuario pueda crear ese directorio.

Ejemplo:

Luego puede usar la aplicación `i3-msg` para ejecutar cualquier comando que se enumere en [Lista de comandos].

4.22. El foco sigue al mouse

Por defecto, el foco de la ventana sigue los movimientos del mouse cuando el mouse cruza los bordes de la ventana. Sin embargo, si tiene una configuración donde su mouse generalmente está en su camino (como un panel táctil en su computadora portátil que no desea deshabilitar por completo), es posible que desee deshabilitar el enfoque después del mouse y controlar el enfoque solo usando su teclado. El mouse seguirá siendo útil dentro de la ventana activa actualmente (por ejemplo, para hacer clic en los enlaces de la ventana de su navegador).

Sintaxis: `focus_follows_mouse yes|no`

Ejemplo: `focus_follows_mouse no`

4.23. Deformación del ratón

De manera predeterminada, al cambiar el foco a una ventana en una salida diferente (por ejemplo, enfocar una ventana en el espacio de trabajo 3 en la salida `VGA-1`, proveniente del espacio de trabajo 2 en `LVDS-1`), el cursor del mouse se deforma al centro de esa ventana.

Con la opción `mouse_warping`, puede controlar cuándo se debe deformar el cursor del mouse. `none` deshabilita la deformación por completo, mientras que la salida es el comportamiento predeterminado descrito anteriormente.

Sintaxis: `mouse_warping output|none`

Ejemplo: `mouse_warping none`

4.24. Ventanas emergentes (popups) durante el modo de pantalla completa

Cuando está en modo de pantalla completa, algunas aplicaciones aún abren ventanas emergentes (tomamos a `Xpdf` por ejemplo). Esto se debe a que estas aplicaciones pueden no ser conscientes de que están en modo de pantalla completa (no verifican la pista correspondiente). Hay tres cosas que se pueden hacer en esta situación:

1. Muestra la ventana emergente si pertenece solo a la aplicación de pantalla completa. Este es el valor predeterminado y debe ser un comportamiento razonable para la mayoría de los usuarios.
2. Simplemente ignore la ventana emergente (no la asigne). Esto no te interrumpirá mientras estés en pantalla completa. Sin embargo, algunas aplicaciones pueden reaccionar mal ante esto (punto muerto hasta que salga de la pantalla completa).
3. Salir del modo de pantalla completa.

Sintaxis: `popup_during_fullscreen smart|ignore|leave_fullscreen`

Ejemplo: `popup_during_fullscreen smart`

4.25. Ajuste de enfoque

De manera predeterminada, cuando se encuentra en un contenedor con varias ventanas o contenedores secundarios, la ventana opuesta se enfocará al intentar mover el foco sobre el borde de un contenedor (y no hay otros contenedores en esa dirección): el foco se ajusta.

Si lo desea, puede deshabilitar este comportamiento estableciendo la directiva de configuración `focus_wrapping` en el valor `no`.

Cuando está habilitado, el ajuste de foco no ocurre por defecto si hay otra ventana o contenedor en la dirección especificada, y el foco se establecerá en esa ventana o contenedor. Este es el comportamiento predeterminado para que pueda navegar a todas sus ventanas sin tener que usar el `focus parent`.

Si desea que el foco siempre se ajuste y es consciente de usar el foco principal para cambiar a diferentes contenedores, puede establecer `focus_wrapping` a el valor `force`.

Sintaxis:

`focus_wrapping yes|no|force`

`# Legacy syntax, equivalent to "focus_wrapping force"`

`force_focus_wrapping yes`

Ejemplo:

```
# Disable focus wrapping
focus_wrapping no

# Force focus wrapping
focus_wrapping force
```

4.26. Forzando Xinerama

Como se explica en profundidad en <https://i3wm.org/docs/multi-monitor.html> [múltiples monitores], algunos controladores de video X11 (especialmente el controlador binario nVidia) solo brindan soporte para Xinerama en lugar de RandR. En tal situación, se le debe decir a i3 que use explícitamente la API Xinerama inferior y, por lo tanto, no brinde soporte para reconfigurar sus pantallas sobre la marcha (se leen solo una vez al inicio y eso es todo).

Para las personas que no pueden modificar su `~/.xsession` para agregar el parámetro de línea de comandos `--force-xinerama`, se proporciona una opción de configuración:

Sintaxis: `force_xinerama yes|no`

Ejemplo: `force_xinerama yes`

También tenga en cuenta que sus nombres de salida no son descriptivos (como HDMI1) cuando usa Xinerama, sino que se cuentan, comenzando en 0: `xinerama-0`, `xinerama-1`, ...

4.27. Avance y retroceso automático al cambiar al espacio de trabajo actual

Esta directiva de configuración habilita el `workspace back_and_forth` (ver [volver al espacio de trabajo previamente enfocado]) al cambiar al espacio de trabajo que está enfocado actualmente.

Por ejemplo: suponga que está en el espacio de trabajo "1: www" y cambie a "2:IM" con `$mod+2` porque alguien le envió un mensaje. No necesita recordar de dónde vino ahora, solo puede presionar `$mod+2` nuevamente para volver a "1:www".

Sintaxis: `workspace_auto_back_and_forth yes|no`

Ejemplo: `workspace_auto_back_and_forth yes`

4.28. Retrasar el restablecimiento de la sugerencia de urgencia en el cambio del espacio de trabajo

Si una aplicación en otro espacio de trabajo establece una sugerencia de urgencia, cambiar a este espacio de trabajo puede conducir a un enfoque inmediato de la aplicación, lo que también significa que el color de la decoración de la ventana se restablecerá inmediatamente a `client.focused`. Esto puede hacer que sea innecesariamente difícil saber qué ventana originó originalmente el evento.

Para evitar esto, puede decirle a i3 que retrase el restablecimiento del estado de urgencia en un momento determinado utilizando la directiva `force_display_urgency_hint`. Establecer el valor en `0` deshabilita esta función.

El valor predeterminado es `500 ms`.

Sintaxis:

```
force_display_urgency_hint <timeout> ms
```

Ejemplo:

```
force_display_urgency_hint 500 ms
```

4.29. Centrarse en la activación de la ventana

Si se activa una ventana, por ejemplo, a través de `google-chrome www.google.com`, puede solicitar que se enfoque. Como esto puede no ser preferible, se pueden configurar diferentes reacciones.

Tenga en cuenta que esto puede no afectar a las ventanas que se están abriendo. Para evitar que se enfoquen nuevas ventanas, consulte [No enfoque la ventana al abrir].

Sintaxis: `focus_on_window_activation smart|urgent|focus|none`

Los diferentes modos actuarán de la siguiente manera:

smart Este es el comportamiento predeterminado. Si la ventana que solicita el foco está en un espacio de trabajo activo, recibirá el foco. De lo contrario, se establecerá la sugerencia de urgencia.

urgent La ventana siempre se marcará como urgente, pero no se robará el foco.

focus La ventana siempre estará enfocada y no se marcará como urgente.

none La ventana no se enfocará ni se marcará como urgente.

4.30. Dibujar marcas en la decoración de ventanas

Si está activado, las marcas (ver [Marcas similares a VIM]) en las ventanas se dibujan en la decoración de sus ventanas. Sin embargo, cualquier marca que comience con un guión bajo en su nombre (`_`) no se dibujará aunque esta opción esté activada.

El valor predeterminado para esta opción es `yes`.

Sintaxis: `show_marks yes|no`

Ejemplo: `show_marks yes`

4.31. Continuación de línea

Los archivos de configuración admiten la continuación de línea, lo que significa que cuando finaliza una línea en un carácter de barra diagonal inversa (`\`), el analizador ignorará el salto de línea. Esta característica se puede utilizar para crear archivos de configuración más legibles. Las líneas comentadas no continúan.

Ejemplo:

```
bindsym Mod1+f \
fullscreen toggle

# this line is not continued \
bindsym Mod1+F fullscreen toggle
```

5. Configurando i3bar

La barra en la parte inferior de su monitor está dibujada por un proceso separado llamado `i3bar`. Tener esta parte de "la interfaz de usuario `i3`" en un proceso separado tiene varias ventajas:

1. Es un enfoque modular. Si no necesita una barra de espacio de trabajo, o si prefiere una diferente (`dzen2`, `xmobar`, tal vez incluso `gnome-panel`?), Puede eliminar la configuración de `i3bar` e iniciar su barra favorita.
2. Sigue la filosofía de `UNIX` de "*Hacer que cada programa haga una cosa y la haga bien*". Si bien `i3` administra bien sus ventanas, `i3bar` es bueno para mostrar una barra en cada monitor (a menos que lo configure de otra manera).
3. Conduce a dos bases de código limpias y separadas. Si desea comprender `i3`, no necesita preocuparse por los detalles de `i3bar` y viceversa.

Dicho esto, `i3bar` está configurado en el mismo archivo de configuración que `i3`. Esto se debe a que está estrechamente acoplado con `i3` (a diferencia de `i3lock` o `i3status`, que son útiles para las personas que usan otros administradores de ventanas). Por lo tanto, no tiene sentido usar un lugar de configuración diferente cuando ya tenemos una buena infraestructura de configuración.

La configuración de su barra de espacio de trabajo comienza con la apertura de un bloque de barras. Puede tener múltiples bloques de barras para usar diferentes configuraciones para diferentes salidas (monitores):

Ejemplo:

```
bar {
    status_command i3status
}
```

5.1. Comando i3bar

Por defecto, `i3` solo pasará `i3bar` y dejará que su shell maneje la ejecución, buscando en su `$PATH` una versión correcta. Si tiene un `i3bar` diferente en alguna parte o el binario no está en su `$PATH`, puede decirle a `i3` qué ejecutar.

El comando especificado se pasará a `sh -c`, por lo que puede usar `globbing` (pegajoso) y debe tener una cita correcta, etc.

Sintaxis: `i3bar_command <command>`

Ejemplo:

```
bar {
    i3bar_command /home/user/bin/i3bar
}
```

5.2. Comando `statusline`

`i3bar` puede ejecutar un programa y mostrar cada línea de su salida estándar en el lado derecho de la barra. Esto es útil para mostrar información del sistema como su dirección IP actual, estado de la batería o fecha/hora.

El comando especificado se pasará a `sh -c`, por lo que puede usar `globbing` y debe tener una cita correcta, etc. Tenga en cuenta que para el manejo de la señal, dependiendo de su `shell` (se sabe que los usuarios del `dash(1)` están afectados), tiene que usar el comando `exec` del `shell` para que las señales pasen a su programa, no al `shell`.

Sintaxis: `status_command <command>`

Ejemplo:

```
bar {
    status_command i3status --config ~/.i3status.conf

    # For dash(1) users who want signal handling to work:
    status_command exec ~/.bin/my_status_command
```

5.3. Modo de visualización

Puede hacer que `i3bar` sea visible permanentemente en un borde de la pantalla (`dock mode`, modo de acoplamiento) o que aparezca cuando presione su tecla modificadora (`hide mode`, modo oculto). También es posible forzar a `i3bar` a permanecer siempre oculto (`invisible mode`, modo invisible). La tecla modificadora se puede configurar usando la opción `modifier`.

La opción de el modo se puede cambiar durante el tiempo de ejecución a través del comando `bar mode`. Al recargar `i3`, el modo volverá a su valor configurado.

El modo oculto maximiza el espacio de la pantalla que se puede usar para ventanas reales. Además, `i3bar` envía las señales `SIGSTOP` y `SIGCONT` al proceso de la línea de estado para ahorrar batería.

El modo invisible permite maximizar permanentemente el espacio de la pantalla, ya que la barra nunca se muestra. Por lo tanto, puede configurar `i3bar` para que no lo moleste apareciendo debido a una sugerencia de urgencia o porque se presiona la tecla modificadora.

Para controlar si `i3bar` está oculto o se muestra en modo oculto, existe la opción `hidden_state`, que no tiene ningún efecto en modo de acoplamiento o modo invisible. Indica el estado oculto actual de la barra:

- (1) La barra actúa como en el modo oculto normal, y solo está oculta en caso de sugerencias de urgencia o presionando la tecla modificadora (`hide state`).
- (2) Es dibujado en la parte superior del espacio de trabajo actualmente visible (`show state`).

Al igual que el modo, `hidden_state` también se puede controlar a través de `i3`, esto se puede hacer mediante el comando `bar hidden_state`.

El modo predeterminado es el modo de acoplamiento (`dock mode`); en modo oculto, el modificador predeterminado es `Mod4` (generalmente la tecla de Windows). El valor predeterminado para `hidden_state` es `hide`.

Sintaxis:

```
mode dock|hide|invisible
hidden_state hide|show
modifier <Modifier>|none
```

Ejemplo:

```
bar {
  mode hide
  hidden_state hide
  modifier Mod1
}
```

Los modificadores disponibles son `Mod1-Mod5`, `Shift`, `Control` (ver `xmodmap (1)`). También puede usar `"none"` (ninguno) si no desea que ningún modificador active este comportamiento.

5.4. Comandos del botón del mouse

Especifica un comando para ejecutar cuando se presiona un botón en `i3bar` para anular el comportamiento predeterminado. Esto es útil, por ejemplo, para deshabilitar la acción de la rueda de desplazamiento o ejecutar scripts que implementen un comportamiento personalizado para estos botones.

Un botón siempre se denomina `button<n>`, donde `1` a `5` son botones predeterminados de la siguiente manera y los números más altos pueden ser botones especiales en dispositivos que ofrecen más botones:

`button1` Boton izquierdo del raton.

`button2` Botón central del ratón.

`button3` Botón derecho del mouse.

`button4` Rueda de desplazamiento hacia arriba.

`button5` Rueda de desplazamiento hacia abajo.

Tenga en cuenta que los antiguos comandos `wheel_up_cmd` y `wheel_down_cmd` están en desuso y se eliminarán en una versión futura. Recomendamos usar el `bindsym` más general con `button4` y `button5` en su lugar.

Sintaxis:

```
bindsym [--release] button<n> <command>
```

Ejemplo:

```
bar {
  # disable clicking on workspace buttons
  bindsym button1 nop
  # Take a screenshot by right clicking on the bar
  bindsym --release button3 exec --no-startup-id import /tmp/latest-screenshot.png
  # execute custom script when scrolling downwards
  bindsym button5 exec ~/.i3/scripts/custom_wheel_down
}
```

5.5. ID de barra

Especifica la **ID** de barra para la instancia de barra configurada. Si falta esta opción, la **ID** se establece en *bar-x*, donde *x* corresponde a la posición del bloque de barra de incrustación en el archivo de configuración (*bar-0*, *bar-1*, ...).

Sintaxis:

```
id <bar_id>
```

Ejemplo:

```
bar {
  id bar-1
}
```

5.6. Posición

Esta opción determina en qué borde de la pantalla debería aparecer **i3bar**.

El valor predeterminado es **bottom**.

Sintaxis:

```
position top|bottom
```

Ejemplo:

```
bar {
  position top
}
```

5.7. Salida (s)

Puede restringir **i3bar** a una o más salidas (monitores). El valor predeterminado es manejar todas las salidas. Restringir las salidas es útil para usar diferentes opciones para diferentes salidas mediante el uso de múltiples bloques de barras.

Para hacer que una instancia particular de **i3bar** maneje múltiples salidas, especifique la directiva de salida varias veces.

Sintaxis: `output primary|<output>`

Ejemplo:

```
# big monitor: everything
bar {
    # The display is connected either via HDMI or via DisplayPort
    output HDMI2
    output DP2
    status_command i3status
}

# laptop monitor: bright colors and i3status with less modules.
bar {
    output LVDS1
    status_command i3status --config ~/.i3status-small.conf
    colors {
        background #000000
        statusline #ffffff
    }
}

# show bar on the primary monitor and on HDMI2
bar {
    output primary
    output HDMI2
    status_command i3status
}
```

Tenga en cuenta que es posible que todavía no tenga una salida principal configurada. Para hacerlo, ejecuta:

```
xrandr --output <output> --primary
```

5.8. Salida de la bandeja

`i3bar` por defecto proporciona un área de bandeja del sistema donde los programas como `NetworkManager`, `VLC`, `Pidgin`, etc. pueden colocar pequeños íconos.

Puede configurar en qué salida (monitor) se deben mostrar los iconos o puede desactivar la funcionalidad por completo.

Puede usar varias directivas `tray_output` en su configuración para especificar una lista de salidas en las que desea que aparezca la bandeja. La primera salida disponible en esa lista, según lo definido por el orden de las directivas, se usará para la salida de la bandeja.

Sintaxis: `tray_output none|primary|<output>`

Ejemplo:

```
disable system tray
bar {
    tray_output none
}

# show tray icons on the primary monitor
bar {
    tray_output primary
}

# show tray icons on the big monitor
bar {
    tray_output HDMI2
}
```

Tenga en cuenta que es posible que todavía no tenga una salida principal configurada. Para hacerlo, ejecuta:

```
xrandr --output <output> --primary
```

Tenga en cuenta que cuando utiliza varios bloques de configuración de barra, tiene que especificar `tray_output primary` en todos ellos o especificar explícitamente `tray_output none` en las barras que no deberían mostrar la bandeja, de lo contrario, las diferentes instancias podrían competir entre sí al intentar mostrar los iconos de la bandeja.

5.9. Relleno de bandeja

La bandeja se muestra en el lado derecho de la barra. De manera predeterminada, se utiliza un relleno de 2 píxeles para el lado superior, inferior y derecho del área de la bandeja y entre los iconos individuales.

Sintaxis:

```
tray_padding <px> [px]
```

Ejemplo:

```
# Obey Fitts's law
tray_padding 0
```

5.10. Fuente

Especifica la fuente que se utilizará en la barra. Ver [Fuentes].

Sintaxis: `font `

Ejemplo:

```
bar {
    font -misc-fixed-medium-r-normal--13-120-75-75-C-70-iso10646-1
    font pango:DejaVu Sans Mono 10
}
```

5.11. Símbolo separador personalizado

Especifica un símbolo personalizado que se utilizará para el separador en lugar del separador vertical de un píxel de grosor.

Sintaxis: `separator_symbol <symbol>`

Ejemplo:

```
bar {
    separator_symbol " : | : "
}
```

5.12. Botones del espacio de trabajo

Especifica si los botones del espacio de trabajo deben mostrarse o no. Esto es útil si desea mostrar una barra de solo línea de estado que contiene información adicional.

El valor predeterminado es mostrar los botones del espacio de trabajo.

Sintaxis: `workspace_buttons yes|no`

Ejemplo:

```
bar {
    workspace_buttons no
}
```

5.13. Ancho mínimo para botones de espacio de trabajo

Por defecto, el ancho del botón de un espacio de trabajo está determinado por el ancho del texto que muestra el nombre del espacio de trabajo. Si el nombre es demasiado corto (digamos, una letra), entonces el botón del espacio de trabajo puede parecer demasiado pequeño.

Esta opción especifica el ancho mínimo para los botones del espacio de trabajo. Si el nombre de un espacio de trabajo es demasiado corto para cubrir el botón, se agrega un relleno adicional a ambos lados del botón para que el texto quede centrado.

El valor predeterminado de cero significa que no se agrega relleno adicional.

La configuración también se aplica al indicador del modo de encuadernación actual.

Tenga en cuenta que los píxeles especificados se refieren a píxeles lógicos, que pueden traducirse en más píxeles en pantallas [HiDPI](#).

Sintaxis: `workspace_min_width <px> [px]`

Ejemplo:

```
bar {
    workspace_min_width 40
}
```

5.14. Números/nombre del espacio de trabajo

Especifica si los números del espacio de trabajo deben mostrarse dentro de los botones del espacio de trabajo. Esto es útil si desea tener un espacio de trabajo con nombre que se mantenga en orden en la barra de acuerdo con su número sin mostrar el prefijo del número.

Cuando `strip_workspace_numbers` se establece en `yes`, cualquier espacio de trabajo que tenga el nombre de la forma "[n][:][NAME]" mostrará solo el nombre. Puede usar esto, por ejemplo, para mostrar números romanos en lugar de dígitos nombrando sus espacios de trabajo a "1:I", "2:II", "3:III", "4:IV", ...

Cuando `strip_workspace_name` se establece en `yes`, cualquier espacio de trabajo que tenga un nombre con la forma "[n][:][NAME]" mostrará solo el número.

El valor predeterminado es mostrar el nombre completo dentro del botón del espacio de trabajo. Tenga en cuenta que los dos puntos en el nombre del espacio de trabajo son opcionales, por lo que `[n][NOMBRE]` también tendrá el nombre y el número del espacio de trabajo correctamente despojado.

Sintáxis:

```
strip_workspace_numbers yes|no
strip_workspace_name yes|no
```

Ejemplo:

```
bar {
    strip_workspace_numbers yes
}
```

5.15. Indicador de modo de enlaces

Especifica si se debe mostrar o no el indicador del modo de enlace actual. Esto es útil si desea ocultar los botones del espacio de trabajo pero aún así puede ver el indicador del modo de enlaces actual. Consulte [Modos de enlace] para conocer qué modos son y cómo usarlos.

El valor predeterminado es mostrar el indicador de modo.

Sintáxis: `binding_mode_indicator yes|no`

Ejemplo:

```
bar {
    binding_mode_indicator no
}
```

5.16. Colores

Al igual que con [i3](#), los colores están en formato hexadecimal HTML (`#rrggbb`). Los siguientes colores se pueden configurar en este momento:

`background`

Color de fondo de la barra.

`statusline`

Color del texto que se utilizará para la línea de estado.

`separator`

Color del texto que se utilizará para el separador.

`focused_background`

Color de fondo de la barra en la salida de monitor enfocada actualmente. Si no se usa, el color se tomará del fondo.

`focused_statusline`

Color de texto que se utilizará para la línea de estado en la salida de monitor enfocada actualmente. Si no se usa, el color se tomará de la línea de estado.

`focused_separator`

Color de texto que se utilizará para el separador en la salida de monitor enfocada actualmente. Si no se usa, el color se tomará del separador.

`focused_workspace`

Color de borde, fondo y texto para un botón de espacio de trabajo cuando el espacio de trabajo tiene foco.

`active_workspace`

Color de borde, fondo y texto para un botón del espacio de trabajo cuando el espacio de trabajo está activo (visible) en alguna salida, pero el foco está en otro. Solo puede distinguir esto aparte del espacio de trabajo enfocado cuando usa varios monitores.

`inactive_workspace`

Color de borde, fondo y texto para un botón de espacio de trabajo cuando el espacio de trabajo no tiene foco y no está activo (visible) en ninguna salida. Este será el caso para la mayoría de los espacios de trabajo.

`urgent_workspace`

Color de borde, fondo y texto para un botón de espacio de trabajo cuando el espacio de trabajo contiene una ventana con el conjunto de sugerencias de urgencia.

`binding_mode`

Color de borde, fondo y texto para el indicador de modo de encuadernación. Si no se utiliza, los colores se tomarán del espacio de trabajo urgente.

Sintaxis

```

colors {
    background <color>
    statusline <color>
    separator <color>

    <colorclass> <border> <background> <text>
}

```

Ejemplo (colores por defecto):

```

bar {
    colors {
        background #000000
        statusline #ffffff
        separator #666666
        focused_workspace #4c7899 #285577 #ffffff
        active_workspace #333333 #5f676a #ffffff
        inactive_workspace #333333 #222222 #888888
        urgent_workspace #2f343a #900000 #ffffff
        binding_mode #2f343a #900000 #ffffff
    }
}

```

5.17. Transparencia

`i3bar` puede soportar la transparencia al pasar el indicador `--transparency` en la configuración:

```

bar {
    i3bar_command i3bar --transparency
}

```

En la configuración de color de `i3bar` y el atributo de color del bloque de estado de `i3bar`, puede usar colores en el formato `RGBA`, es decir, los dos últimos dígitos (hexadecimales) especifican la opacidad. Por ejemplo, `#00000000` será completamente transparente, mientras que `#000000FF` será un negro completamente opaco (igual que `#000000`).

Tenga en cuenta que, debido a la forma en que funciona la especificación de la bandeja, habilitar esta marca hará que todos los iconos de la bandeja tengan un fondo transparente.

6. Lista de comandos

Los comandos son los que vinculan a las pulsaciones de teclas específicas. También puede emitir comandos en tiempo de ejecución sin presionar una tecla utilizando la interfaz `IPC`. Una manera fácil de hacer esto es usar la utilidad `i3-msg`:

Ejemplo:

```
# execute this on your shell to make the current container borderless
i3-msg border none
```

Los comandos se pueden encadenar usando ; (un punto y coma). Entonces, para mover una ventana a un espacio de trabajo específico e inmediatamente cambiar a ese espacio de trabajo, puede configurar la siguiente combinación de teclas:

```
bindsym $mod+x move container to workspace 3; workspace 3
```

Además, puede cambiar el alcance de un comando (criterios de comando), es decir, qué contenedores deberían verse afectados por ese comando, utilizando varios criterios. Los criterios se especifican antes de cualquier comando en un par de corchetes y están separados por espacio.

Cuando use varios comandos, sepárelos usando , (una coma) en lugar de un punto y coma. Los criterios se aplican solo hasta el próximo punto y coma, por lo que si usa un punto y coma para separar los comandos, solo se ejecutará el primero para las ventanas coincidentes.

Ejemplo:

```
# if you want to kill all windows which have the class Firefox, use:
bindsym $mod+x [class="Firefox"] kill

# same thing, but case-insensitive
bindsym $mod+x [class="(?)firefox"] kill

# kill only the About dialog from Firefox
bindsym $mod+x [class="Firefox" window_role="About"] kill

# enable floating mode and move container to workspace 4
for_window [class="^evil-app$"] floating enable, move container to workspace 4

# move all floating windows to the scratchpad
bindsym $mod+x [floating] move scratchpad
```

Los criterios que se implementan actualmente son:

class

Compara la clase de ventana (la segunda parte de `WM_CLASS`). Use el valor especial `__focused__` para hacer coincidir todas las ventanas que tengan la misma clase de ventana que la ventana enfocada actualmente.

instance

Compara la instancia de la ventana (la primera parte de `WM_CLASS`). Use el valor especial `__focused__` para hacer coincidir todas las ventanas que tengan la misma instancia de ventana que la ventana enfocada actualmente.

`window_role`

Compara el rol de la ventana (`WM_WINDOW_ROLE`). Use el valor especial `__focused__` para hacer coincidir todas las ventanas que tienen el mismo rol de ventana que la ventana enfocada actualmente.

`window_type`

Compare el tipo de ventana (`_NET_WM_WINDOW_TYPE`). Los valores posibles son `normal, dialog, utility, toolbar, splash, menu, dropdown_menu, popup_menu, tooltip notification` (normal, diálogo, utilidad, barra de herramientas, bienvenida, menú, menú desplegable, menú emergente, información sobre herramientas y notificación).

`id`

Compara la ID de la ventana X11, que puede obtener a través de `xwininfo`, por ejemplo.

`title`

Compara el título de la ventana X11 (`_NET_WM_NAME` o `WM_NAME` como reserva). Use el valor especial `__focused__` para hacer coincidir todas las ventanas que tengan el mismo título de ventana que la ventana enfocada actualmente.

`urgent`

Compara el estado urgente de la ventana. Puede ser `"latest"` u `"oldest"`. Coincide con la ventana urgente más reciente o más antigua, respectivamente. (Los siguientes alias también están disponibles: `newest, last, recent, first`, más nuevo, último, reciente, primero).

`workspace`

Compara el nombre del espacio de trabajo al que pertenece la ventana. Use el valor especial `__focused__` para hacer coincidir todas las ventanas en el espacio de trabajo actualmente enfocado.

`con_mark`

Compara las marcas establecidas para este contenedor, consulte [Marcas similares a VIM]. Se hace una coincidencia si alguna de las marcas del contenedor coincide con la marca especificada.

`con_id`

Compara el ID del contenedor interno i3, que puede obtener a través de la interfaz IPC. Práctico para scripting. Use el valor especial `__focused__` para que coincida solo con la ventana actualmente enfocada.

`floating`

Solo coincide con ventanas flotantes. Este criterio no requiere ningún valor.

`tiling`

Solo coincide con ventanas de mosaico. Este criterio no requiere ningún valor.

Los criterios de las clases, `class`, `instance`, `role`, `title`, `workspace` y `mark`, (instancia, rol, título, espacio de trabajo y marca) son en realidad expresiones regulares (PCRE). Consulte `pcresyntax(3)` o `perldoc perlre` para obtener información sobre cómo usarlos.

6.1. Ejecutando aplicaciones (`exec`)

¿De qué sirve un administrador de ventanas si realmente no puede iniciar ninguna aplicación? El comando `exec` inicia una aplicación pasando el comando que especifique a un shell. Esto implica que puede usar globbing (comodines) y se buscarán programas en su `$PATH`.

Vea [varios comandos] para detalles sobre el significado especial de `;` (punto y coma) y `,` (coma): encadenan los comandos en `i3`, por lo que debe usar cadenas entre comillas (como se muestra en [comando a ejecutar]) si aparecen en su comando.

Sintaxis:

```
exec [--no-startup-id] <command>
```

Ejemplo:

```
# Start the GIMP
bindsym $mod+g exec gimp

# Start the terminal emulator urxvt which is not yet startup-notification-aware
bindsym $mod+Return exec --no-startup-id urxvt
```

El parámetro `--no-startup-id` deshabilita el soporte de notificación de inicio para este comando `exec` en particular. Con la notificación de inicio, `i3` puede asegurarse de que aparezca una ventana en el espacio de trabajo en el que utilizó el comando `exec`. Además, cambiará el cursor `X11` para mirar (un reloj) mientras se inicia la aplicación. Por lo tanto, si una aplicación no reconoce la notificación de inicio (aunque la mayoría de las aplicaciones que usan `GTK` y `Qt` parecen serlo), terminará con un cursor de observación durante 60 segundos.

Si el comando a ejecutar contiene un `;` (punto y coma) y/o una `,` (coma), se debe citar todo el comando. Por ejemplo, para tener una combinación de teclas para el comando de shell `notify-send Hello, i3`, `i3`, agregaría una entrada a su archivo de configuración como esta:

Sintaxis:

```
# Execute a command with a comma in it
bindsym $mod+p exec "notify-send Hello, i3"
```

Sin embargo, si un comando con una coma (`,`) y/o punto y coma (`;`) requiere comillas, debe escapar a las comillas internas con barras diagonales inversas, como esta:

Ejemplo:

```
# Execute a command with a comma, semicolon and internal quotes
bindsym $mod+p exec "notify-send \"Hello, i3; from $USER\""
```

6.2. División de contenedores

El comando `splitv` (dividir) convierte la ventana actual en un contenedor dividido. Los contenedores divididos pueden contener múltiples ventanas. Dependiendo del diseño del contenedor dividido, las ventanas nuevas se colocan a la derecha del actual (`splith`) o las ventanas nuevas se colocan debajo del actual (`splitv`).

Si aplica este comando a un contenedor dividido con la misma orientación, no sucederá nada. Si usa una orientación diferente, la orientación del contenedor dividido cambiará (si no tiene más de una ventana). La opción de alternar alternará la orientación del contenedor dividido si contiene una sola ventana. De lo contrario, convierte la ventana actual en un contenedor dividido con orientación opuesta en comparación con el contenedor principal. Use la distribución de alternancia de diseño para cambiar el diseño de cualquier contenedor dividido de `splitv` a `splith` o viceversa. También puede definir una secuencia personalizada de diseños para alternar con la alternancia de diseño, consulte `[manipulating_layout]`.

6.3. Diseño de manipulación

Utilice `layout toggle split` (alternar diseño dividido), `layout stacking` (diseño apilado), `layout tabbed`, (diseño con pestañas), `layout splitv` o `layout splith`, el diseño de `splitv` o el diseño de `splith` para cambiar el diseño actual del contenedor a `splith/splitv`, `stacking`, `tabbed layout`, `splitv` o `splith`, `splitv` o `splith`, respectivamente.

Especifique hasta cuatro diseños después de alternar el diseño para recorrerlos. Cada vez que se ejecuta el comando, se aplicará el diseño especificado después del actualmente activo. Si el diseño actualmente activo no está en la lista, se activará el primer diseño de la lista.

Para hacer que la ventana actual (!) se muestre en pantalla completa, use `fullscreen enable` (habilitar pantalla completa), o `fullscreen enable global` para habilitar pantalla completa en forma global, para deshabilitar el modo de pantalla completa use `fullscreen disable`, y para alternar entre estos dos estados use `fullscreen toggle` (o `fullscreen toggle global` para alternar la pantalla completa o no, en forma global).

Del mismo modo, para hacer que la ventana actual sea flotante (o de nuevo en mosaico), use `floating enable`, `floating disable` (o `floating toggle`), respectivamente.

Sintaxis:

```
layout default|tabbed|stacking|splitv|splith
layout toggle [split|all]
layout toggle [split|tabbed|stacking|splitv|splith] [split|tabbed|stacking|splitv|splith]...
```

Ejemplo:

```
bindsym $mod+s layout stacking
bindsym $mod+l layout toggle split
bindsym $mod+w layout tabbed

# Toggle between stacking/tabbed/split:
bindsym $mod+x layout toggle
```

```
# Toggle between stacking/tabbed/splith/splitv:
bindsym $mod+x layout toggle all

# Toggle between stacking/tabbed/splith:
bindsym $mod+x layout toggle stacking tabbed splith

# Toggle between splitv/tabbed
bindsym $mod+x layout toggle splitv tabbed

# Toggle between last split layout/tabbed/stacking
bindsym $mod+x layout toggle split tabbed stacking

# Toggle fullscreen
bindsym $mod+f fullscreen toggle

# Toggle floating/tiling
bindsym $mod+t floating toggle
```

6.4. Enfoque de contenedores

Para cambiar el enfoque, puede usar el comando `focus`. Las siguientes opciones están disponibles:

`<criteria>`

Establece el foco en el contenedor que coincide con los criterios especificados. Ver [criterios de comando].

`left|right|up|down`

Establece el foco en el contenedor más cercano en la dirección dada.

`parent`

Establece el foco en el contenedor principal del contenedor actual.

`child`

Lo opuesto a `focus parent`, establece el foco en el último contenedor secundario enfocado.

`next|prev`

Establece automáticamente el foco en el contenedor adyacente. Si se especifica `sibling`, el comando enfocará el contenedor hermano exacto, incluidos los contenedores que no son hojas como los contenedores divididos. De lo contrario, es una versión automática de `focus left|right|up|down` en la orientación del contenedor principal.

`floating`

Establece el foco en el último contenedor flotante enfocado.

tiling

Establece el foco en el último contenedor de mosaico enfocado.

modo_toggle

Alterna entre contenedores floating/tiling (flotante o mosaico).

output

Seguido de una dirección o un nombre de salida, esto enfocará la salida correspondiente.

Sintaxis:

```

<criteria> focus
focus left|right|down|up
focus parent|child|floating|tiling|mode_toggle
focus next|prev [sibling]
focus output left|right|up|down|primary|<output>

```

Ejemplo:

```

# Focus firefox
bindsym $mod+F1 [class="Firefox"] focus

# Focus container on the left, bottom, top, right
bindsym $mod+j focus left
bindsym $mod+k focus down
bindsym $mod+l focus up
bindsym $mod+semicolon focus right

# Focus parent container
bindsym $mod+u focus parent

# Focus last floating/tiling container
bindsym $mod+g focus mode_toggle

# Focus the output right to the current one
bindsym $mod+x focus output right

# Focus the big output
bindsym $mod+x focus output HDMI-2

# Focus the primary output
bindsym $mod+x focus output primary

```

Tenga en cuenta que es posible que todavía no tenga una salida principal configurada. Para hacerlo, ejecuta:

```
xrandr --output <output> --primary
```

6.5. Mover contenedores

Use el comando `move` para mover un contenedor.

Sintáxis:

```
Moves the container into the given direction.
# The optional pixel argument specifies how far the
# container should be moved if it is floating and
# defaults to 10 pixels.
move <left|right|down|up> [<px> px]

# Moves the container to the specified pos_x and pos_y
# coordinates on the screen.
move position <pos_x> [px] <pos_y> [px]

# Moves the container to the center of the screen.
# If 'absolute' is used, it is moved to the center of
# all outputs.
move [absolute] position center

# Moves the container to the current position of the
# mouse cursor. Only affects floating containers.
move position mouse
```

Ejemplo:

```
# Move container to the left, bottom, top, right
bindsym $mod+j move left
bindsym $mod+k move down
bindsym $mod+l move up
bindsym $mod+semicolon move right

# Move container, but make floating containers
# move more than the default
bindsym $mod+j move left 20 px

# Move floating container to the center of all outputs
bindsym $mod+c move absolute position center

# Move container to the current position of the cursor
bindsym $mod+m move position mouse
```

6.6. Intercambiar contenedores

Se pueden intercambiar dos contenedores (es decir, moverse a la posición del otro) mediante el comando de intercambio. Asumirán la posición y la geometría del contenedor con el que se intercambian.

El primer contenedor que participa en el intercambio se puede seleccionar a través del proceso normal de criterios de comando, siendo la ventana enfocada el respaldo habitual si no se especifican criterios. El segundo contenedor se puede seleccionar utilizando uno de los siguientes métodos:

ID

La **ID** de la ventana **X11** de una ventana de cliente.

con_id

El **ID** del contenedor **i3** de un contenedor.

mark

Un contenedor con la marca especificada, consulte [Marcas similares a VIM].

Tenga en cuenta que el intercambio no funciona con todos los contenedores. En particular, el intercambio de contenedores que tienen una relación padre-hijo entre sí no funciona.

Sintáxis:

```
swap container with id|con_id|mark <arg>
```

Ejemplo:

```
# Swaps the focused container with the container marked »swapee«.
swap container with mark swapee
```

```
# Swaps container marked »A« and »B«
[con_mark="^A$"] swap container with mark B
```

6.7. Ventanas flotantes adhesivas

Si desea que una ventana se adhiera al cristal, es decir, que permanezca en la pantalla incluso si cambia a otro espacio de trabajo, puede usar el comando **sticky** (adhesivo). Por ejemplo, esto puede ser útil para blocs de notas, un reproductor multimedia o una ventana de video chat.

Tenga en cuenta que si bien cualquier ventana se puede hacer pegajosa mediante este comando, solo tendrá efecto si la ventana está flotando.

Sintáxis:

```
sticky enable|disable|toggle
```

Ejemplo:

```
# make a terminal sticky that was started as a notepad
for_window [instance=notepad] sticky enable
```

6.8. Cambio a espacios de trabajo (con nombre)/traslado a espacios de trabajo

Para cambiar a un espacio de trabajo específico, use el comando `workspace`, seguido del número o nombre del espacio de trabajo. Pase el indicador opcional `--no-auto-back-and-forward` para deshabilitar [Avance y retroceso automático] solo para esta llamada específica.

Para mover contenedores a espacios de trabajo específicos, use `move container to workspace`.

También puede cambiar al espacio de trabajo siguiente y anterior con los comandos `workspace next` y `workspace prev`, lo cual es útil, por ejemplo, si tiene el espacio de trabajo 1, 3, 4 y 9 y desea recorrerlos con una sola combinación de teclas. Para restringirlos a la salida actual, use `workspace next_on_output` y `workspace prev_on_output`. De manera similar, puede usar `move container to workspace next`, `move container to workspace prev` para mover un contenedor al espacio de trabajo siguiente/anterior y `move container to workspace current` (el último solo tiene sentido cuando se usa con criterios).

`move container next` pasa por espacios de trabajo numerados o con nombre. Pero cuando alcanza el último espacio de trabajo numerado/nombrado, busca espacios de trabajo nombrados después de agotar los numerados y busca los numerados después de agotar los nombrados.

Consulte [Mover contenedores/espacios de trabajo a salidas RandR] para saber cómo mover un contenedor/espacio de trabajo a una salida `RandR` diferente.

`I3bar` analiza los nombres de los espacios de trabajo como marcas de Pango.

Para volver al espacio de trabajo previamente enfocado, use `workspace back_and_forth`; del mismo modo, puede mover contenedores al espacio de trabajo previamente enfocado usando `move container to workspace back_and_forth`.

6.8.1. Espacios de Trabajos Nombrados

Los espacios de trabajo se identifican por su nombre. Entonces, en lugar de usar números en el comando del espacio de trabajo, puede usar un nombre arbitrario:

Ejemplo:

```
bindsym $mod+1 workspace mail
```

...

Si desea que el espacio de trabajo tenga un número y un nombre, simplemente prefije el número, así:

Ejemplo:

```
bindsym $mod+1 workspace 1: mail
```

```
bindsym $mod+2 workspace 2: www
```

...

Tenga en cuenta que el espacio de trabajo realmente se llamará "`1:mail`". `i3` trata los nombres de espacios de trabajo que comienzan con un número de una manera ligeramente especial.

Normalmente, los espacios de trabajo con nombre se ordenan de la forma en que aparecen. Cuando comienzan con un número, `i3` los ordenará numéricamente. Además, podrá usar el espacio

de trabajo número 1 para cambiar al espacio de trabajo que comienza con el número 1, independientemente del nombre que tenga. Esto es útil en caso de que cambie el nombre del espacio de trabajo dinámicamente. Para combinar ambos comandos, puede usar el espacio de trabajo número 1: correo para especificar un nombre predeterminado si actualmente no hay un espacio de trabajo que comience con un "1".

6.8.2. Renombrar espacios de trabajo

Puede cambiar el nombre de los espacios de trabajo. Esto puede ser útil para comenzar con los espacios de trabajo numerados predeterminados, hacer su trabajo y cambiar el nombre de los espacios de trabajo después para reflejar lo que realmente hay en ellos. También puede omitir el nombre anterior para cambiar el nombre del espacio de trabajo enfocado actualmente. Esto es útil si desea utilizar el comando de cambio de nombre con `i3-input`.

Sintaxis:

```
rename workspace <old_name> to <new_name>
rename workspace to <new_name>
```

Ejemplo:

```
3-msg 'rename workspace 5 to 6'
i3-msg 'rename workspace 1 to "1: www"'
i3-msg 'rename workspace "1: www" to "10: www"'
i3-msg 'rename workspace to "2: mail"'
bindsym $mod+r exec i3-input -F 'rename workspace to "%s"' -P 'New name: '
```

Si desea cambiar el nombre de los espacios de trabajo a pedido mientras mantiene estable la navegación, puede usar una configuración como esta:

Ejemplo:

```
bindsym $mod+1 workspace number "1: www"
bindsym $mod+2 workspace number "2: mail"
...
```

Si no existe un espacio de trabajo, el número de espacio de trabajo de comando `"1: mail"` creará el espacio de trabajo `"1:mail"`.

Si ya existe un espacio de trabajo con el número 1, el comando cambiará a este espacio de trabajo e ignorará la parte de texto. Por lo tanto, incluso cuando el espacio de trabajo ha cambiado de nombre a `"1:web"`, el comando anterior seguirá cambiando a él.

6.9. Mover espacios de trabajo a una pantalla diferente

Consulte [Mover contenedores/espacios de trabajo a salidas RandR] para saber cómo mover un contenedor/espacio de trabajo a una salida `RandR` diferente.

6.10. Mover contenedores/espacios de trabajo a salidas RandR

Para mover un contenedor a otra salida RandR (dirigida por nombres como LVDS1 o VGA1) o a una salida RandR identificada por una dirección específica (como left, right, up o down), hay dos comandos:

Sintaxis:

```
move container to output left|right|down|up|current|primary|<output>
move workspace to output left|right|down|up|current|primary|<output>
```

Ejemplo:

```
# Move the current workspace to the next output
# (effectively toggles when you only have two outputs)
bindsym $mod+x move workspace to output right

# Put this window on the presentation output.
bindsym $mod+x move container to output VGA1

# Put this window on the primary output.
bindsym $mod+x move container to output primary
```

Tenga en cuenta que es posible que todavía no tenga una salida principal configurada. Para hacerlo, ejecuta:

```
xrandr --output <output> --primary
```

6.11. Mover contenedores/ventanas a marcas

Para mover un contenedor a otro contenedor con una marca específica (consulte [Marcas similares a VIM]), puede usar el siguiente comando.

La ventana se moverá justo después del contenedor marcado en el árbol, es decir, termina en la misma posición que si hubiera abierto una nueva ventana cuando el contenedor marcado estaba enfocado. Si la marca está en un contenedor dividido, la ventana aparecerá como un nuevo hijo después del hijo enfocado actualmente dentro de ese contenedor.

Sintaxis:

```
move window|container to mark <mark>
```

Ejemplo:

```
for_window [instance="tabme"] move window to mark target
```

6.12. Cambio de tamaño de contenedores/ventanas

Si desea cambiar el tamaño de los contenedores/ventanas usando su teclado, puede usar el comando de cambio de tamaño:

Sintaxis:

```
size grow|shrink <direction> [<px> px [or <ppt> ppt]]
resize set [width] <width> [px | ppt]
resize set height <height> [px | ppt]
resize set [width] <width> [px | ppt] [height] <height> [px | ppt]
```

La dirección puede ser `up`, `down`, `left` o `right` (arriba, abajo, izquierda o derecha). O puede ser menos específico y usar `width` o `height` (ancho o alto), en cuyo caso `i3` ocupará/dará espacio a todos los demás contenedores. El argumento opcional de píxeles especifica cuántos de ellos debe crecer o reducirse un contenedor (el valor predeterminado es 10 píxeles). El argumento opcional `ppt` significa "puntos porcentuales" y, si se especifica, indica que un contenedor de mosaico debe crecer o reducirse en tantos puntos, en lugar del valor de `px`.

Nota sobre `resize set` (cambio de tamaño): un valor de `0` para `<width>` o `<height>` (<ancho> o <alto>) significa "no cambiar el tamaño en esta dirección".

Se recomienda definir enlaces para cambiar el tamaño en un modo de enlace dedicado. Consulte [Modos de enlace] y el ejemplo en la [Configuración por defecto] de `i3` para obtener más contexto.

Ejemplo:

```
for_window [class="urxvt"] resize set 640 480
```

6.13. Saltar a ventanas específicas

A menudo, en un entorno de monitores múltiples, desea saltar rápidamente a una ventana específica. Por ejemplo, mientras trabaja en el espacio de trabajo 3, puede saltar a su cliente de correo para enviarle un correo electrónico a su jefe que ha logrado un objetivo importante. En lugar de descubrir cómo navegar a su cliente de correo, sería más conveniente tener un acceso directo. Puede usar el comando de enfoque con criterios (`focus`) para eso.

Sintaxis:

```
[class="class"] focus
[title="title"] focus
```

Ejemplo:

```
# Get me to the next open VIM instance
bindsym $mod+a [class="urxvt" title="VIM"] focus
```

6.14. Marcas similares a VIM (mark/goto)

Esta característica es como la función de salto: le permite saltar directamente a una ventana específica (esto significa cambiar al espacio de trabajo apropiado y establecer el foco en las ventanas). Sin embargo, puede marcar directamente una ventana específica con una etiqueta arbitraria y usarla después. Puede desmarcar la etiqueta de la misma manera, usando el comando desmark. Si no especifica una etiqueta, desmark elimina todas las marcas. No necesita asegurarse de que sus ventanas tengan clases o títulos únicos, y no necesita cambiar su archivo de configuración.

Como el comando debe incluir la etiqueta con la que desea marcar la ventana, no puede simplemente vincularlo a una tecla. `i3-input` es una herramienta creada para este propósito: le permite ingresar un comando y envía el comando a `i3`. También puede prefijar este comando y mostrar un mensaje personalizado para el diálogo de entrada.

La opción adicional `--toggle` eliminará la marca si la ventana ya tiene esta marca o la agregará de otra manera. Tenga en cuenta que es posible que necesite usar esto en combinación con `--add` (ver más abajo) ya que de lo contrario se eliminarán otras marcas.

El indicador `--replace` hace que `i3` elimine las marcas existentes, que también es el comportamiento predeterminado. Puede usar la bandera `--add` para colocar más de una marca en una ventana.

Consulte [Dibujar marcas en la decoración de ventanas] si no desea que se muestren marcas en la decoración de la ventana.

Sintaxis:

```
mark [--add|--replace] [--toggle] <identifier>
[con_mark="identifier"] focus
unmark <identifier>
```

Puede usar `i3-input` para solicitar un nombre de marca, luego use los comandos de marca y foco para crear y saltar a marcas personalizadas:

Ejemplo:

```
# read 1 character and mark the current window with this character
bindsym $mod+m exec i3-input -F 'mark %s' -l 1 -P 'Mark: '

# read 1 character and go to the window with the character
bindsym $mod+g exec i3-input -F '[con_mark="%s"] focus' -l 1 -P 'Goto: '
```

Alternativamente, si no desea ensuciar `i3-input`, puede crear enlaces separados para un conjunto específico de etiquetas y luego usar solo esas etiquetas:

Ejemplo (en una terminal):

```
#marks the focused container
mark irssi
```

```
# focus the container with the mark "irssi"
'[con_mark="irssi"] focus'

# remove the mark "irssi" from whichever container has it
unmark irssi

# remove all marks on all firefox windows
[class="(?)firefox"] unmark
```

6.15. Formato de título de ventana

Por defecto, `i3` simplemente imprimirá el título de la ventana `X11`. Usando `title_format`, esto se puede personalizar configurando el formato a la salida deseada. Esta directiva admite [Pango markup](#) (Lenguaje de marcado simple con atributos para texto) y los siguientes marcadores de posición que se reemplazarán:

`%title`

Para ventanas normales, este es el título de la ventana `X11` (`_NET_WM_NAME` o `WM_NAME` como reserva). Cuando se utiliza en contenedores sin ventana (por ejemplo, un contenedor dividido dentro de un diseño con `tabbed/stacked`, con pestañas/apiladas), esta será la representación en árbol del contenedor (por ejemplo, "H [xterm xterm]").

`%class`

La clase de ventana `X11` (segunda parte de `WM_CLASS`). Esto corresponde al criterio de clase, ver [criterios de comando].

`%instance`

La instancia de la ventana `X11` (primera parte de `WM_CLASS`). Esto corresponde al criterio de instancia, ver [criterios de comando].

Usando la directiva [Comandos arbitrarios para ventanas específicas], puede establecer el formato del título para cualquier ventana basada en [criterios de comando].

Sintaxis:

```
title_format <format>
```

Ejemplo:

```
give the focused window a prefix
bindsym $mod+p title_format "Important | %title"

# print all window titles bold
for_window [class=".*"] title_format "<b>%title</b>"

# print window titles of firefox windows red
for_window [class="(?)firefox"] title_format "<span foreground='red'>%title</span>"
```

6.16. Cambiar estilo de borde

Para cambiar el borde del cliente actual, puede usar `border normal` (borde normal) para usar el borde normal (incluido el título de la ventana), `border pixel 1` para usar un borde de 1 píxel (sin título de ventana) y `border none` para que el cliente no tenga bordes.

También hay `border toggle` que alternará los diferentes estilos de borde. El argumento opcional de píxeles se puede utilizar para especificar el ancho del borde al cambiar a los estilos normal y de píxeles.

Tenga en cuenta que "pixel" se refiere a píxeles lógicos. En las pantallas HiDPI, un píxel lógico puede estar representado por múltiples píxeles físicos, por lo que el `pixel 1` no necesariamente se traduce en un borde ancho de una fila de píxeles.

Sintaxis:

```
border normal|pixel|toggle [<n>]
border none

# legacy syntax, equivalent to "border pixel 1"
border 1pixel
```

Ejemplo:

```
# use window title, but no border
bindsym $mod+t border normal 0
# use no window title and a thick border
bindsym $mod+y border pixel 3
# use neither window title nor border
bindsym $mod+u border none
```

6.17. Habilitar el registro de memoria compartida

Como se describe en <https://i3wm.org/docs/debugging.html>, i3 puede iniciar sesión en un búfer de memoria compartida, que puede volcar usando `i3-dump-log`. El comando `shmlog` le permite habilitar o deshabilitar el registro de memoria compartida en tiempo de ejecución.

Tenga en cuenta que cuando use `shmlog <size_in_bytes>`, el registro actual se descartará y se iniciará uno nuevo.

Sintaxis:

```
shmlog <size_in_bytes>
shmlog on|off|toggle
```

Ejemplo:

```
# Enable/disable logging
bindsym $mod+x shmlog toggle
```

```
# or, from a terminal:
# increase the shared memory log buffer to 50 MiB
i3-msg shmlog $((50*1024*1024))
```

6.18. Habilitar el registro de depuración

El comando `debuglog` le permite habilitar o deshabilitar el registro de depuración en tiempo de ejecución. El registro de depuración es mucho más detallado que el registro sin depuración. Este comando no activa el registro de memoria compartida (`shmlog`) y, como tal, es muy útil en combinación con el comando [Habilitar el registro de memoria compartida] descrito anteriormente.

Sintaxis:

```
debuglog on|off|toggle
```

Ejemplo:

```
# Enable/disable logging
bindsym $mod+x debuglog toggle
```

6.19. Recarga/reinicio/salida

Puede hacer que `i3` recargue su archivo de configuración con `reload`. También puede reiniciar `i3` in situ con el comando `reload` para salir de un estado extraño (si eso ocurriera alguna vez) o para realizar una actualización sin tener que reiniciar su sesión `X`. Para salir de `i3` correctamente, puede usar el comando `exit`, sin embargo, no es necesario (simplemente matar su sesión `X` también está bien).

Ejemplo:

```
bindsym $mod+Shift+r restart
bindsym $mod+Shift+w reload
bindsym $mod+Shift+e exit
```

6.20. Scratchpad

Hay dos comandos para usar cualquier ventana existente como ventana de `scratchpad`. `move scratchpad` moverá una ventana al espacio de trabajo de `scratchpad`. Esto lo hará invisible hasta que lo muestres nuevamente. No hay forma de abrir ese espacio de trabajo. En cambio, cuando use el programa `scratchpad show`, la ventana se mostrará nuevamente, como una ventana flotante, centrada en su espacio de trabajo actual (el uso de `scratchpad show` en una ventana visible de `scratchpad` la volverá a ocultar, por lo que puede tener una combinación de teclas para alternar). Tenga en cuenta que esta es solo una ventana flotante normal, por lo que si desea "eliminarla del bloc de notas", simplemente puede hacer que se vuelva a colocar en mosaico (`floating toggle`).

Como su nombre lo indica, esto es útil para tener una ventana con su editor favorito siempre a mano. Sin embargo, también puede usar esto para otras aplicaciones que se ejecutan permanentemente y que no desea ver todo el tiempo: ¿Su reproductor de música, `alsamixer`, tal vez incluso su cliente de correo ...?

Sintáxis:

```
move scratchpad
scratchpad show
```

Ejemplo:

```
Make the currently focused window a scratchpad
bindsym $mod+Shift+minus move scratchpad

# Show the first scratchpad window
bindsym $mod+minus scratchpad show

# Show the sup-mail scratchpad window, if any.
bindsym mod4+s [title="^Sup :"] scratchpad show
```

6.21. Nop

Hay un comando de operación (`nop`) que le permite anular el comportamiento predeterminado. Esto puede ser útil para, por ejemplo, deshabilitar un cambio de foco en los clics con el botón central del mouse.

El argumento opcional `comment` se ignora, pero se imprimirá en el archivo de registro para fines de depuración.

Sintáxis:

```
nop [<comment>]
```

Ejemplo:

```
# Disable focus change for clicks on titlebars
# with the middle mouse button
bindsym button2 nop
```

6.22. Control de i3bar

Hay dos opciones en la configuración de cada instancia de `i3bar` que se pueden cambiar durante el tiempo de ejecución invocando un comando a través de `i3`. Los comandos `bar hidden_state` y `bar mode` permiten configurar la opción actual de `modo hidden_state` de cada barra respectivamente. También es posible alternar entre ocultar el estado y mostrar el estado, así como entre el modo de acoplamiento y el modo oculto. Cada instancia de `i3bar` se puede controlar individualmente especificando un `bar_id`, si no se proporciona ninguno, el comando se ejecuta para todas las instancias de barra.

Sintáxis:

```
bar hidden_state hide|show|toggle [<bar_id>]
bar mode dock|hide|invisible|toggle [<bar_id>]
```

Ejemplo:

```
# Toggle between hide state and show state
bindsym $mod+m bar hidden_state toggle

# Toggle between dock mode and hide mode
bindsym $mod+n bar mode toggle

# Set the bar instance with id 'bar-1' to switch to hide mode
bindsym $mod+b bar mode hide bar-1

# Set the bar instance with id 'bar-1' to always stay hidden
bindsym $mod+Shift+b bar mode invisible bar-1
```

7. Múltiples monitores

Esta sección explicará cómo manejar múltiples monitores.

Cuando solo tiene un monitor, las cosas son simples. Por lo general, comienza con el espacio de trabajo 1 en su monitor y abre otros nuevos cuando los necesita.

Cuando tiene más de un monitor, cada monitor obtendrá un espacio de trabajo inicial. El primer monitor obtiene 1, el segundo obtiene 2 y un posible tercero obtendría 3. Cuando cambie a un espacio de trabajo en un monitor diferente, `i3` cambiará a ese monitor y luego cambiará al espacio de trabajo. De esta manera, no necesita atajos para cambiar a un monitor específico, y no necesita recordar dónde coloca qué espacio de trabajo. Se abrirán nuevos espacios de trabajo en el monitor actualmente activo. No es posible tener un monitor sin un espacio de trabajo.

La idea de hacer que los espacios de trabajo sean globales se basa en la observación de que la mayoría de los usuarios tienen un conjunto muy limitado de espacios de trabajo en sus monitores adicionales. A menudo se usan para una tarea específica (navegador, shell) o para monitorear varias cosas (correo, IRC, syslog, ...). Por lo tanto, usar un espacio de trabajo en un monitor y "el resto" en los otros monitores a menudo tiene sentido. Sin embargo, como puede crear un número ilimitado de espacios de trabajo en `i3` y vincularlos a pantallas específicas, puede tener el enfoque "tradicional" de tener `X` espacios de trabajo por pantalla cambiando su configuración (usando modos, por ejemplo).

7.1. Configurando sus monitores

Para ayudarlo a ponerse en marcha si nunca antes ha usado múltiples monitores, aquí hay una breve descripción de las opciones de `xrandr` que probablemente le interesarán. Siempre es útil obtener una visión general de la configuración actual de la pantalla. Simplemente ejecute "`xrandr`" y obtendrá una salida como la siguiente:

```
$ xrandr
Screen 0: minimum 320 x 200, current 1280 x 800, maximum 8192 x 8192
VGA1 disconnected (normal left inverted right x axis y axis)
LVDS1 connected 1280x800+0+0 (normal left inverted right x axis y axis) 261mm x 163mm
    1280x800    60.0*+   50.0
```

1024x768	85.0	75.0	70.1	60.0	
832x624	74.6				
800x600	85.1	72.2	75.0	60.3	56.2
640x480	85.0	72.8	75.0	59.9	
720x400	85.0				
640x400	85.1				
640x350	85.1				

Aquí hay varias cosas importantes: puede ver que `LVDS1` está conectado (por supuesto, es la pantalla plana interna) pero `VGA1` no. Si tiene un monitor conectado a uno de los puertos pero `xrandr` todavía dice "desconectado", debe verificar su cable, monitor o controlador de gráficos.

La resolución máxima que puede ver al final de la primera línea es la resolución combinada máxima de sus monitores. Por defecto, suele ser demasiado bajo y debe aumentarse editando `/etc/X11/xorg.conf`.

Entonces, supongamos que conectó `VGA1` y desea usarlo como una pantalla adicional:

```
xrandr --output VGA1 --auto --left-of LVDS1
```

Este comando hace que `xrandr` intente encontrar la resolución nativa del dispositivo conectado a `VGA1` y la configura a la izquierda de su panel plano interno. Al ejecutar "`xrandr`" nuevamente, el resultado se ve así:

```
$ xrandr
Screen 0: minimum 320 x 200, current 2560 x 1024, maximum 8192 x 8192
VGA1 connected 1280x1024+0+0 (normal left inverted right x axis y axis) 338mm x 270mm
 1280x1024    60.0*+   75.0
 1280x960     60.0
 1152x864     75.0
 1024x768     75.1    70.1    60.0
 832x624      74.6
 800x600      72.2    75.0    60.3    56.2
 640x480      72.8    75.0    66.7    60.0
 720x400      70.1
LVDS1 connected 1280x800+1280+0 (normal left inverted right x axis y axis) 261mm x 163mm
 1280x800     60.0*+   50.0
 1024x768     85.0    75.0    70.1    60.0
 832x624      74.6
 800x600      85.1    72.2    75.0    60.3    56.2
 640x480      85.0    72.8    75.0    59.9
 720x400      85.0
 640x400      85.1
 640x350      85.1
```

Tenga en cuenta que `i3` usa exactamente la misma `API` que `xrandr`, por lo que solo verá lo que puede ver en `xrandr`.

Consulte también [presentaciones] para obtener más ejemplos de configuraciones de monitores múltiples.

7.2. Configuración interesante para entornos de múltiples monitores.

Hay varias cosas para configurar en i3 que pueden ser interesantes si tiene más de un monitor:

1. Puede especificar qué espacio de trabajo se debe colocar en qué pantalla. Esto le permite tener un conjunto diferente de espacios de trabajo al iniciar que solo 1 para el primer monitor, 2 para el segundo y así sucesivamente. Ver [espacios de trabajo en pantallas específicas].
2. Si desea que algunas aplicaciones se abran generalmente en la pantalla más grande (MPlayer, Firefox, ...), puede asignarlas a un espacio de trabajo específico, consulte [clientes en espacios de trabajo específicos].
3. Si tiene muchos espacios de trabajo en muchos monitores, puede ser difícil hacer un seguimiento de qué ventana coloca dónde. Por lo tanto, puede usar marcas similares a vim para cambiar rápidamente entre ventanas. Ver [Marcas similares a VIM].

Para obtener información sobre cómo mover espacios de trabajo existentes entre monitores, consulte [Mover contenedores/espacios de trabajo a salidas RandR].

8. i3 y el resto de su mundo de software

8.1. Mostrar una línea de estado

Una cosa muy común entre los usuarios de gestores de ventanas exóticos es una línea de estado en algún rincón de la pantalla. Es un reemplazo a menudo superior al enfoque de widget que tiene en la barra de tareas de un entorno de escritorio tradicional.

Si aún no tiene su forma favorita de generar una línea de estado de este tipo (secuencias de comandos autoescritas, `conky`, ...), `i3status` es la herramienta recomendada para esta tarea. Fue escrito en C con el objetivo de utilizar la menor cantidad posible de llamadas al sistema para reducir el tiempo en que su CPU se despierta de los estados de suspensión. Debido a que `i3status` solo escupe texto, debe combinarlo con alguna otra herramienta, como `i3bar`. Consulte [Comando statusline] para ver cómo mostrar `i3status` en `i3bar`.

Independientemente de la aplicación que use para mostrar la línea de estado, debe asegurarse de que se registre como una ventana de acoplamiento mediante sugerencias EWMH. i3 colocará la ventana en la parte superior o inferior de la pantalla, según el consejo que establezca la aplicación. Con `i3bar`, puede configurar su posición, consulte [Posición `i3bar`].

8.2. Dar presentaciones (multimonitor)

Al dar una presentación, normalmente desea que la audiencia vea lo que ve en su pantalla y luego revise una serie de diapositivas (si la presentación es simple). Para presentaciones más complejas, es posible que desee tener algunas notas que solo usted puede ver en su pantalla, mientras que la audiencia solo puede ver las diapositivas.

8.2.1. Caso 1: todos obtienen la misma salida

Este es el caso simple. Conecta su computadora al proyector de video, enciende ambos (computadora y proyector de video) y configura su servidor X para clonar la pantalla plana interna de su computadora a la salida de video:

```
xrandr --output VGA1 --mode 1024x768 --same-as LVDS1
```

i3 utilizará el subconjunto común más bajo de resoluciones de pantalla, el resto de la pantalla quedará intacta (mostrará el fondo X). Entonces, en nuestro ejemplo, esto sería 1024x768 (mi computadora portátil tiene 1280x800).

8.2.2. Caso 2: puedes ver más que tu audiencia

Este caso es un poco más difícil. En primer lugar, debe configurar la salida VGA en algún lugar cerca de su pantalla plana interna, digamos a la derecha:

```
xrandr --output VGA1 --mode 1024x768 --right-of LVDS1
```

Ahora, i3 colocará un nuevo espacio de trabajo (dependiendo de su configuración) en la nueva pantalla y usted estará en modo de monitores múltiples (consulte [Múltiples monitores]).

Debido a que i3 no es un administrador de ventanas de composición, no existe la posibilidad de mostrar una ventana en dos pantallas al mismo tiempo. En cambio, su software de presentación necesita hacer este trabajo (es decir, abrir una ventana en cada pantalla).

8.3. Pantallas de alta resolución (también conocidas como pantallas HIDPI)

Consulte <https://wiki.archlinux.org/index.php/HiDPI> para obtener detalles sobre cómo habilitar el escalado en varias partes del escritorio de Linux. i3 leerá el DPI deseado de la propiedad `Xft.dpi`. El valor predeterminado de la propiedad es 96 DPI, por lo que para lograr una escala del 200%, debe configurar `Xft.dpi:192` en `~/.Xresources`.

Si es un usuario de i3 desde hace mucho tiempo que acaba de obtener un nuevo monitor, verifique que:

- Está utilizando una fuente escalable (que comienza con "pango:") en su configuración de i3.
- Está utilizando un emulador de terminal que admite el escalado. Puede cambiar temporalmente a `gnome-terminal`, que es conocido por admitir el escalado de fábrica, hasta que descubra cómo ajustar el tamaño de fuente en su emulador de terminal favorito.

9. Configuración por defecto

```
# WARNING
```

```
# WARNING: This configuration file is a template for the i3-config-wizard
to
# WARNING: generate a config which uses keysyms in your current layout. It
does
# WARNING: not get loaded by i3. Please do not change it.
# WARNING
```

```
# i3 config file (v4)
#
# Please see https://i3wm.org/docs/userguide.html for a complete
reference!
```

```
set $mod Mod1
```

```
# Font for window titles. Will also be used by the bar unless a different
font
# is used in the bar {} block below.
font pango:monospace 8
```

```
# This font is widely installed, provides lots of unicode glyphs, right-
to-left
# text rendering and scalability on retina/hidpi displays (thanks to
pango).
#font pango:DejaVu Sans Mono 8
```

```
# The combination of xss-lock, nm-applet and pactl is a popular choice, so
# they are included here as an example. Modify as you see fit.
```

```
# xss-lock grabs a logind suspend inhibit lock and will use i3lock to lock
the
# screen before suspend. Use loginctl lock-session to lock your screen.
exec --no-startup-id xss-lock --transfer-sleep-lock -- i3lock --nofork
```

```
# NetworkManager is the most popular way to manage wireless networks on
Linux,
# and nm-applet is a desktop environment-independent system tray GUI for
it.
exec --no-startup-id nm-applet
```

```

# Use pactl to adjust volume in PulseAudio.
set $refresh_i3status killall -SIGUSR1 i3status
bindsym XF86AudioRaiseVolume exec --no-startup-id pactl set-sink-volume
@DEFAULT_SINK@ +10% && $refresh_i3status
bindsym XF86AudioLowerVolume exec --no-startup-id pactl set-sink-volume
@DEFAULT_SINK@ -10% && $refresh_i3status
bindsym XF86AudioMute exec --no-startup-id pactl set-sink-mute
@DEFAULT_SINK@ toggle && $refresh_i3status
bindsym XF86AudioMicMute exec --no-startup-id pactl set-source-mute
@DEFAULT_SOURCE@ toggle && $refresh_i3status

# Use Mouse+$mod to drag floating windows to their wanted position
floating_modifier $mod

# start a terminal
bindcode $mod+36 exec i3-sensible-terminal

# kill focused window
bindcode $mod+Shift+24 kill

# start dmenu (a program launcher)
bindcode $mod+40 exec dmenu_run
# There also is the (new) i3-dmenu-desktop which only displays
applications
# shipping a .desktop file. It is a wrapper around dmenu, so you need that
# installed.
# bindsym $mod+d exec --no-startup-id i3-dmenu-desktop

# change focus
bindcode $mod+44 focus left
bindcode $mod+45 focus down
bindcode $mod+46 focus up
bindcode $mod+47 focus right

# alternatively, you can use the cursor keys:
bindcode $mod+113 focus left
bindcode $mod+116 focus down
bindcode $mod+111 focus up
bindcode $mod+114 focus right

# move focused window
bindcode $mod+Shift+44 move left
bindcode $mod+Shift+45 move down
bindcode $mod+Shift+46 move up

```

```

bindcode $mod+Shift+47 move right

# alternatively, you can use the cursor keys:
bindcode $mod+Shift+113 move left
bindcode $mod+Shift+116 move down
bindcode $mod+Shift+111 move up
bindcode $mod+Shift+114 move right

# split in horizontal orientation
bindcode $mod+43 split h

# split in vertical orientation
bindcode $mod+55 split v

# enter fullscreen mode for the focused container
bindcode $mod+41 fullscreen toggle

# change container layout (stacked, tabbed, toggle split)
bindcode $mod+39 layout stacking
bindcode $mod+25 layout tabbed
bindcode $mod+26 layout toggle split

# toggle tiling / floating
bindcode $mod+Shift+65 floating toggle

# change focus between tiling / floating windows
bindcode $mod+65 focus mode_toggle

# focus the parent container
bindcode $mod+38 focus parent

# focus the child container
#bindsym $mod+d focus child

# Define names for default workspaces for which we configure key bindings
later on.
# We use variables to avoid repeating the names in multiple places.
set $ws1 "1"
set $ws2 "2"
set $ws3 "3"
set $ws4 "4"
set $ws5 "5"
set $ws6 "6"
set $ws7 "7"

```

```

set $ws8 "8"
set $ws9 "9"
set $ws10 "10"

# switch to workspace
bindcode $mod+10 workspace number $ws1
bindcode $mod+11 workspace number $ws2
bindcode $mod+12 workspace number $ws3
bindcode $mod+13 workspace number $ws4
bindcode $mod+14 workspace number $ws5
bindcode $mod+15 workspace number $ws6
bindcode $mod+16 workspace number $ws7
bindcode $mod+17 workspace number $ws8
bindcode $mod+18 workspace number $ws9
bindcode $mod+19 workspace number $ws10

# move focused container to workspace
bindcode $mod+Shift+10 move container to workspace number $ws1
bindcode $mod+Shift+11 move container to workspace number $ws2
bindcode $mod+Shift+12 move container to workspace number $ws3
bindcode $mod+Shift+13 move container to workspace number $ws4
bindcode $mod+Shift+14 move container to workspace number $ws5
bindcode $mod+Shift+15 move container to workspace number $ws6
bindcode $mod+Shift+16 move container to workspace number $ws7
bindcode $mod+Shift+17 move container to workspace number $ws8
bindcode $mod+Shift+18 move container to workspace number $ws9
bindcode $mod+Shift+19 move container to workspace number $ws10

# reload the configuration file
bindcode $mod+Shift+54 reload
# restart i3 inplace (preserves your layout/session, can be used to
upgrade i3)
bindcode $mod+Shift+27 restart
# exit i3 (logs you out of your X session)
bindcode $mod+Shift+26 exec "i3-nagbar -t warning -m 'You pressed the exit
shortcut. Do you really want to exit i3? This will end your X session.' -B
'Yes, exit i3' 'i3-msg exit'"

# resize window (you can also use the mouse for that)
mode "resize" {
# These bindings trigger as soon as you enter the resize mode

# Pressing left will shrink the window's width.
# Pressing right will grow the window's width.

```

```

# Pressing up will shrink the window's height.
# Pressing down will grow the window's height.
bindcode 44 resize shrink width 10 px or 10 ppt
bindcode 45 resize grow height 10 px or 10 ppt
bindcode 46 resize shrink height 10 px or 10 ppt
bindcode 47 resize grow width 10 px or 10 ppt

# same bindings, but for the arrow keys
bindcode 113 resize shrink width 10 px or 10 ppt
bindcode 116 resize grow height 10 px or 10 ppt
bindcode 111 resize shrink height 10 px or 10 ppt
bindcode 114 resize grow width 10 px or 10 ppt

# back to normal: Enter or Escape or $mod+r
bindcode 36 mode "default"
bindcode 9 mode "default"
bindcode $mod+27 mode "default"
}

bindcode $mod+27 mode "resize"

# Start i3bar to display a workspace bar (plus the system information
i3status
# finds out, if available)
bar {
status_command i3status
}

```

10. La situación de múltiples monitores

Michael Stapelberg
 <michael@i3wm.org>
 April 2013

10.1. La solución rápida

Si está utilizando el controlador de gráficos binarios [nVidia](#) (también conocido como blob) antes de la versión [302.17](#), debe usar el indicador `--force-xinerama` (en su `.xsession`) al iniciar `i3`, así:

Ejemplo: `exec i3 --force-xinerama -V >>~/i3/i3log 2>&1`

O use: `force_xinerama yes` en su archivo de configuración.

10.2. La explicación

A partir de la versión 3.ε, i3 usa la API RandR (Rotar y cambiar el tamaño) en lugar de Xinerama. La razón de esto es que RandR proporciona más información sobre sus salidas y pantallas conectadas que Xinerama. Para ser específicos, el código que manejó la reconfiguración de la pantalla sobre la marcha (es decir, sin reiniciar el servidor X) fue una heurística muy desordenada y la mayoría de las veces no funcionó correctamente, eso no es posible con la poca información que ofrece Xinerama (solo una lista de resoluciones de pantalla, sin identificadores para las pantallas o cualquier información adicional). Xinerama simplemente no fue diseñado para una configuración dinámica.

Entonces apareció RandR, como una alternativa más poderosa (RandR 1.2 para ser específico). Ofrece todas las posibilidades de Xinerama y mucho más. El uso de la API RandR hizo que nuestro código fuera mucho más robusto y limpio. Además, ahora puede asignar espacios de trabajo de manera confiable a nombres de salida en lugar de algún identificador de pantalla poco confiable (posición dentro de la lista de pantallas, que podría cambiar, etc.).

Como RandR ha existido durante aproximadamente tres años a partir de este escrito, nos pareció una muy buena idea, y todavía es muy buena. Sin embargo, lo que no esperábamos era el controlador binario nVidia. Todavía no es compatible con RandR (a partir de marzo de 2010), a pesar de que nVidia ha anunciado que eventualmente será compatible con RandR. ¿Qué significa esto para usted, si está atrapado con el controlador binario por alguna razón (por ejemplo, los controladores gratuitos no funcionan con su tarjeta)? En primer lugar, está atascado con TwinView y no puede usar xrandr. Si bien esto arruina la experiencia del usuario, el problema más grave es que el controlador nVidia no solo no admite la configuración dinámica con RandR, sino que tampoco expone la información correcta de varios monitores a través de la API RandR. Entonces, en algunas configuraciones, i3 no encontrará ninguna pantalla; en otros, encontrará una pantalla grande que en realidad contiene ambas pantallas físicas (pero no sabrá que se trata de dos pantallas).

Por esta misma razón, decidimos implementar la siguiente solución alternativa: siempre que el controlador nVidia no sea compatible con RandR, una opción llamada `--force-xinerama` está disponible en i3 (como alternativa, puede usar la directiva del archivo de configuración `force_xinerama`). Esta opción obtiene la lista de pantallas una vez al iniciar, y nunca la actualiza. Como el controlador nVidia no puede hacer una configuración dinámica de todos modos, esto no es un gran problema.

También tenga en cuenta que sus nombres de salida no son descriptivos (como HDMI1) cuando usa Xinerama, sino que se cuentan, comenzando en 0: `xinerama-0`, `xinerama-1`, ...

11. Ver también

Para obtener más información sobre cómo usar configuraciones de monitores múltiples, consulte la [Guía del Usuario] de i3.