

Configuración inicial de i3wm en Arch Linux

Javier Niño Sánchez

14 de abril de 2026

Guía de referencia y recopilación de métodos para conseguir un sistema inicial razonable en Arch Linux, partiendo desde la TTY hasta un escritorio minimalista con i3wm (Xorg) o sway (Wayland). Incluye ejemplos de configuración, diagramas conceptuales y buenas prácticas.

Índice general

1. Introducción	6
1.1. Objetivo	6
1.2. Motivación	6
1.3. Convenciones	6
2. Preparación desde la TTY	8
2.0.1. ¿Qué es un TTY?	8
2.1. Instalación mínima	8
2.1.1. Software para i3wm y sway	9
2.1.2. Software para i3wm	9
2.2. Gestor de paquetes y reflector	11
3. Stack de red	12
3.1. iwctl (iwd)	12
3.1.1. Introducción a iwd	12
3.1.2. Configuración de eduroam	13
3.2. NetworkManager (alternativa)	13
3.3. Configuración de DNS / resolv.conf	14
4. Stack gráfico: i3wm (Xorg)	16
4.1. Diferencias conceptuales: Xorg vs Wayland	16
4.1.1. Qué son los servidores gráficos	16
4.1.2. ¿Qué servidor gráfico es recomendable utilizar?	17
4.2. i3wm (Xorg)	17
4.2.1. Estructura de <code>/.config/i3/config</code>	17
4.2.2. Compositores y picom	17
4.2.3. Rofi, el good for all de i3wm	19
4.3. Recursos gráficos compartidos	23
4.3.1. Fuentes	23
4.3.2. Iconos	23
4.3.3. Temas	23
5. Polkit, permisos y pulseaudio/pipewire	24
5.1. polkit: por qué y ejemplos de reglas simples	24
5.2. Audio: PipeWire como backend moderno	24
5.2.1. Integración con herramientas de PulseAudio	24
5.2.2. Apps útiles	24
6. Ricing y estética	25
6.1. <code>.Xresources</code> , themes y <code>notify_send</code>	25
6.2. Wallpaper, bar (i3bar, waybar), y launchers (rofi)	25
6.3. Ejemplos de configuración para picom, waybar y rofi	25
7. Herramientas extra y flujo de trabajo	26
7.1. Terminales recomendadas y atajos	26

7.2. Programas de usuario: gestor de archivos, navegador, terminal, compositor de texto	26
8. Solución de problemas y checklist	27
8.1. Logs relevantes (journalctl, Xorg.0.log, sway logs)	27
8.2. Problemas frecuentes	27
A. Cheat sheets	28
B. Ejemplo de diagrama conceptual	29

Índice de figuras

3.1. Flujo desde el servidor DHCP hasta DNS	14
B.1. Flujo general desde TTY hasta entorno gráfico	29

1. Introducción

1.1. Objetivo

Este manual pretende recopilar el proceso de configuración de un sistema basado en GNU/Linux[1] que utiliza gestores de ventanas manuales como i3[2] y sway[3]. Se ofrecen soluciones a las posibles necesidades de los usuarios de dichos gestores de ventana. Este manual existe con el objetivo de centralizar la documentación para su futura consulta.

Aunque aparece sway en la documentación, la versión actual del manual solo está diseñada para i3wm. En un futuro se incluirá la versión para sway. Por suerte, como sway ha sido diseñado como un drop-in debería de resultar sencillo al lector adaptar buena parte del manual para sway.

La distribución utilizada en el manual es Arch Linux, esto se debe a su extensa documentación, instalación base minimalista y el software disponible a través del gestor de paquetes[4]¹ pacman[5].

Se busca demostrar que no es necesario tener un entorno de escritorio para tener un sistema perfectamente funcional, y que, por lo tanto, no es necesario depender exclusivamente en un suite de software que en un futuro puede cambiar a un estado no deseado.

1.2. Motivación

Este manual pretende compartir conocimientos relacionados con el funcionamiento de Linux, así como ofrecer una guía básica que pueda ser utilizada para empezar a configurar un sistema GNU/Linux.

Esta guía pretende ser un documento en el que se pueda ya no solo aprender a configurar i3wm o sway, sino que sea un resumen del funcionamiento del sistema operativo para futuro estudio.

1.3. Convenciones

En este manual, se establecerá `$XDG_CONFIG_HOME` en `$HOME/.config`, de forma que la mayoría de la configuración se realizará en este directorio. A su vez, se recomienda tener un directorio auxiliar (`$HOME/.i3`) para guardar posibles configuraciones extra y scripts utilizados para i3wm.

¹Un gestor de paquetes es software utilizado, generalmente en distribuciones para facilitar la instalación, gestión y eliminación de software de manera controlada

Por regla general se utilizará BASH (Bourne Again Shell) para los comandos utilizados, así como lenguaje de scripting. Para los directorios, se usará como base \$HOME, usando como locale es_ES-utf8

Se asume que el usuario no ha instalado ningún entorno de escritorio, pero este manual se puede seguir perfectamente aún si se ha instalado algún entorno.

El nombre de los paquetes a instalar vendrá monoespaciado, a su vez, si los paquetes deben de ser instalados desde la AUR se usará un superíndice como el que sigue paquete^{AUR}

2. Preparación desde la TTY

2.0.1. ¿Qué es un TTY?

Un TTY[6]¹ en GNU/Linux supone un emulador de terminal² en el cual se pueden ejecutar comandos para interactuar con el kernel y los recursos del hardware. Usualmente, en sistemas Linux habrá 7 TTYs virtuales, accesibles mediante `Ctrl`+`Alt`+`F1` hasta `Ctrl`+`Alt`+`F6`.

Por regla general, las distribuciones usarán la primera TTY para el login y la última para el entorno gráfico, pero esto depende de cada distribución y de como se gestione la interfaz gráfica.

Cada TTY tiene sus propios streams de entrada y salida, por lo que permite usar múltiples terminales de manera independiente. En nuestro caso, la TTY actuará como terminal principal para la instalación inicial, no nos preocuparemos porque la interfaz gráfica esté en la última TTY o en la primera, por regla general, el greeter se encargará de gestionar estos asuntos.

2.1. Instalación mínima

Hay que descargar múltiples paquetes para que el sistema pueda funcionar correctamente, nos basaremos en la guía de instalación de Arch Linux[7], que incluye los siguientes paquetes:

- `amd-ucode` o `intel-ucode`: Microcode para la CPU del sistema. **Instalar solo uno.**
- `linux-firmware`, `sof-firmware`: Instalacion del firmware necesario, `linux-firmware` es firmware general que podría ser necesario para `mkinitcpio` y otras utilidades del sistema, `sof-firmware` es firmware dedicado para audio. **Instalar ambos.**
- `iw` o `NetworkManager`: Gestores del stack de red desde el espacio de usuario. Se pueden utilizar software como `wpa_supplicant`, pero para en el manual se usará `iw` o `NetworkManager`. Se recomienda **instalar solo uno**, puede haber conflictos si ambos están funcionando en el sistema al mismo tiempo. Se priorizará `iw`, ya que no hay buenos argumentos para utilizar `NetworkManager` (que puede usar `iw` como backend) si solo se quiere gestionar conexiones Wi-Fi.
- `nano`, `vim`, `nvim` y `emacs`: Editores de texto destacados. **Se recomienda instalar nano por lo menos.**
- `man`: Documentación para sistemas POSIX.

¹TeleTypewriter

²Se entiende por emulador de terminal todo software que intente simular el comportamiento de las terminales originales que tenían los ordenadores antiguamente

Aunque no es mencionado directamente en la guía, se recomienda instalar un polkit[8] para la gestión de sesiones. Observando las opciones resulta evidente que habrá que tener cuidado con la elección que hagamos. Nuestro objetivo es encontrar un polkit ligero y que sea compatible tanto con X11 y sway, trayendo consigo el menor número de dependencias posible.

En base a estos criterios (aunque se recomienda experimentar con otros polkit) se usará polkit-gnome.

También hay que instalar software adicional dependiendo del gestor de ventanas que utilicemos. A continuación se detallará el software que se necesita o se recomienda instalar.

2.1.1. Software para i3wm y sway

- Alacritty: Emulador de terminal³ recomendado.
- htop, nvtop, btop: TUIs⁴ para gestión de recursos. htop es sobre todo para procesador, nvtop es para GPUs y btop es para el sistema en general.
- brightnessctl: CLI para la gestión del brillo de las pantallas. Muy útil para scripts.
- impala: TUI para gestión de conexiones Wi-Fi.
- bluetoothtui: TUI para gestión de dispositivos bluetooth.
- ranger: TUI file manager.
- yay: Gestor de paquetes de la AUR.
- fzf: Fuzzy finder para archivos del sistema. Extremadamente potente.

En base a esta lista, es fácil percatarse de que la mayoría de aplicaciones mencionadas no dependen del servidor gráfico que el usuario haya iniciado en su sesión actual. Esto se debe a que, para esta sección, se ha buscado maximizar la compatibilidad de los programas con los sistemas que se basan en i3wm (X11) y sway (wayland).

2.1.2. Software para i3wm

- i3: Paquete principal de i3wm. Incluye los paquetes i3-wm (window manager), i3blocks i3status (programas para escribir status en i3bar mediante stdout) y i3lock (screen locker).
- feh: Programa simple para visualizar imágenes, y establecer fondos de pantalla. Si se quiere una aplicación que se encargue exclusivamente de establecer fondos de pantalla, se puede utilizar nitrogen.
- rofi: Posiblemente de los mejores programas que se han creado para X11, es un program launcher que permite la creación de modos. Permite hacer de todo, en este manual se utilizará junto rofimoji (listado de emojis para poder utilizarlos de manera conveniente), rofi-calc (calculadora rápida). Pero también se puede usar como un power menu o como un gestor de portapapeles.

³Gestores de ventana como i3wm ya incluyen su propia terminal, pero es preferible usar otras

⁴Terminal User Interface, un programa cuya interfaz gráfica se renderiza en el terminal, normalmente usando librerías como ncurses.

- `maim`: Para capturas de pantalla. Se puede utilizar `scrot`, pero este último es bastante anticuado
- `xorg-xrandr`: Para gestión de resoluciones, tasas de refresco y posicionamiento de las pantallas.
- **Opcionales:**
 - `polybar`: Programa para reemplazar `i3bar` en el caso de que se quiera configurar a mayor profundidad el status-line.
 - `clipmenu`: Programa para gestión de historial del portapapeles.
 - `rofi-bluetoot`: Modo de `rofi` para gestión de dispositivos bluetooth conectados.

En esta sección no se han incluido los programas para el stack de red, esto se debe a que en el siguiente capítulo se tratará la configuración del stack de red según los paquetes que se utilicen.

2.2. Gestor de paquetes y reflector

El paquete `reflector` [9] es un script de python que se encarga de actualizar la lista de espejos utilizada para instalar paquetes usando `pacman`. Dicho paquete incluye un `.service` y un `.timer` que se pueden configurar. `Reflector` permite configurar la generación de la lista de espejos, modificando el archivo de configuración que se encuentra en `/etc/xdg/reflector/reflector.conf`.

Un ejemplo de configuración, en el que se ordenan por velocidad y se eligen solo los 10 mejores sería:

```

1  # Reflector configuration file for the systemd service.
2  #
3  # Empty lines and lines beginning with "#" are ignored. All other lines should
4  # contain valid reflector command-line arguments. The lines are parsed with
5  # Python's shlex modules so standard shell syntax should work. All arguments are
6  # collected into a single argument list.
7  #
8  # See "reflector --help" for details.
9
10 # Recommended Options
11
12 # Set the output path where the mirrorlist will be saved (--save).
13 --save /etc/pacman.d/mirrorlist
14
15 # Select the transfer protocol (--protocol).
16 --protocol https
17
18 # Select the country (--country).
19 # Consult the list of available countries with "reflector --list-countries" and
20 # select the countries nearest to you or the ones that you trust. For example:
21 # --country France,Germany
22 --country ES,PT,GB,DE,IT
23
24 # Use only the most recently synchronized mirrors (--latest).
25 --latest 15
26
27 # Sort the mirrors by synchronization time (--sort).
28 --sort rate

```

3. Stack de red

3.1. iwctl (iwd)

3.1.1. Introducción a iwd

`iwd`^[10] es un gestor de conexiones inalámbricas desarrollado por intel como sustituto de `wpa_supplicant`. El objetivo es ser más eficiente con los recursos a la hora de gestionar dichas conexiones.

`wpa_supplicant` es un programa para conexiones WPA, WPA2, WPA3¹ que actúa como suplicante, puede ser utilizado por programas como NetworkManager como backend al igual que `iwd`. La principal diferencia es que `iwd` intenta aprovechar lo máximo posible la funcionalidad ofrecida por el kernel, mientras mantiene reducido el número de dependencias.

Se puede usar como standalone (que es como se utilizará en esta guía), también se puede usar como backend para NetworkManager, systemd-networkd y otros.

Hay que tener en cuenta que este guión está pensado para utilizarse junto a systemd, así que la configuración que se recomienda utilizar estará basada en el uso de systemd-resolved para la configuración del DNS (más información en el punto 3.3)

Aunque en la Wiki de Arch se menciona, es recomendable instalar algún front-end como interfaz gráfica, en este caso se recomienda `impala`, una TUI moderna que funciona muy bien en cualquier entorno gráfico.

Aunque se use `impala` (o cualquier front-end gráfico que se quiera), es recomendable saber manejarse con `iwd` directamente. Por un lado se pueden ejecutar comandos mediante el modo interactivo:

```
1 $ iwctl #ejecutar modo interactivo
2 $ [iwd] device list #ver las interfaces del sistema, para el ejemplo se asume que
   → wlan0 es la interfaz de Wi-Fi
3 $ [iwd] station wlan0 scan #scanear redes Wi-Fi
4 $ [iwd] station wlan0 get-networks #devuelve una lista (que se puede seguir
   → actualizando si sigue scaneando) con las redes disponibles
5 $ [iwd] station wlan0 connect SSID #SSID es el nombre de la Wi-Fi, pedirá una
   → contraseña
6 $ [iwd] station wlan0 disconnect #desconectarse de la red
7 $ [iwd] station wlan0 show #muestra información de la interfaz
8 $ [iwd] help #Ayuda sobre comando extremadamente útil
```

Se puede hacer la conexión en un solo comando, pero es recomendable usar el modo interactivo si no se está usando un front-end.

¹En otras palabras, para conexiones Wi-Fi

3.1.2. Configuración de eduroam

La UGR ofrece un script de python que hace el trabajo de configurar eduroam automáticamente, sin necesidad de que el estudiante tenga que configurarlo manualmente. El principal problema de esta configuración es que depende de que se esté utilizando NetworkManager, ya que el script asume que el sistema está basado en algún entorno de escritorio (que por defecto suelen traer NetworkManager en la mayoría de entornos populares)

Debido a esto, el script fallará cuando se intente lanzar en sistemas basados en iwd, ya que busca la interfaz de NetworkManager. Una de las posibles soluciones es la configuración manual, la cual escribiremos en /var/lib/iwd, que es donde iwd espera encontrar las configuraciones de red. Para ello, escribiremos en el archivo /var/lib/iwd/eduroam.8021x

```

1  [Security]
2  EAP-Method=PEAP
3  EAP-Identity=anonymous2022@ugr.es
4  EAP-PEAP-CA Cert=/var/lib/iwd/CA-UGR-eduroam.pem
5  EAP-PEAP-ServerDomainMask=*.ugr.es
6  EAP-PEAP-Phase2-Method=MSCHAPV2
7  EAP-PEAP-Phase2-Identity=estudianteejemplo@correo.ugr.es
8  EAP-PEAP-Phase2-Password=passwordejemplo
9
10
11 [Settings]
12 AutoConnect=true

```

Asegúrate de haber instalado el certificado de la UGR y que esté en /var/lib/iwd, es probable que tengas que convertir el certificado .cert a .pem como en el ejemplo. Para ello, puedes ejecutar el siguiente comando.

```

1  /var/lib/iwd $ openssl x509 -in CA-UGR-eduroam.crt -outform PEM -out
    ↪ CA-UGR-eduroam.pem

```

3.2. NetworkManager (alternativa)

La configuración de NetworkManager es bastante directa en comparación², una vez instalado el paquete bastaría con hacer lo siguiente (antes de lanzar el servicio):

```

1  $ sudo systemctl enable --now systemd-resolved #habilitar systemd-resolved
2  $ sudo ln -s /run/systemd/resolve/stub-resolv.conf /etc/resolv.conf #enlazar el
    ↪ stub a resolv.conf

```

Tras ello, antes de iniciar NetworkManager hay que actualizar su configuración en /etc/NetworkManager/NetworkManager.conf³ para incluir la siguiente línea:

²uno de sus puntos a favor en comparación con iwd

³Se puede crear un archivo en /etc/NetworkManager/conf.d/ en caso de que el usuario quiera separar la configuración de distintos aspectos de NetworkManager.

```

1 [main]
2 dns=systemd-resolved

```

Tras ello, sería cuestión de lanzar NetworkManager:

```

1 $ sudo systemctl enable --now NetworkManager #habilitar NetworkManager

```

Si por algún motivo NetworkManager ya estuviera funcionando, entonces es tan simple como reiniciar los servicios:

```

1 $ sudo systemctl restart systemd-resolved
2 $ sudo systemctl restart NetworkManager

```

Y para comprobar que está funcionando:

```

1 $ resolvectl status #resolvectl es el cli de systemd-resolved
2 $ ls -la /etc/resolv.conf #debería de devolver el enlace creado anteriormente

```

3.3. Configuración de DNS / resolv.conf

Hay muchas formas de configurar `/etc/resolv.conf` y muchos argumentos para usar un método u otro. Entre los métodos se incluye la configuración por NetworkManager, o mediante el uso de programas específicos basados en DHCP como `dhclient` (esta clase de solución es la más coherente con la filosofía UNIX⁴).

Si el lector ha optado por utilizar NetworkManager, entonces no necesitará preocuparse de popular `/etc/resolv.conf`, puesto que en su configuración por defecto este gestor de conexiones ya edita dicho archivo (normalmente a través de `systemd-resolved`). En ese caso, el flujo sería:

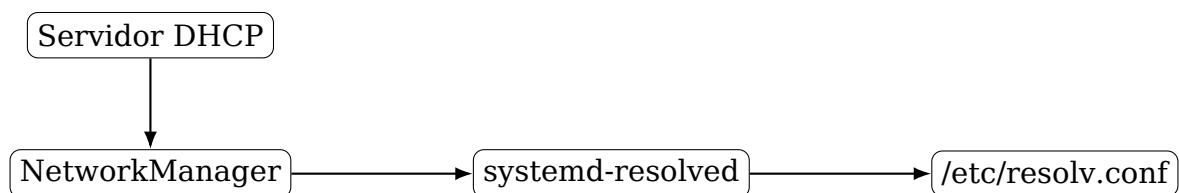


Figura 3.1.: Flujo desde el servidor DHCP hasta DNS

Aunque lo que queda de sección concierne a aquellos usuarios que hayan optado por usar `iw`, se ha dejado como una sección a parte en el caso en el que los usuarios de NetworkManager quieran tener varios métodos para completar `/etc/resolv.conf`, en cuyo caso se aconseja instalar `openresolv` para la sincronización de escrituras sobre el archivo.

En este caso aprovecharemos que `iw` tiene la capacidad de interactuar con el servidor DHCP y usaremos `systemd-resolved` para acabar con un flujo parecido al

⁴“Do one thing and do it well”, el software debería de ser minimalista y modular.

que tiene NetworkManager.

Para ello es necesario habilitar systemd-resolved, esto creará un stub que escuchará en el puerto 53 (puerto para DNS), de esta manera cuando se quiera consultar un dominio, resolv.conf apuntará a dicho puerto y systemd-resolved hará el resto del trabajo. En una terminal:

```
1 $ sudo systemctl enable --now systemd-resolved #habilitar systemd-resolved
2 $ sudo ln -s /run/systemd/resolve/stub-resolv.conf /etc/resolv.conf #enlazar el
   ↪ stub a resolv.conf
```

Si observamos /etc/resolv.conf, debería de aparecer algo como lo que sigue:

```
1 # This is /run/systemd/resolve/stub-resolv.conf managed by man:systemd-resolved(8).
2 # Do not edit.
3 #
4 # This file might be symlinked as /etc/resolv.conf. If you're looking at
5 # /etc/resolv.conf and seeing this text, you have followed the symlink.
6 #
7 # This is a dynamic resolv.conf file for connecting local clients to the
8 # internal DNS stub resolver of systemd-resolved. This file lists all
9 # configured search domains.
10 #
11 # Run "resolvectl status" to see details about the uplink DNS servers
12 # currently in use.
13 #
14 # Third party programs should typically not access this file directly, but only
15 # through the symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a
16 # different way, replace this symlink by a static file or a different symlink.
17 #
18 # See man:systemd-resolved.service(8) for details about the supported modes of
19 # operation for /etc/resolv.conf.
20
21 nameserver 127.0.0.53
22 options edns0 trust-ad
23 search .
```

Lo único que faltaría sería configurar iwd para que utilice systemd-resolved, en /etc/iwd/main.conf hay que escribir:

```
1 [General]
2 EnableNetworkConfiguration=true
3
4 [Network]
5 NameResolvingService=systemd
```

4. Stack gráfico: i3wm (Xorg)

4.1. Diferencias conceptuales: Xorg vs Wayland

El objetivo de este manual es ofrecer una guía para configurar gestores de ventana como i3wm o sway, pero es de interés entender como funcionan los servidores gráficos que permiten el funcionamiento de las interfaces de usuario.

4.1.1. Qué son los servidores gráficos

Un servidor gráfico es un programa que se encarga de gestionar recursos limitados relacionados con interfaces gráficas. En otras palabras, esta clase de programas son los responsables de la comunicación entre programas gráficos mientras gestiona los recursos¹ que a cada programa se le permite ocupar.

En la actualidad, esta clase de programas son muy complejos, pero en Linux hay principalmente dos: X11 y Wayland. X11[11] es la versión número 11 del protocolo X, la cual lleva siendo la versión principal por más de 40 años. Este protocolo tiene el problema de que su estructura está basada en entornos gráficos más simples, por lo que tiene peor soporte de sistemas multi monitor y/o con distintas resoluciones. A su vez, el protocolo tiene pocas comprobaciones de seguridad sobre la información a la que pueden acceder a los distintos procesos, lo cual, desde un punto de vista de scripting, es muy útil, pero a su vez reduce mucho la seguridad del sistema. El lado bueno es la estabilidad respecto al hardware, suele usarse como fallback en muchos casos y tiende a funcionar mejor con las gráficas de Nvidia.

Por otro lado está wayland[12], el cual es un proyecto diseñado para solucionar los problemas que presentaba X11 que tiene ya 17 años de desarrollo. Este protocolo carece de los problemas clásicos de X11,² en cambio, entre sus principales problemas se tiene que el entorno de desarrollo sigue siendo inmaduro,³ tiende a tener mayor inestabilidad (tanto por software como hardware).⁴

¹Espacio en la pantalla, superposición de programas, configuración de pantallas y framebuffers, DPI...

²el soporte multi monitor, sobre todo al hacer hot-plugging, es muy bueno. También mejora muchísimo la seguridad del sistema

³cabe recalcar que con cada año va mejorando, hay gran aceptación por parte de las distribuciones, pero sigue teniendo problemas en cuanto a software para automatización y scripting.

⁴Este es el principal problema/crítica que se puede hacer al proyecto, que se realizó como un "borrón y cuenta nueva", lo que implica que a lo largo de los 17 años se han encontrado con multitud de edge-cases que ya estaban solucionados en X11.

4.1.2. ¿Qué servidor gráfico es recomendable utilizar?

Este apartado está dedicado a aquellos lectores que consulten este manual sin tener una intención clara de qué servidor gráfico utilizar. Debido a que ya se han expuesto a grandes rasgos los beneficios e inconvenientes de cada modelo quedaría hacer una síntesis de los perfiles de usuario que podrían interesarse por un servidor u otro.

Se recomienda instalar X11 (y por lo tanto i3wm) al usuario si prioriza el control sobre su sistema, la compatibilidad con hardware (en caso de que wayland le de problemas), la extensa documentación que hay sobre el software y si se conforma con un sistema menos seguro.

Se recomienda instalar wayland (y por lo tanto sway) al usuario si prioriza la seguridad del sistema, un software que en la actualidad recibe mucho mantenimiento y mejoras, la buena documentación que hay sobre el software y si se conforma con un menor control del sistema, alguna incompatibilidad con el software (mayormente solucionado con xwayland) y no tiene problemas con su hardware.

4.2. i3wm (Xorg)

i3wm es un gestor de ventanas en mosaico tiling window manager⁵ para X11. La filosofía es la simplicidad, la configuración principal se hace en un único fichero de texto plano y el comportamiento del sistema está controlado por el teclado.

En secciones anteriores se ha cubierto el software a instalar para lograr una instalación mínima del sistema, así que en estas secciones se explicará la estructura básica de configuración y el software más relevante.

4.2.1. Estructura de `/.config/i3/config`

Como se ha comentado anteriormente, i3wm cuando inicia busca el archivo `/.config/i3/config` para obtener la configuración realizada por el usuario. Aunque se puede usar la keyword **include** para utilizar múltiples archivos.

Al iniciar i3wm por primera vez este archivo no existirá, por lo que i3wm invocará una ventana preguntando al usuario si quiere que genere la configuración por defecto. Para los usuarios con poca experiencia se les recomienda empezar con la configuración por defecto.

Dicho esto, se recomienda mirar la documentación para entender el funcionamiento del fichero de configuración[13]. Aún así, a continuación se incluyen ejemplos de configuración para demostrar el potencial de i3wm

4.2.2. Compositores y picom

Este apartado es opcional, no es necesario tener un compositor para que i3wm funcione. Pero es útil para añadir animaciones y para arreglar el posible screen tearing causado por la falta de uno.

⁵Es decir, basado en árboles de estado, siendo el usuario el que controla la distribución de las ventanas manualmente.

Un compositor[14] es software que contiene buffers de las ventanas gráficas y combina efectos visuales para generar frames. De esta manera, se puede añadir animaciones, solucionar problemas como screen tearing, aplicar transparencia a ventanas, entre otros.

Entre todos los compositors que hay disponibles, se recomienda utilizar `picom`, ya que es uno de lo más actualizados y eficientes[15]. El motivo por el que hay que instalar un compositor standalone surge por la filosofía minimalista de `i3wm`⁶. Este gestor de ventanas no asume nada sobre la configuración del sistema, por lo que el usuario tiene que tomar esta clase de decisiones por su cuenta.

`Picom` tiene buena documentación, se pueden encontrar comentarios explicando que hace cada cosa en el archivo de configuración que se puede encontrar en `/usr/share/doc/picom/picom/picom.conf.example`. También se pueden establecer reglas sobre opacidad, sombreado de ventanas, entre otros. Es recomendable copiar el este documento a `/.config/picom.conf`. Algunas de las posibles modificaciones a la configuración encontrada en el sistema son:

```

1 #####
2 # Transparency / Opacity #
3 #####
4
5 # Opacity of window titlebars and borders.
6 #
7 # Range: 0.1 - 1.0
8 # Default: 1.0 (disabled)
9 # Deshabilitamos opacidad general
10 frame-opacity = 1.0;
11
12 # añadimos opacidad por ventana
13 # estructura:
14 # "<% opacidad>:<atributo de ventana> = <valor atributo>"
15 opacity-rule = [
16     "95:class_g = 'Spotify'", #95% opacidad para Spotify
17 ];
18
19 #####
20 # Blur #
21 #####
22
23 # Parameters for background blurring, see BLUR section in the man page for more
24 ↪ information.
25 blur-method = "dual_kawase"
26 blur-strength = 3
27 # Specify the blur convolution kernel, with the following format:
28 # example:
29 # blur-kern = "5,5,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1";
30 # Can also be a pre-defined kernel, see the man page.
31 #
32 # Default: ""
33 # Este es bastante efectivo
34 blur-kern = "3x3box";

```

⁶Sway, aunque diseñado como un drop-in de `i3wm` para Wayland, tiene integrado un compositor, pero esto tiene que ver más con el funcionamiento de wayland que por la propia filosofía del gestor de ventanas

```

34
35 blur-background-exclude = [
36     "class_g = 'slop'",
37     "class_g = 'Polybar'"
38 ];
39
40
41 #####
42 #       General Settings       #
43 #####
44
45 # Daemonize process. Fork to background after initialization. Causes issues with
46 ↪ certain (badly-written) drivers.
47 # daemon = false
48
49 # Specify the backend to use: `xrender`, `glx`, or `egl`.
50 #
51 # Default: "xrender"
52 # glx usa OpenGL, que es mucho más eficiente que xrender (que usa X Render
53 ↪ extension)
54 # solo usar xrender si se encuentran problemas de estabilidad
55 backend = "glx"
56
57 # Use of damage information for rendering. This cause the only the part of the
58 # screen that has actually changed to be redrawn, instead of the whole screen
59 # every time. Should improve performance.
60 #
61 # Default: false
62 use-damage = true;

```

Cuidado al activar use-damage ya que algunos drivers no están bien implementados, causando artefactos.

Hay parte de la configuración que se podría optimizar en función del hardware que use cada uno, hasta cierto punto, los cambios propuestos son bastante generales y no suelen entrar en específicos. Se recomienda al lector el mirar la configuración directamente en internet por detalles[16].

4.2.3. Rofi, el good for all de i3wm

El paquete rofi[17] incluye un binario que es muy útil en i3wm. Es un programa muy popular entre gestores de ventana basados en X11 y está muy integrado en los paquetes de arch linux, permitiendo instalar muchas de sus extensiones usando pacman directamente, sin tener que compilar ni añadir nuevos repositorios a la base de datos del gestor de paquetes.

Aunque en las referencias se incluyen los paquetes extras que se pueden instalar, se considera conveniente incluir un listado a continuación.

- rofimoji: Integración básica de Unicode para caracteres y emojis. (Recomendable)
- rofi-emoji: Un selector de emojis integrado. (Extremadamente recomendable)

- `rofi-power-menu`^{AUR}: Permite usar rofi como un menú para apagar, suspender, reiniciar... Especialmente útil si se plantea utilizar rofi para esta clase de tareas. (Recomendable según el caso)
- `rofi-calc`: Permite realizar operaciones y conversiones entre unidades. (Extremadamente recomendable).

También se puede utilizar junto a algunos gestores de portapapeles como `clipmenu`, pero no suelen ser plug-and-play como todos los paquetes que hemos enseñado anteriormente.

Además de ser un desktop application launcher⁷, rofi permite la configuración, tanto de los modos⁸ de uso como de la GUI. Por lo tanto, a continuación se presentan dos posibles configuraciones, una general, y otra específica de tema.

El ejemplo de configuración es candidato de ser movido a un apéndice propio

Por un lado, la configuración general existe en el archivo `/.config/rofi/config.rasi`:

```

1 configuration {
2     modi: "run,drun>window,emoji,calc";
3     font: "Iosevka 18";
4     fixed-num-lines: true;
5     show-icons: true;
6     display-run: "Applications:";
7     display-window: "Windows:";
8     timeout {
9         action: "kb-cancel";
10        delay: 0;
11    }
12 }
13 @theme "~/.config/rofi/nord.rasi"
14
15 listview {
16     lines: 8;
17     columns: 2;
18 }
19
20 window{
21     width: 45%;
22 }
23
```

Por otro lado, y como se presenta en la configuración general, los temas pueden residir en cualquier parte del sistema, pero por convención se recomienda utilizar el directorio `/.config/rofi/` o `/.config/rofi/themes`:

```

1 /**
2  * Nordic rofi theme
3  * Adapted by undiabler <undiabler@gmail.com>
4  *
```

⁷En este caso, una subclase caracterizada por una interfaz minimalista, de control por teclado y diseñado para lanzar aplicaciones o ejecutar acciones rápidamente sin navegación gráfica tradicional.

⁸En términos de rofi, los modi

```
5  * Nord Color palette imported from https://www.nordtheme.com/
6  *
7  */
8
9
10 * {
11     nord0: #2e3440;
12     nord1: #3b4252;
13     nord2: #434c5e;
14     nord3: #4c566a;
15
16     nord4: #d8dee9;
17     nord5: #e5e9f0;
18     nord6: #eceff4;
19
20     nord7: #8fbcbb;
21     nord8: #88c0d0;
22     nord9: #81a1c1;
23     nord10: #5e81ac;
24     nord11: #bf616a;
25
26     nord12: #d08770;
27     nord13: #ebcb8b;
28     nord14: #a3be8c;
29     nord15: #b48ead;
30
31     foreground: @nord9;
32     backlight: #ccffeedd;
33     background-color: transparent;
34
35     highlight:      underline bold #eceff4;
36
37     transparent: rgba(46,52,64,0);
38 }
39
40 window {
41     location: center;
42     anchor: center;
43     transparency: "screenshot";
44     padding: 10px;
45     border: 0px;
46     border-radius: 6px;
47
48     background-color: @transparent;
49     spacing: 0;
50     children: [mainbox];
51     orientation: horizontal;
52 }
53
54 mainbox {
55     spacing: 0;
56     children: [ inputbar, message, listview ];
57 }
58
59 message {
60     color: @nord0;
61     padding: 5;
62     border-color: @foreground;
```

```

63     border: 0px 2px 2px 2px;
64     background-color: @nord7;
65 }
66
67 inputbar {
68     color: @nord6;
69     padding: 11px;
70     background-color: #3b4252;
71
72     border: 1px;
73     border-radius: 6px 6px 0px 0px;
74     border-color: @nord10;
75 }
76
77 entry, prompt, case-indicator {
78     text-font: inherit;
79     text-color: inherit;
80 }
81
82 prompt {
83     margin: 0px 1em 0em 0em ;
84 }
85
86 listview {
87     padding: 8px;
88     border-radius: 0px 0px 6px 6px;
89     border-color: @nord10;
90     border: 0px 1px 1px 1px;
91     background-color: rgba(46,52,64,0.9);
92     dynamic: false;
93 }
94
95 element {
96     padding: 3px;
97     vertical-align: 0.5;
98     border-radius: 4px;
99     background-color: transparent;
100    color: @foreground;
101    text-color: rgb(216, 222, 233);
102 }
103
104 element selected.normal {
105     background-color: @nord7;
106     text-color: #2e3440;
107 }
108
109 element-text, element-icon {
110     background-color: inherit;
111     text-color: inherit;
112 }
113
114 button {
115     padding: 6px;
116     color: @foreground;
117     horizontal-align: 0.5;
118
119     border: 2px 0px 2px 2px;
120     border-radius: 4px 0px 0px 4px;

```

```
121     border-color: @foreground;
122 }
123
124 button selected normal {
125     border: 2px 0px 2px 2px;
126     border-color: @foreground;
127 }
128
129 textbox {
130     padding: 8px;
131     border-radius: 6px 6px 6px 6px;
132     border-color: @nord10;
133     border: 1px 1px 1px 1px;
134     background-color: rgba(46,52,64,0.9);
135     dynamic: true;
136     text-color: @backlight;
137 }
```

En este ejemplo se ha usado el tema de nord, adaptado por undiabler. Pero se recomienda al usuario a crear el suyo propio a partir de uno existente con la ayuda de la documentación de rofi[18].

4.3. Recursos gráficos compartidos

Esta sección aún está en desarrollo, por suerte la documentación de arch es extensa, así que se recomienda buscar información ahí.

4.3.1. Fuentes

4.3.2. Iconos

4.3.3. Temas

5. Polkit, permisos y pulseaudio/pipewire

5.1. polkit: por qué y ejemplos de reglas simples

5.2. Audio: PipeWire como backend moderno

5.2.1. Integración con herramientas de PulseAudio

5.2.2. Apps útiles

6. Ricing y estética

6.1. .Xresources, themes y notify_send

6.2. Wallpaper, bar (i3bar, waybar), y launchers (rofi)

6.3. Ejemplos de configuración para picom, waybar y rofi

7. Herramientas extra y flujo de trabajo

7.1. Terminales recomendadas y atajos

7.2. Programas de usuario: gestor de archivos, navegador, terminal, compositor de texto

8. Solución de problemas y checklist

8.1. Logs relevantes (journalctl, Xorg.0.log, sway logs)

8.2. Problemas frecuentes

A. Cheat sheets

B. Ejemplo de diagrama conceptual

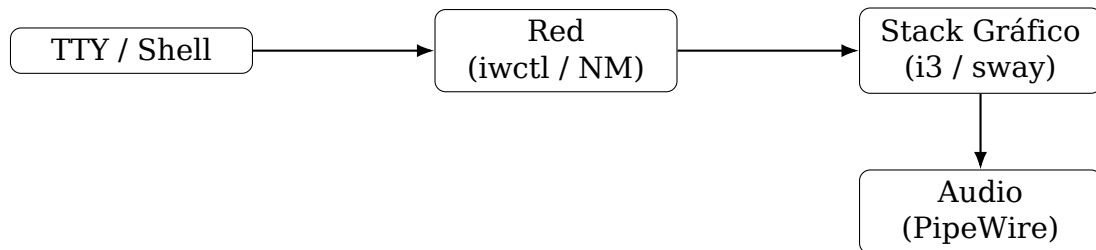


Figura B.1.: Flujo general desde TTY hasta entorno gráfico

Bibliografía

- [1] Wikipedia. *GNU/Linux*. 2026. URL: <https://en.wikipedia.org/wiki/Linux> (visitado 24-02-2026).
- [2] i3. *i3wm*. 2026. URL: <https://i3wm.org/> (visitado 24-02-2026).
- [3] sway. *swaywm*. 2026. URL: <https://swaywm.org/> (visitado 24-02-2026).
- [4] Wikipedia. *Package_manager*. Definición general de gestor de paquetes y su historia. 2026. URL: https://en.wikipedia.org/wiki/Package_manager (visitado 24-02-2026).
- [5] Arch Linux Wiki. *Pacman*. Pacman es el gestor de paquetes utilizado por Arch Linux, dicho gestor está escrito en C y busca ser minimalista y rápido. 2026. URL: <https://wiki.archlinux.org/title/Pacman> (visitado 24-02-2026).
- [6] Wikipedia. *Computer_terminal*. 2026. URL: https://en.wikipedia.org/wiki/Computer_terminal#Emulation (visitado 25-02-2026).
- [7] Arch Linux Wiki. *Install_essential_packages*. 2026. URL: https://wiki.archlinux.org/title/Installation_guide#Install_essential_packages (visitado 25-02-2026).
- [8] Arch Linux Wiki. *Polkit*. 2026. URL: <https://wiki.archlinux.org/title/Polkit> (visitado 28-02-2026).
- [9] Arch Linux Wiki. *Reflector*. 2026. URL: <https://wiki.archlinux.org/title/Reflector> (visitado 28-02-2026).
- [10] Arch Linux Wiki. *iwd*. 2026. URL: <https://wiki.archlinux.org/title/Iwd> (visitado 07-03-2026).
- [11] Wikipedia. *X Window System*. 2026. URL: https://en.wikipedia.org/wiki/X_Window_System (visitado 15-03-2026).
- [12] Wikipedia. *Wayland (protocol)*. 2026. URL: [https://en.wikipedia.org/wiki/Wayland_\(protocol\)](https://en.wikipedia.org/wiki/Wayland_(protocol)) (visitado 15-03-2026).
- [13] i3. *i3wm configuration docs*. 2026. URL: <https://i3wm.org/docs/userguide.html> (visitado 14-04-2026).
- [14] Wikipedia. *Compositing_manager*. Definición general de lo que es un compositor. 2026. URL: https://en.wikipedia.org/wiki/Compositing_manager (visitado 14-04-2026).
- [15] Arch Linux Wiki. *picom*. 2026. URL: <https://wiki.archlinux.org/title/Picom> (visitado 14-04-2026).
- [16] Yuxuan Shui. *picom configuration*. 2026. URL: <https://picom.app/> (visitado 14-04-2026).
- [17] Arch Linux Wiki. *rofi*. 2026. URL: <https://wiki.archlinux.org/title/Rofi> (visitado 14-04-2026).
- [18] davatorium. *rofi docs*. 2026. URL: <https://davatorium.github.io/rofi/> (visitado 14-04-2026).