
Crea una distro GNU/Linux

Índex

Objectius	1
Requisits previs	1
1. Instal·la les eines necessàries	2
2. Crea l'estructura del projecte	2
3. Configura l'arquitectura i el tipus d'imatge	3
4. Tria els paquets del sistema	4
5. Posa-li nom a la distro	6
5.1. El fitxer clau: /etc/os-release	6
5.2. El missatge de benvinguda al terminal (/etc/issue i /etc/motd)	6
5.3. El nom al menú d'arrencada (GRUB/isolinux)	6
5.4. El nom de l'ordinador per defecte (hostname)	7
6. Personalitza la distro	7
6.1. El logo al panell "Quant a"	7
6.2. La pantalla del menú d'arrencada (isolinux/syslinux)	9
6.3. El logo a l'arrencada (Plymouth --- pantalla de càrrega)	11
6.4. Configura el tema Yaru (GTK, icones, cursor i so)	11
6.5. Edita les aplicacions al Dash to Dock	12
6.6. Mostra el logo a fastfetch (opcional)	13
7. Afegeix fitxers propis al sistema final	14
7.1. Afegeix un tapís d'escriptori	14
7.2. Elimina la visita guiada	15
7.3. Activa els botons maximitza i minimitza de les finestres	15
7.4. Instal·la i activa les extensions Dash to Dock i Ubuntu AppIndicators	15
8. Automatitza tasques amb hooks	16
9. Configura el català com a únic idioma del sistema	17
9.1. Defineix el locale per defecte	17
9.2. Afegeix els paquets de traducció al català	18
9.3. Configura el teclat en català	18
9.4. Fes que l'idioma es mantingui a cada arrencada (live-config)	19
9.5. Configura el fus horari	20
9.6. Purga la resta de locales (opcional)	20
10. Afegeix un instal·lador amb Calamares (opcional)	22
10.1. Afegeix els paquets necessaris	22
10.2. Personalitza la configuració de Calamares	23
10.3. Configura el fus horari per defecte de l'instal·lador	27
10.4. Personalitza la drecera d'instal·lació nativa	27
10.5. Elimina la icona d'instal·lació un cop instal·lat el sistema	28
10.6. Elimina l'usuari de la sessió live i assigna l'avatar de l'usuari instal·lat	30
10.7. Personalitza la fortalesa de la contrasenya de l'usuari	31

11. Construeix l'ISO	32
12. Prova l'ISO	34
13. Neteja i reconstrueix	43
Resum del flux de treball	44
Referències	45
Índex de figures	46

Una distribució Linux (“distro”) és, en essència, un kernel Linux acompanyat d'un conjunt de programari, eines de gestió de paquets i configuracions que la converteixen en un sistema operatiu complet i utilitzable. Crear-ne una des de zero (compilant cada component) és possible amb projectes com *Linux From Scratch*, però per a un ús pràctic i educatiu resulta molt més eficient partir d'una base existent i personalitzar-la.

En aquest document s'explica com construir una **ISO personalitzada** basada en Debian Sid (la branca inestable/rolling) utilitzant `live-build`, l'eina oficial del projecte Debian Live per generar imatges *en viu* i instal·lables.

NOTA

Una “distro derivada” (com Ubuntu ho és de Debian, o Kali ho és d'Ubuntu/Debian) es construeix normalment amb aquesta mateixa filosofia: agafar una base sòlida i afegir-hi personalització, paquets i configuració pròpia.

Objectius

- Entendre l'estructura d'un projecte `live-build`
- Configurar l'arquitectura, els dipòsits i el tipus d'imatge
- Personalitzar la llista de paquets preinstal·lats
- Afegir fitxers propis (wallpapers, configuracions) al sistema final
- Automatitzar tasques amb *hooks*
- Generar i provar l'ISO resultant

Requisits previs

- Una màquina (física o virtual) amb **Debian** instal·lat
- Accés a internet (el procés descarrega paquets del dipòsit oficial)
- Com a mínim 20 GB d'espai lliure en disc i 2 GB de RAM

Si vols personalitzar la distro necessitaràs alguns fitxers. Pots aportar els teus o bé descarregar-los. En qualsevol cas, els has de tenir al teu home:

Fitxer	Enllaç de descàrrega
avatar.png	https://proferamon.com/tic/imatges/distro/avatar.png
benvinguda.png	https://proferamon.com/tic/imatges/distro/benvinguda.png
diapositiva1.png	https://proferamon.com/tic/imatges/distro/diapositiva1.png
diapositiva2.png	https://proferamon.com/tic/imatges/distro/diapositiva2.png
distrothos-ascii.ans	https://proferamon.com/tic/imatges/distro/distrothos-ascii.ans
distrothos-logo.png	https://proferamon.com/tic/imatges/distro/distrothos-logo.png
distrothos-logo.svg	https://proferamon.com/tic/imatges/distro/distrothos-logo.svg
distrothos-logo-quadrat.svg	https://proferamon.com/tic/imatges/distro/distrothos-logo-quadrat.svg
wallpaper.png	https://proferamon.com/tic/imatges/distro/wallpaper.png

L'estructura de fitxers dins del teu home hauria de ser:

```
/home/<usuari>/
├── avatar.png
├── benvinguda.png
├── diapositiva1.png
├── diapositiva2.png
├── distrothos-ascii.ans
├── distrothos-logo.png
├── distrothos-logo.svg
├── distrothos-logo-quadrat.svg
├── wallpaper.png
└── la-meva-distro/      <- Carpeta que crearàs a la secció 2
```

1. Instal·la les eines necessàries

Actualitza la llista de paquets:

```
sudo apt update
```

Instal·la les eines:

```
sudo apt install live-build live-config live-boot debootstrap
```

Paquet	Funció
live-build	Conjunt de scripts (lb) per construir la imatge
live-config	Configuració que s'aplica en arrencar el sistema live
live-boot	Scripts d'arrencada per a sistemes live (initramfs)
debootstrap	Crea el sistema base de Debian dins d'un chroot

2. Crea l'estructura del projecte

Crea la carpeta de treball:

```
mkdir ~/la-meva-distro
```

Mou-te a la carpeta creada:

```
cd ~/la-meva-distro
```

3. Configura l'arquitectura i el tipus d'imatge

```
lb config \
--distribution sid \
--distribution-chroot sid \
--distribution-binary sid \
--architectures amd64 \
--binary-images iso-hybrid \
--archive-areas "main contrib non-free non-free-firmware" \
--firmware-chroot false \
--bootappend-live "boot=live components locales=ca_ES.UTF-8
↳ keyboard-layouts=es keyboard-variants=cat
↳ timezone=Europe/Madrid" \
--iso-application "DistroThos" \
--iso-publisher "Ramon López" \
--iso-volume "DistroThos"
```

Explicació dels paràmetres:

- `--distribution sid` --- versió de Debian que es farà servir com a base
- `--architectures amd64` --- arquitectura de processador de destí
- `--binary-images iso-hybrid` --- genera una ISO híbrida, vàlida tant per gravar en DVD com per copiar a un USB booteable amb dd
- `--archive-areas` --- habilita dipòsits addicionals com non-free (necessari, per exemple, per a alguns drivers de xarxa o gràfics)
- `--firmware-chroot false` --- desactiva la inclusió automàtica de tot el firmware disponible a non-free-firmware; en triaràs explícitament només el que necessitis
- `--bootappend-live` --- defineix els paràmetres d'arrencada del kernel
- `--iso-application` --- nom que apareix com a metadada de l'ISO (per exemple, en detectar-la alguns gravadors o eines de creació d'USB)
- `--iso-publisher` --- informació de l'editor
- `--iso-volume` --- etiqueta del volum (limitada a certs caràcters, sense accents ni espais llargs recomanables)

Aquesta ordre genera la carpeta `config/`, que conté tota la configuració modular del projecte. És, en certa manera, el "codi font" de la nostra distro.

NOTA

En lloc de `sid` que és la branca *inestable/rolling*, pots triar `forky` (testing) o `trixie` (stable).

4. Tria els paquets del sistema

Els paquets es defineixen en fitxers de text dins de `config/package-lists/`. És recomanable separar-los per categories:

`config/package-lists/escriptori.list.chroot`

```
mkdir -p config/package-lists
nano config/package-lists/escriptori.list.chroot
```

```
task-gnome-desktop
```

NOTA

`task-gnome-desktop` instal·la l'entorn GNOME complet (amb totes les aplicacions habituals de GNOME Core i GNOME Extra). Si vols un GNOME més lleuger, pots substituir-ho per `gnome-core`, que només inclou els components essencials.

Alternatives:

Metapaquet	Escriptori
<code>task-gnome-desktop</code>	GNOME
<code>gnome</code>	GNOME Core + Extra (jocs, Cheese, Rhythmbox, Totem, utilitats addicionals...)
<code>gnome-core</code>	El nucli essencial de GNOME. Només els paquets imprescindibles
<code>task-kde-desktop</code>	KDE Plasma
<code>task-xfce-desktop</code>	Xfce
<code>task-mate-desktop</code>	MATE
<code>task-cinnamon-desktop</code>	Cinnamon
<code>task-lxde-desktop</code>	LXDE
<code>task-lxqt-desktop</code>	LXQt
<code>task-phosh-desktop</code>	Phosh (entorn mòbil, basat en GNOME Shell)
<code>task-lomiri-desktop</code>	Lomiri (antic Unity8, d'Ubuntu Touch)
<code>task-lomiri-tablet</code>	Variant de Lomiri per a tauletes

`config/package-lists/aplicacions.list.chroot`

```
nano config/package-lists/aplicacions.list.chroot
```

```
bash-completion
build-essential
bzip2
evince
fastfetch
firefox-esr
firefox-esr-l10n-ca
foliate
geany
gedit
gimp
gnome-tweaks
```

```
gparted
htop
ksnip
libreoffice
libreoffice-l10n-ca
linux-headers-amd64
ptyxis
synaptic
tar
virtualbox
virtualbox-dkms
```

NOTA

Aquesta llista conté els paquets que vulguis que estiguin instal·lats per defecte. Pots afegir-ne molts més.

config/package-lists/xarxa.list.chroot

```
nano config/package-lists/xarxa.list.chroot
```

```
network-manager
openssh-client
```

config/package-lists/firmware.list.chroot

```
nano config/package-lists/firmware.list.chroot
```

```
firmware-linux-nonfree
firmware-realtek
firmware-atheros
firmware-iwlwifi
firmware-amd-graphics
firmware-intel-graphics
firmware-intel-sound
firmware-intel-misc
```

ACTIVITAT

Decideix quin serà el propòsit de la teva distro (ciberseguretat, ofimàtica, servidor mínim, docència...) i crea les teves llistes de paquets justificant cada tria. Pots consultar els paquets disponibles amb `apt search <paraula clau>` des d'una màquina Debian.

5. Posa-li nom a la distro

5.1. El fitxer clau: /etc/os-release

config/includes.chroot/etc/os-release

```
mkdir -p config/includes.chroot/etc
nano config/includes.chroot/etc/os-release
```

```
PRETTY_NAME="DistroThos 1.0"
NAME="DistroThos"
VERSION_ID="1.0"
VERSION="1.0 (Sid)"
VERSION_CODENAME=sid
ID=distrothos
ID_LIKE=debian
HOME_URL="https://www.iesthosisicodina.cat/"
SUPPORT_URL="https://www.iesthosisicodina.cat/suport"
BUG_REPORT_URL="https://www.iesthosisicodina.cat/"
```

5.2. El missatge de benvinguda al terminal (/etc/issue i /etc/motd)

config/includes.chroot/etc/issue

```
nano config/includes.chroot/etc/issue
```

```
DistroThos 1.0 \n \l
```

config/includes.chroot/etc/motd

```
nano config/includes.chroot/etc/motd
```

```
Benvingut a DistroThos --- distribució educativa per al CFGS ASIX
```

5.3. El nom al menú d'arrencada (GRUB/isolinux)

Aquí es fa servir el paràmetre `--iso-application` i similars de `lb config`. Vegeu la secció 3

- `--iso-application` --- nom que apareix com a metadada de l'ISO (per exemple, en detectar-la alguns gravadors o eines de creació d'USB)
- `--iso-publisher` --- informació de l'editor
- `--iso-volume` --- etiqueta del volum (limitada a certs caràcters, sense accents ni espais llargs recomanables)

5.4. El nom de l'ordinador per defecte (hostname)

config/includes.chroot/etc/hostname

```
nano config/includes.chroot/etc/hostname
```

```
distrothos
```

6. Personalitza la distro

6.1. El logo al panell “Quant a”

Amb el paquet imagemagick genera les mides necessàries, i sobreescriu els fitxers del paquet.

```
sudo apt install imagemagick
```

Genera les variants PNG del teu logo, a les mides que desktop-base espera:

```
for mida in 64 128 256; do
    interior=$(( mida * 80 / 100 ))
    convert -background none ~/distrothos-logo.svg \
        -resize ${interior}x${interior} \
        -gravity center \
        -extent ${mida}x${mida} \
        logo-$mida.png
done
```

Sobreescriu els fitxers reals reutilitzant els fitxers generats:

```
mkdir -p
↳ config/includes.chroot/usr/share/icons/desktop-base/64x64/emblems
mkdir -p config/includes.chroot/usr/share/icons/desktop-base/128x128
↳ /emblems
mkdir -p config/includes.chroot/usr/share/icons/desktop-base/256x256
↳ /emblems
mkdir -p config/includes.chroot/usr/share/icons/desktop-base/scalabl
↳ e/emblems
cp logo-64.png config/includes.chroot/usr/share/icons/desktop-base/6
↳ 4x64/emblems/emblem-debian.png
cp logo-128.png config/includes.chroot/usr/share/icons/desktop-base/
↳ 128x128/emblems/emblem-debian.png
cp logo-256.png config/includes.chroot/usr/share/icons/desktop-base/
↳ 256x256/emblems/emblem-debian.png
cp ~/distrothos-logo-quadrat.svg config/includes.chroot/usr/share/ic
↳ ons/desktop-base/scalable/emblems/emblem-debian.svg
```

Sobreescriu també els logos de GDM i la pantalla de bloqueig

Reutilitza els PNG generats. Aquests fitxers viuen en una carpeta diferent del paquet desktop-base (desktop-base/debian-logos/, no icons/desktop-base/), i alimenten un component diferent del sistema: la pantalla de login/bloqueig de GDM, no el panell “Quant a” que ja hem cobert amb el bloc anterior. Sense aquest segon bloc, el logo apareixeria correctament a “Quant a” però la pantalla de GDM continuaria mostrant el remolí clàssic de Debian:

```
mkdir -p config/includes.chroot/usr/share/desktop-base/debian-logos
cp ~/avatar.png config/includes.chroot/usr/share/desktop-base/debian-
↳ logos/avatar.png
cp logo-64.png config/includes.chroot/usr/share/desktop-base/debian-
↳ logos/logo-64.png
cp logo-128.png config/includes.chroot/usr/share/desktop-base/debian-
↳ logos/logo-128.png
cp logo-256.png config/includes.chroot/usr/share/desktop-base/debian-
↳ logos/logo-256.png
cp logo-128.png config/includes.chroot/usr/share/desktop-base/debian-
↳ logos/logo-text-128.png
cp logo-256.png config/includes.chroot/usr/share/desktop-base/debian-
↳ logos/logo-text-256.png
cp logo-64.png config/includes.chroot/usr/share/desktop-base/debian-
↳ logos/logo-text-64.png
cp logo-128.png config/includes.chroot/usr/share/desktop-base/debian-
↳ logos/logo-text-version-128.png
cp logo-256.png config/includes.chroot/usr/share/desktop-base/debian-
↳ logos/logo-text-version-256.png
cp logo-64.png config/includes.chroot/usr/share/desktop-base/debian-
↳ logos/logo-text-version-64.png
```

6.2. La pantalla del menú d'arrencada (isolinux/syslinux)

La pantalla que veuries en arrencar l'ISO (amb el logo groc, el "Boot menu" i el text tècnic sobre la construcció) pertany al bootloader **isolinux/syslinux**, i **live-build** permet substituir-la completament amb un tema propi.

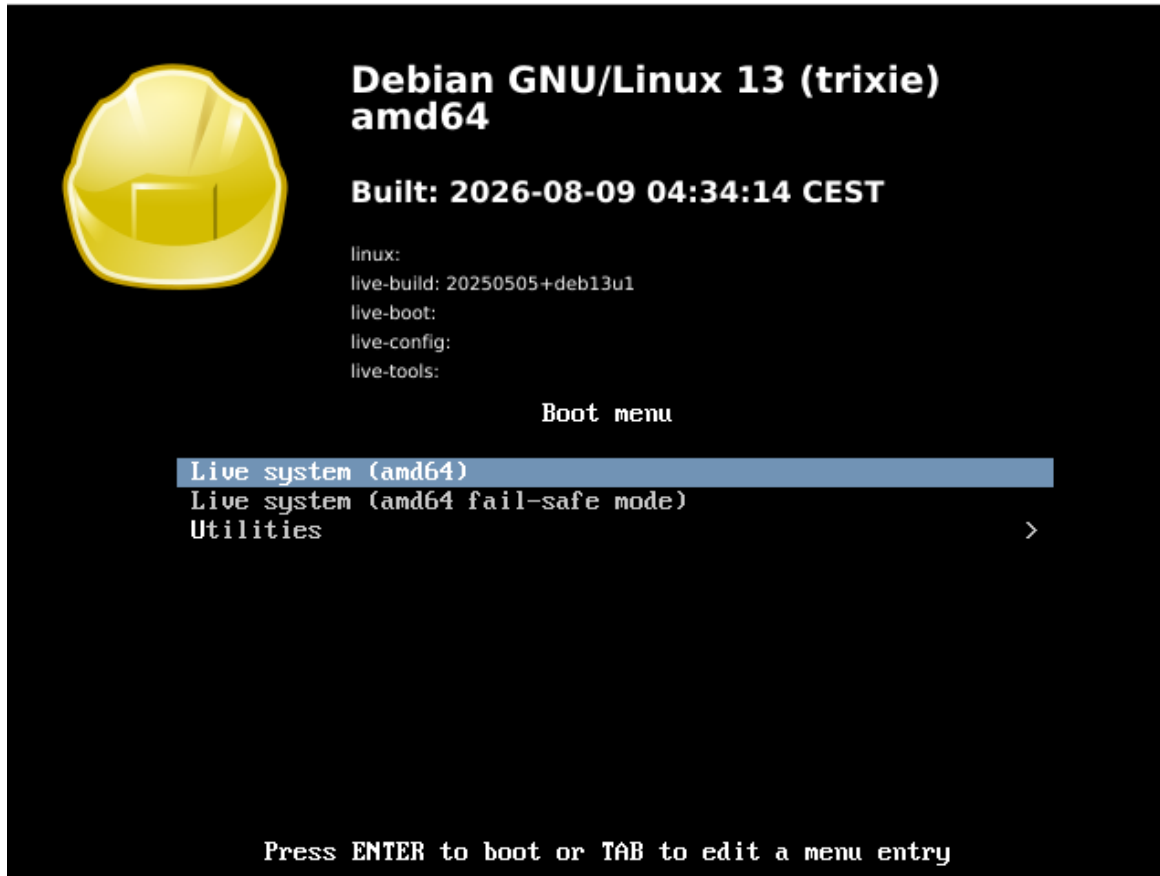


Figura 1: isolinux live

Crea la carpeta boot loaders:

```
mkdir -p config/bootloaders
```

Assegura't que tens els paquets de syslinux instal·lats a l'ordinador (calen per disposar dels mòduls `.c32` que farem servir més endavant):

```
sudo apt install syslinux-common syslinux-utils
```

Copia la plantilla per defecte com a punt de partida:

```
cp -r /usr/share/live/build/bootloaders/isolinux  
↪ config/bootloaders/isolinux
```

Aquesta plantilla **no inclou tots els mòduls `.c32` necessaris** perquè el menú gràfic (`vesamenu.c32`) funcioni --- només porta `isolinux.bin` i `ldlinux.c32`. Sense la

resta de mòduls (vesamenu.c32, libutil.c32, libcom32.c32, libmenu.c32...), l'arrencada cau a un prompt de text pla (boot:) en lloc de mostrar el menú gràfic amb la imatge de fons. Copia'ls des de la carpeta oficial que live-build ja reuneix internament:

```
cp /usr/share/live/build/bootloaders/syslinux_common/*.c32
↪ config/bootloaders/isolinux/
```

Substitueix el fons per una imatge pròpia.

El splash.svg original és qui genera dinàmicament tot el text tècnic ("Built:", "linux:", "live-build:...") --- en substituir-lo per un PNG estàtic, aquest text desapareix i et quedes només amb la imatge que tu vulguis:

Instal·la el paquet libsvg2-bin:

```
sudo apt install libsvg2-bin
```

Converteix el teu logo (distrothos-logo.svg) a un fitxer temporal:

```
rsvg-convert -w 400 -b none ~/distrothos-logo.svg -o /tmp/logo-400.png
```

Converteix el fitxer temporal en el fitxer splash.png que es veurà al fons del menú:

```
convert -size 640x480 xc:"#2c3e50" /tmp/logo-400.png -gravity center
↪ -composite -alpha off -depth 8
↪ config/bootloaders/isolinux/splash.png
```

Personalitza el text del menú i els colors:

Copia els fitxers originals:

```
cp /usr/share/live/build/bootloaders/syslinux_common/menu.cfg
↪ config/bootloaders/
cp /usr/share/live/build/bootloaders/syslinux_common/stdmenu.cfg
↪ config/bootloaders/isolinux/
```

I edita només la línia del títol, sense sobreescriure la resta del fitxer:

```
sed -i 's/^menu title.*/menu title DistroThos - Sistema Live/'
↪ config/bootloaders/isolinux/menu.cfg
```

NOTA

Aquest tema només afecta les arrencades en mode BIOS (isolinux/syslinux). Si vols el mateix aspecte també per a arrencades UEFI (que fan servir GRUB), cal repetir el mateix procés amb `cp -r /usr/share/live/build/bootloaders/grub-pc config/bootloaders/grub-pc` i `cp -r /usr/share/live/build/bootloaders/grub-efi config/bootloaders/grub-efi` --- la lògica de splash.png/splash.svg i els fitxers de configuració és equivalent, amb sintaxi pròpia de GRUB.

ACTIVITAT

Personalitza el fons i el títol del menú d'arrencada, reconstrueix l'ISO i comprova el resultat arrencant-la en una màquina virtual.

6.3. El logo a l'arrencada (Plymouth --- pantalla de càrrega)

Un tema de Plymouth complet és força feina (requereix un script `.plymouth`, imatges d'animació, etc.), però la versió mínima amb un logo estàtic centrat és senzilla amb el tema text o reutilitzant-ne un d'existent.

Afegeix plymouth a la teva llista de paquets:

```
nano config/package-lists/branding.list.chroot
```

```
plymouth
plymouth-themes
librsvg2-common
```

I després sobreescriu el logo:

```
mkdir -p config/includes.chroot/usr/share/plymouth/themes/spinner
cp ~/distrothos-logo.png config/includes.chroot/usr/share/plymouth/themes/spinner/watermark.png
```

NOTA

Fer un tema de Plymouth totalment personalitzat des de zero és bastant més laboriós (calen diversos fitxers `.script`, `.plymouth` i `sprites`). Reaprofitar un tema existent i substituir-ne només la imatge del logo (`watermark.png` o similar, segons el tema) és molt més ràpid per a un projecte docent.

6.4. Configura el tema Yaru (GTK, icones, cursor i so)

Per aplicar el tema visual **Yaru** (el d'Ubuntu) a l'escriptori, cal instal·lar els paquets corresponents i indicar-li a GNOME, via `dconf`, que els faci servir per defecte. El tema de GNOME Shell (la barra superior) necessita un pas addicional, perquè GNOME no l'aplica sense una extensió activada.

Amplia `config/package-lists/branding.list.chroot`:

```
nano config/package-lists/branding.list.chroot
```

Afegeix al contingut existent:

```
yaru-theme-gtk
yaru-theme-icon
```

```
varu-theme-sound
varu-theme-unity
```

NOTA

varu-theme-unity no és necessari per a GNOME (és per a l'escriptori Unity), però no fa mal deixar-lo si vols mantenir el conjunt complet de paquets Yaru sencer. Si vols estalviar espai, el pots ometre.

Aplica el tema GTK, les icones, el cursor, el so i el tema de GNOME Shell via dconf:
config/includes.chroot/etc/dconf/db/local.d/00-tema-varu

```
mkdir -p config/includes.chroot/etc/dconf/db/local.d
nano config/includes.chroot/etc/dconf/db/local.d/00-tema-varu
```

```
[org/gnome/desktop/interface]
gtk-theme='Yaru-sage'
icon-theme='Yaru-sage'
cursor-theme='Yaru'
color-scheme='default'
accent-color='orange'
```

NOTA

Aquest fitxer 00-tema-varu s'ampliarà més endavant.

6.5. Edita les aplicacions al Dash to Dock

Les icones fixades al Dash to Dock es controlen amb la clau favorite-apps, dins del mateix grup [org/gnome/shell] on ja tens enabled-extensions. Com que aquest valor viu a /etc/dconf/db/local.d/, es compila amb el mateix dconf update del hook i es copia sencer al sistema instal·lat, **no cal fer res diferent per a la live i per a l'instal·lat**: n'hi ha prou amb aquest únic canvi.

Amplia config/includes.chroot/etc/dconf/db/local.d/00-tema-varu:

```
nano config/includes.chroot/etc/dconf/db/local.d/00-tema-varu
```

Afegeix favorite-apps dins del bloc [org/gnome/shell] ja existent:

```
[org/gnome/shell]
enabled-extensions=['dash-to-dock@micxgx.gmail.com',
↵ 'ubuntu-appindicators@ubuntu.com']
```

```
favorite-apps=['calamares-install-debian.desktop',
↳ 'firefox-esr.desktop', 'org.gnome.Evolution.desktop',
↳ 'org.gnome.Nautilus.desktop', 'org.gnome.gedit.desktop',
↳ 'org.gnome.Ptyxis.desktop', 'libreoffice-startcenter.desktop',
↳ 'org.gnome.Evince.desktop', 'gimp.desktop', 'virtualbox.desktop',
↳ 'synaptic.desktop']
```

6.6. Mostra el logo a fastfetch (opcional)

fastfetch (ja inclòs a `config/package-lists/aplicacions.list.chroot`, secció 4) mostra per defecte el logo de la distro detectat a partir de l'ID d'/etc/os-release. Com que distrothos no existeix a la seva base de dades interna de logos, cau a un logo genèric. Per mostrar-hi el teu, cal un fitxer de configuració `config.jsonc` que apunti a un logo propi.

Crea la configuració, perquè s'apliqui a qualsevol usuari nou (l'alumne de la sessió live, i el que crea Calamares en instal·lar):

config/includes.chroot/etc/skel/.config/fastfetch/config.jsonc

```
mkdir -p config/includes.chroot/etc/skel/.config/fastfetch
cp ~/distrothos-ascii.ans
↳ config/includes.chroot/etc/skel/.config/fastfetch/logo.ans
nano config/includes.chroot/etc/skel/.config/fastfetch/config.jsonc
```

```
{
  "$schema": "https://github.com/fastfetch-cli/fastfetch/raw/dev/doc_
↳ /json_schema.json",
  "logo": {
    "type": "file",
    "source": "~/config/fastfetch/logo.ans",
    "height": 15
  },
  "modules": [
    "title",
    "separator",
    "os",
    "host",
    "kernel",
    "uptime",
    "packages",
    "shell",
    "display",
    "de",
    "wm",
    "wmtheme",
    "theme",
    "icons",
    "font",
    "cursor",
```



```
"terminal",
"terminalfont",
"cpu",
"gpu",
"memory",
"swap",
"disk",
"localip",
"locale",
"colors"
]
}
```

7. Afegeix fitxers propis al sistema final

7.1. Afegeix un tapís d'escriptori

Tot el que es col·loqui dins de `config/includes.chroot/usr/share/backgrounds/` s'incorpora directament al sistema de fitxers de l'ISO final, respectant la ruta real:

Crea la carpeta `backgrounds` i copia el teu fons d'escriptori a la carpeta creada:

```
mkdir -p config/includes.chroot/usr/share/backgrounds
cp ~/wallpaper.png config/includes.chroot/usr/share/backgrounds/
```

Per exemple, per definir un wallpaper per defecte a GNOME, cal fer servir `dconf` per establir valors per defecte pel que fa al sistema, ja que GNOME emmagatzema la seva configuració en aquesta base de dades en lloc de fitxers `.conf` tradicionals:

`config/includes.chroot/etc/dconf/db/local.d/00-fons-pantalla`

```
mkdir -p config/includes.chroot/etc/dconf/db/local.d
nano config/includes.chroot/etc/dconf/db/local.d/00-fons-pantalla
```

```
[org/gnome/desktop/background]
picture-uri='file:///usr/share/backgrounds/wallpaper.png'
picture-uri-dark='file:///usr/share/backgrounds/wallpaper.png'
```

```
mkdir -p config/includes.chroot/etc/dconf/profile
```

`config/includes.chroot/etc/dconf/profile/user`

```
nano config/includes.chroot/etc/dconf/profile/user
```

```
user-db:user
system-db:local
```

7.2. Elimina la visita guiada

config/includes.chroot/etc/dconf/db/local.d/00-benvinguda

```
nano config/includes.chroot/etc/dconf/db/local.d/00-benvinguda
```

```
[org/gnome/shell]
welcome-dialog-last-shown-version='48.4'
```

7.3. Activa els botons maximitza i minimitza de les finestres

config/includes.chroot/etc/dconf/db/local.d/00-finestres

```
nano config/includes.chroot/etc/dconf/db/local.d/00-finestres
```

```
[org/gnome/desktop/wm/preferences]
button-layout=':minimize,maximize,close'
```

7.4. Instal·la i activa les extensions Dash to Dock i Ubuntu AppIndicators

config/package-lists/branding.list.chroot

```
nano config/package-lists/branding.list.chroot
```

Afegeix aquestes línies:

```
gnome-shell-extension-dashtodock
gnome-shell-extension-appindicator
```

config/includes.chroot/etc/dconf/db/local.d/00-tema-yaru

```
nano config/includes.chroot/etc/dconf/db/local.d/00-tema-yaru
```

Afegeix aquestes línies al contingut del fitxer:

```
[org/gnome/shell/extensions/dash-to-dock]
apply-custom-theme=false
apply-glossy-effect=false
background-opacity=0.8
custom-background-color=false
custom-theme-running-dots-border-color='rgb(255,255,255) '
custom-theme-running-dots-color='rgb(255,255,255) '
custom-theme-shrink=true
dash-max-icon-size=40
dock-fixed=true
```

```
dock-position='LEFT'
extend-height=true
height-fraction=1.0
hide-tooltip=false
icon-size-fixed=true
running-indicator-style='DOTS'
scroll-action='switch-workspace'
unity-backlit-items=false
show-trash=true
```

Aquests valors s'han de “compilar” a la base de dades dconf mitjançant un hook (vegeu la secció següent), ja que dconf update s'ha d'executar dins del chroot perquè el canvi tingui efecte.

8. Automatitza tasques amb *hooks*

Els *hooks* són scripts que s'executen dins del chroot durant la construcció, útils per a tasques que no es poden resoldre només copiant fitxers (activar serveis, crear usuaris, etc.).

config/hooks/live/0100-configuracio.hook.chroot

```
mkdir -p config/hooks/live
nano config/hooks/live/0100-configuracio.hook.chroot
```

```
#!/bin/sh
set -e

# Exemple: habilitar el servei NetworkManager per defecte
systemctl enable NetworkManager

# Exemple: crear un usuari per defecte
useradd -m -s /bin/bash -G sudo alumne
echo "alumne:alumne" | chpasswd

# --- Elimina els logos originals de Debian que farien ombra als teus
↳ (secció 6.1) ---
rm -f /usr/share/desktop-base/debian-logos/logo.svg
rm -f /usr/share/desktop-base/debian-logos/logo-text.svg
rm -f /usr/share/desktop-base/debian-logos/logo-text-version.svg
rm -f /usr/share/icons/desktop-base/icon-theme.cache

# --- Avatar de l'usuari alumne a GDM (AccountsService) ---
mkdir -p /var/lib/AccountsService/icons
↳ /var/lib/AccountsService/users
cp /usr/share/desktop-base/debian-logos/avatar.png
↳ /var/lib/AccountsService/icons/alumne
chmod 644 /var/lib/AccountsService/icons/alumne
printf '[User]\nIcon=/var/lib/AccountsService/icons/alumne\n' >
↳ /var/lib/AccountsService/users/alumne
```

```
# Selecciona el tema "spinner" de Plymouth i regenera l'initramfs
plymouth-set-default-theme spinner
update-initramfs -u

# Compilar la configuració de dconf (necessari perquè el wallpaper
# i altres valors per defecte de GNOME tinguin efecte)
dconf update
```

```
chmod +x config/hooks/live/0100-configuracio.hook.chroot
```

AVÍS

Definir contrasenyes en text pla dins d'un hook és útil per a entorns docents o de proves, però **mai** s'hauria de fer en una distro destinada a producció o distribució pública.

9. Configura el català com a únic idioma del sistema

Per defecte, live-build genera un sistema amb els locals habituals de Debian (anglès inclòs). Si vols que la distro live arrenqui íntegrament en català, cal actuar en diversos punts: el *locale* per defecte, els paquets de traducció, el teclat i, opcionalment, la neteja d'altres idiomes.

9.1. Defineix el locale per defecte

Amplia el hook `config/hooks/live/0100-configuracio.hook.chroot` (creat a la secció 8) afegint-hi el següent, abans de la línia `dconf update`:

```
nano config/hooks/live/0100-configuracio.hook.chroot
```

```
# Preseed del paquet locales: generar només ca_ES.UTF-8
echo "locales locales/locales_to_be_generated multiselect ca_ES.UTF-8
↳ UTF-8" | debconf-set-selections
echo "locales locales/default_environment_locale select ca_ES.UTF-8"
↳ | debconf-set-selections

# Reconfigurar el paquet locales perquè apliqui aquests valors
dpkg-reconfigure -f noninteractive locales

# Fixar-ho explícitament per si de cas
update-locale LANG=ca_ES.UTF-8 LANGUAGE=ca_ES:ca
```

Això fa que **només** es generi el locale català (no els habituals en_US.UTF-8, C.UTF-8, etc.) i que sigui el que s'utilitza per defecte a tot el sistema, GNOME inclòs --- la sessió hereta automàticament el LANG del sistema.

9.2. Afegeix els paquets de traducció al català

`config/package-lists/idioma.list.chroot`

```
nano config/package-lists/idioma.list.chroot
```

```
task-catalan
task-catalan-desktop
locales
keyboard-configuration
```

`task-catalan` és el *task* de Debian que instal·la automàticament els paquets de suport lingüístic disponibles: diccionaris, correctors ortogràfics i els paquets `-l10n-ca` de moltes aplicacions (LibreOffice, etc.) quan n'hi ha.

NOTA

Alguns programes (com Firefox ESR) necessiten el paquet específic de la interfície en català a banda, per exemple `firefox-esr-l10n-ca`. Val la pena revisar quins paquets concrets de la teva llista d'aplicacions tenen versió en català disponible amb `apt search l10n-ca`.

9.3. Configura el teclat en català

Amplia el hook `config/hooks/live/0100-configuracio.hook.chroot` (creat a la secció 8) afegint-hi el següent, abans de la línia `dconf update`:

```
nano config/hooks/live/0100-configuracio.hook.chroot
```

```
echo "keyboard-configuration keyboard-configuration/layoutcode string
↵ es" | debconf-set-selections
echo "keyboard-configuration keyboard-configuration/variantcode
↵ string cat" | debconf-set-selections
dpkg-reconfigure -f noninteractive keyboard-configuration
```

Això configura la distribució espanyola amb variant catalana (`es(cat)`), l'habitual per a teclats amb aquesta disposició.

9.4. Fes que l'idioma es mantingui a cada arrencada (live-config)

Amb els passos anteriors, el sistema es **construeix** en català, però hi ha un parany habitual: els sistemes live porten el paquet `live-config`, que s'executa **a cada arrencada** (no durant `lb build`) i reconfigura l'idioma, el teclat i altres paràmetres a partir dels valors d'arrencada del GRUB/isolinux. Si no li dius explícitament que faci servir català, aplicarà el seu valor per defecte (normalment anglès), **sobreescrivint** la configuració que vas fixar al hook. És per això que l'ISO es pot construir “en català” però arrencar en anglès.

Hi ha dues maneres de solucionar-ho, complementàries:

A. Paràmetre d'arrencada, amb `--bootappend-live` a `lb config`

Vegeu la secció 3

Això afegeix els paràmetres a la línia d'arrencada per defecte, de manera que `live-config` els aplica automàticament sense que l'usuari hagi de teclejar res al menú de GRUB.

B. Configuració persistent de `live-config` (recomanat, no depèn del menú d'arrencada):

`config/includes.chroot/etc/live/config.conf.d/locale.conf`

```
mkdir -p config/includes.chroot/etc/live/config.conf.d
nano config/includes.chroot/etc/live/config.conf.d/locale.conf
```

```
LIVE_LOCALES="ca_ES.UTF-8"
LIVE_KEYBOARD_LAYOUTS="es"
LIVE_KEYBOARD_VARIANTS="cat"
LIVE_TIMEZONE="Europe/Madrid"
```

Aquest fitxer es llegeix a cada arrencada i estableix aquests valors com a comportament per defecte del sistema live, independentment dels paràmetres passats des del GRUB.

CONSELL

Es poden combinar les dues opcions: la B com a valor per defecte fiable, i la A disponible per si algú vol arrencar temporalment en un altre idioma des del menú d'arrencada.

Resum:

Aspecte	Mecanisme
Locale per defecte (construcció)	<code>debconf-set-selections + dpkg-reconfigure locales</code>
Paquets de traducció	<code>task-catalan a package-lists</code>
Teclat (construcció)	<code>debconf-set-selections per keyboard-configuration</code>
Idioma i teclat a cada arrencada	<code>live-config</code> via <code>--bootappend-live</code> o <code>/etc/live/config.conf.d/locale.conf</code>
Sessió GNOME	Hereta el <code>LANG</code> del sistema automàticament
Neteja d'altres idiomes	<code>localepurge</code> (opcional)

ACTIVITAT

Configura el teu hook i el fitxer `locale.conf` de `live-config` perquè la distro arrenqui únicament en català (locale, teclat i paquets de traducció) i comprova-ho arrencant l'ISO

en una màquina virtual: revisa el menú d'aplicacions, la configuració regional i el teclat, tant a “Quant a” com a “Regió i idioma”.

9.5. Configura el fus horari

Amplia el hook `config/hooks/live/0100-configuracio.hook.chroot` (creat a la secció 8) afegint-hi el següent, abans de la línia `dconf update`:

```
nano config/hooks/live/0100-configuracio.hook.chroot
```

Afegeix aquestes línies abans de `dconf update`:

```
# --- Fus horari (Europe/Madrid, vàlid per a Catalunya) ---
echo "tzdata tzdata/Areas select Europe" | debconf-set-selections
echo "tzdata tzdata/Zones/Europe select Madrid" |
↳ debconf-set-selections
rm -f /etc/localtime /etc/timezone
dpkg-reconfigure -f noninteractive tzdata
```

9.6. Purga la resta de locales (opcional)

Si vols anar més enllà i eliminar fitxers de locale d'altres idiomes que hagin quedat instal·lats per dependències, per estalviar espai:

```
nano config/hooks/live/0100-configuracio.hook.chroot
```

Afegeix al final:

```
apt-get install -y localepurge
echo "localepurge localepurge/nopurge multiselect ca_ES.UTF-8" |
↳ debconf-set-selections
echo "localepurge localepurge/use-dpkg-feature boolean false" |
↳ debconf-set-selections
dpkg-reconfigure -f noninteractive localepurge
localepurge
```

AVÍS

`localepurge` és una eina agressiva que pot trencar alguns paquets si no es configura bé la llista de locales a conservar. Prova-ho en una VM abans de donar-ho per definitiu, i assegura't que `ca_ES.UTF-8` (exactament amb aquest format) hi és inclòs a `nopurge`.

El hook hauria de quedar així:

```
#!/bin/sh
set -e
```

```

# Exemple: habilitar el servei NetworkManager per defecte
systemctl enable NetworkManager

# --- Usuari per defecte (secció 8) ---
useradd -m -s /bin/bash -G sudo alumne
echo "alumne:alumne" | chpasswd

# --- Elimina els logos originals de Debian que farien ombra als teus
↳ (secció 6.1) ---
rm -f /usr/share/desktop-base/debian-logos/logo.svg
rm -f /usr/share/desktop-base/debian-logos/logo-text.svg
rm -f /usr/share/desktop-base/debian-logos/logo-text-version.svg
rm -f /usr/share/icons/desktop-base/icon-theme.cache

# --- Avatar de l'usuari alumne a GDM (AccountsService) ---
mkdir -p /var/lib/AccountsService/icons
↳ /var/lib/AccountsService/users
cp /usr/share/desktop-base/debian-logos/avatar.png
↳ /var/lib/AccountsService/icons/alumne
chmod 644 /var/lib/AccountsService/icons/alumne
printf '[User]\nIcon=/var/lib/AccountsService/icons/alumne\n' >
↳ /var/lib/AccountsService/users/alumne

# Selecciona el tema "spinner" de Plymouth i regenera l'initramfs
↳ (secció 6.3)
plymouth-set-default-theme spinner
update-initramfs -u

# --- Configuració d'idioma (secció 9.1) ---
echo "locales locales/locales_to_be_generated multiselect ca_ES.UTF-8
↳ UTF-8" | debconf-set-selections
echo "locales locales/default_environment_locale select ca_ES.UTF-8"
↳ | debconf-set-selections
dpkg-reconfigure -f noninteractive locales
update-locale LANG=ca_ES.UTF-8 LANGUAGE=ca_ES:ca

# --- Configuració del teclat (secció 9.3) ---
echo "keyboard-configuration keyboard-configuration/layoutcode string
↳ es" | debconf-set-selections
echo "keyboard-configuration keyboard-configuration/variantcode
↳ string cat" | debconf-set-selections
dpkg-reconfigure -f noninteractive keyboard-configuration

# --- Fus horari (Europe/Madrid, vàlid per a Catalunya) ---
echo "tzdata tzdata/Areas select Europe" | debconf-set-selections
echo "tzdata tzdata/Zones/Europe select Madrid" |
↳ debconf-set-selections
rm -f /etc/localtime /etc/timezone
dpkg-reconfigure -f noninteractive tzdata

# --- Compilar dconf (secció 7/8) ---
dconf update

```



```
# --- Neteja de locales (secció 9.6) --- SEMPRE AL FINAL ---
apt-get install -y localepurge
echo "localepurge localepurge/nopurge multiselect ca_ES.UTF-8" |
↪ debconf-set-selections
echo "localepurge localepurge/use-dpkg-feature boolean false" |
↪ debconf-set-selections
dpkg-reconfigure -f noninteractive localepurge
localepurge
```

10. Afegeix un instal·lador amb Calamares (opcional)

Per defecte, l'ISO que es generaria si no s'afegís un instal·lador seria només un sistema **live**: arrencaria i s'executaria sense instal·lar res al disc. Si vols que els usuaris puguin instal·lar la distro a una màquina, cal afegir-hi un instal·lador. Una opció molt habitual en distros derivades de Debian (Kali, MX Linux, i moltes altres) és **Calamares**, un instal·lador gràfic modern i senzill de configurar.

10.1. Afegeix els paquets necessaris

`config/package-lists/instal·lador.list.chroot`

```
nano config/package-lists/instal·lador.list.chroot
```

```
calamares
calamares-settings-debian
grub-pc
grub-efi-amd64-bin
shim-signed
```

- `grub-pc` --- necessari per a `grub-install --target=i386-pc`
- `grub-efi-amd64-bin` + `shim-signed` --- equivalent per a màquines UEFI (per si algú instal·la la distro en un ordinador més modern)
- `calamares-settings-debian` proporciona una configuració base ja adaptada a sistemes Debian (particions, xarxa, usuaris, etc.), que després es pot personalitzar.

10.2. Personalitza la configuració de Calamares

La configuració de Calamares es troba a `/etc/calamares/` un cop instal·lat el paquet. Si vols modificar-la (per exemple, canviar el nom o el logotip de la distro que apareix durant la instal·lació), pots sobre escriure aquests fitxers copiant-los a `config/includes.chroot/etc/calamares/`:

```
mkdir -p config/includes.chroot/etc/calamares/branding/meva-distro
```

Copia les imatges a la carpeta:

```
cp ~/distrothos-logo.png  
  ↪ config/includes.chroot/etc/calamares/branding/meva-distro/  
cp ~/benvinguda.png  
  ↪ config/includes.chroot/etc/calamares/branding/meva-distro/  
cp ~/diapositiva1.png  
  ↪ config/includes.chroot/etc/calamares/branding/meva-distro/  
cp ~/diapositiva2.png  
  ↪ config/includes.chroot/etc/calamares/branding/meva-distro/
```

Dins d'aquesta carpeta se sol definir un fitxer `branding.desc` amb el nom, els colors i els logotips propis. Vegeu a continuació un exemple complet:

`config/includes.chroot/etc/calamares/branding/meva-distro/branding.desc`

```
nano config/includes.chroot/etc/calamares/branding/meva-distro/brand  
  ↪ ing.desc
```

```
componentName: meva-distro  
  
welcomeStyleCalamares: true  
welcomeExpandingLogo: true  
  
strings:  
  productName:      "DistroThos"  
  shortProductName:  "DistroThos"  
  version:          "1.0"  
  shortVersion:     "1.0"  
  versionedName:    "DistroThos 1.0"  
  shortVersionedName: "DistroThos 1.0"  
  bootloaderEntryName: "DistroThos"  
  productUrl:       "https://www.iesthodicodina.cat/"  
  supportUrl:       "https://www.iesthodicodina.cat/suport"  
  releaseNotesUrl:  ↪ "https://www.iesthodicodina.cat/notes-versio"  
  
images:  
  productLogo:      "distrothos-logo.png"  
  productIcon:      "distrothos-logo.png"  
  productWelcome:   "benvinguda.png"
```

```

slideshow: "show.qml"

style:
  SidebarBackground:      "#2c3e50"
  SidebarText:            "#ffffff"
  SidebarTextCurrent:     "#3498db"
  SidebarBackgroundCurrent: "#34495e"

```

Camps més rellevants:

Camp	Funció
componentName	Ha de coincidir exactament amb el nom de la carpeta
strings.productName	Nom que apareix a totes les pantalles de l'instal·lador
strings.productUrl	Enllaç que es mostra a la pantalla de benvinguda
images.productLogo	Logotip a la barra lateral (recomanat: SVG o PNG amb fons transparent)
images.productWelcome	Imatge gran de la pantalla inicial
slideshow	Fitxer QML opcional que mostra una presentació animada mentre s'instal·la el sistema
style.sidebarBackground	Color de fons de la barra lateral de navegació (codi hexadecimal)

Finalment, cal indicar a Calamares quina marca ha d'utilitzar, mitjançant `config/includes.chroot/etc/calamares/settings.conf`:

```
nano config/includes.chroot/etc/calamares/settings.conf
```

```

modules-search: [ local ]
instances:
- id: debian
  module: license
  config: license.conf
sequence:
- show:
  - welcome
  - locale
  - keyboard
  - partition
  - users
  - summary
- exec:
  - partition
  - mount
  - unpackfs
  - machineid
  - fstab
  - locale
  - keyboard
  - localecfg
  - users
  - displaymanager

```

```

- networkcfg
- hwclock
- services-systemd
- initramfscfg
- initramfs
- bootloader
- umount
- show:
  - finished
branding: meva-distro
dont-chroot: false
oem-setup: false
disable-cancel: false
disable-cancel-during-exec: false
quit-at-end: false

```

NOTA

Aquest fitxer `settings.conf` s'ampliarà més endavant.

El fitxer `show.qml`: la presentació durant la instal·lació

El camp `slideshow` del `branding.desc` apunta a un fitxer escrit en **QML** (*Qt Modeling Language*), el llenguatge declaratiu del framework Qt (amb el qual està fet Calamares) per definir interfícies i animacions --- una mica com HTML+CSS però per a aplicacions d'escriptori. Aquest fitxer defineix la presentació de diapositives que es mostra mentre Calamares copia els fitxers al disc.

`config/includes.chroot/etc/calamares/branding/meva-distro/show.qml`

```

nano config/includes.chroot/etc/calamares/branding/meva-distro/show.
↩ qml

```

```

import QtQuick 2.5;
import calamares.slideshow 1.0;

Presentation {
    id: presentation

    Timer {
        interval: 5000
        running: true
        repeat: true
        onTriggered: presentation.goToNextSlide()
    }

    Slide {
        Image {
            source: "diapositiva1.png"
            anchors.centerIn: parent
        }
        Text {

```

```

        text: "Benvingut/da a DistroThos"
        font.pixelSize: 20
        color: "#ffffff"
        anchors.horizontalCenter: parent.horizontalCenter
        anchors.top: parent.verticalCenter
    }
}

Slide {
    Image {
        source: "diapositiva2.png"
        anchors.centerIn: parent
    }
    Text {
        text: "Pràctica per aprendre a crear una distro
↵ personalitzada"
        font.pixelSize: 20
        color: "#ffffff"
        anchors.horizontalCenter: parent.horizontalCenter
        anchors.top: parent.verticalCenter
    }
}
}

```

Aquest exemple defineix dues diapositives que canvien automàticament cada 5 segons (interval: 5000 mil·lisegons), cadascuna amb una imatge i un text centrats.

Punts clau:

- El fitxer s'ha de col·locar a la mateixa carpeta que `branding.desc`
- Les imatges que referencia (`diapositiva1.png`, `diapositiva2.png`) també han d'estar a la mateixa carpeta
- Calamares proporciona components predefinitos com `Presentation` i `Slide`, sense necessitat de saber QML a fons
- És totalment opcional: si s'omet el camp `slideshow` del `branding.desc`, Calamares mostra la pantalla de progrés sense diapositives

L'estructura de fitxers dins de la carpeta hauria de quedar així:

```

config/includes.chroot/etc/calamares/branding/meva-distro/
├── benvinguda.png
├── branding.desc
├── diapositiva1.png
├── diapositiva2.png
├── distrothos-logo.png
└── show.qml

```

NOTA

QML és un exemple de llenguatge declaratiu per a interfícies, un concepte que reapareix en molts contextos de desenvolupament (HTML/CSS, XAML...).

CONSELL

El fitxer `show.qml` és opcional --- si no en tens cap, pots ometre la línia `slideshow` i Calamares mostrarà la pantalla d'instal·lació sense presentació animada.

ACTIVITAT

Dissenya un logotip senzill (per exemple amb Inkscape) per a la teva distro i crea el teu propi branding.desc. Comprova que el nom i el logotip apareixen correctament a totes les pantalles de l'instal·lador.

10.3. Configura el fus horari per defecte de l'instal·lador

La secció 9.5 fixa el fus horari del sistema **live**, però l'instal·lador Calamares té la seva pròpia pantalla de regió/idioma amb un mapa on l'usuari pot triar el fus horari, i aquesta pantalla parteix d'un valor per defecte independent. Aquest valor es configura al mòdul `locale` de Calamares (el fus horari es va fusionar dins d'aquest mòdul; ja no existeix un `timezone.conf` separat).

config/includes.chroot/etc/calamares/modules/locale.conf

```
mkdir -p config/includes.chroot/etc/calamares/modules
nano config/includes.chroot/etc/calamares/modules/locale.conf
```

```
---
region: "Europe"
zone: "Madrid"
useSystemTimezone: false
```

- `region / zone` --- defineixen el “pin” inicial al mapa (per defecte, si no es posen, és `America/New_York`)
- `useSystemTimezone: false` --- evita que Calamares agafi el fus horari del sistema live en lloc del valor fixat aquí

10.4. Personalitza la drecera d'instal·lació nativa

El paquet `calamares` inclou el seu propi mecanisme per mostrar una icona d'instal·lació a l'escriptori/vista general a cada arrencada --- no cal crear-ne un de nou. Aquest mecanisme funciona així:

1. `/etc/xdg/autostart/calamares-desktop-icon.desktop` s'executa a l'inici de la sessió
2. Crida l'script `add-calamares-desktop-icon`, que copia el fitxer font `/usr/share/applications/calamares-install-debian.desktop` a l'escriptori de l'usuari

Aquest fitxer font és el que mostra “Install Debian” per defecte. Sobreescriu-lo amb el nom i la icona propis:

config/includes.chroot/usr/share/applications/calamares-install-debian.desktop

```
mkdir -p config/includes.chroot/usr/share/applications
nano config/includes.chroot/usr/share/applications/calamares-install-
↳ debian.desktop
```

```
[Desktop Entry]
Type=Application
Version=1.0
Name=Instal·la DistroThos
GenericName=Instal·lador de DistroThos
Exec=calamares-install-debian
Comment=Instal·lador per a DistroThos Live
Keywords=calamares;system;install;distrothos;installer
Icon=calamares
Terminal=false
Categories=Qt;System;
StartupWMClass=calamares
StartupNotify=True
```

AVÍS

Mantén `Exec=calamares-install-debian` tal com és --- és un script propi de Debian que gestiona permisos i comprovacions prèvies abans de llançar Calamares. No cal (ni convé) substituir-lo per `pkexec calamares` ni per `sudo calamares`.

Com que aquest mecanisme ja genera automàticament la icona a l'escriptori a cada arrencada, un cop personalitzat aquest fitxer **no cal cap llançador manual addicional** --- evites tenir dues dreceres diferents fent la mateixa funció.

10.5. Elimina la icona d'instal·lació un cop instal·lat el sistema

Quan Calamares copia el sistema de fitxers al disc, ho copia **tot**, inclosos els paquets `calamares` i `calamares-settings-debian` --- per això la icona "Instal·la DistroThos" continua apareixent fins i tot en el sistema ja instal·lat, no només a la sessió live. Cal dir-li explícitament a Calamares que elimini aquests paquets del sistema final un cop acabada la instal·lació.

Crea la configuració del mòdul packages:

```
mkdir -p config/includes.chroot/etc/calamares/modules
nano config/includes.chroot/etc/calamares/modules/packages.conf
```

```
---
backend: apt
operations:
- try_remove:
- calamares
```

```
- calamares-settings-debian
update_db: true
```

NOTA

`try_remove` (en lloc de `remove`) evita que la instal·lació falli si algun d'aquests paquets és, per algun motiu, dependència d'un altre --- simplement s'ometrà si no es pot eliminar amb seguretat.

Afegeix packages a la seqüència de `settings.conf`, abans de `initramfscfg/initramfs` (perquè la regeneració de l'`initramfs` ja reflecteixi els paquets eliminats):

```
nano config/includes.chroot/etc/calamares/settings.conf
```

```
- exec:
  - partition
  - mount
  - unpackfs
  - machineid
  - fstab
  - locale
  - keyboard
  - localecfg
  - users
  - displaymanager
  - networkcfg
  - hwclock
  - services-systemd
  - packages
  - initramfscfg
  - initramfs
  - bootloader
  - umount
```

AVÍS

No treguis `live-boot` ni `live-config` amb aquest mateix mòdul --- són molt més delicats de retirar correctament d'un sistema ja instal·lat (afecten l'`initramfs` i el procés d'arrencada). Eliminar `calamares` i `calamares-settings-debian` ja n'hi ha prou: com que la icona ve del fitxer `.desktop` i de l'script `add-calamares-desktop-icon`, tots dos propietat del paquet `calamares`, en desinstal·lar-lo desapareixen del sistema instal·lat i la icona ja no torna a aparèixer.

10.6. Elimina l'usuari de la sessió live i assigna l'avatar de l'usuari instal·lat

Pel mateix motiu que la icona d'instal·lació (secció 10.5), l'usuari alumne que vam crear al hook per a la sessió live (secció 8) també es copia sencer al sistema instal·lat --- per això continua apareixent a la pantalla de GDM al costat de l'usuari real creat durant la instal·lació.

Crea l'script que assigna l'avatar (més robust i llegible que una comanda inline dins el YAML):

config/includes.chroot/usr/local/sbin/set-user-avatar.sh

```
mkdir -p config/includes.chroot/usr/local/sbin
nano config/includes.chroot/usr/local/sbin/set-user-avatar.sh
```

```
#!/bin/sh
set -e

# Un cop eliminat 'alumne' (secció anterior), l'única carpeta que
↪ queda
# a /home/ és la de l'usuari real creat durant la instal·lació.
USERNAME="$(ls /home | head -n1)"

if [ -n "$USERNAME" ]; then
    mkdir -p /var/lib/AccountsService/icons
    ↪ /var/lib/AccountsService/users
    cp /usr/share/desktop-base/debian-logos/avatar.png
    ↪ "/var/lib/AccountsService/icons/$USERNAME"
    chmod 644 "/var/lib/AccountsService/icons/$USERNAME"
    printf '[User]\nIcon=/var/lib/AccountsService/icons/%s\n'
    ↪ "$USERNAME" \
        > "/var/lib/AccountsService/users/$USERNAME"
fi
```

```
chmod +x config/includes.chroot/usr/local/sbin/set-user-avatar.sh
```

Crea la configuració del mòdul shellprocess:

```
mkdir -p config/includes.chroot/etc/calamares/modules
nano config/includes.chroot/etc/calamares/modules/shellprocess.conf
```

```
---
dontChroot: false
timeout: 30
script:
  - "userdel -r alumne || true"
  - "/usr/local/sbin/set-user-avatar.sh"
```

- **dontChroot: false** --- executa l'ordre **dins** del sistema ja instal·lat (el chroot final), no a la sessió live

- `|| true ---` evita que la instal·lació falli si per algun motiu l'usuari ja no existeix o l'eliminació no és possible
- L'ordre importa: `set-user-avatar .sh` s'ha d'executar **després** d'eliminar a l'umne, ja que es basa en el fet que només quedi una carpeta a `/home/`

Afegeix `shellprocess` a la seqüència de `settings.conf`, després de `packages`:

```
nano config/includes.chroot/etc/calamares/settings.conf
```

```
- exec:
- partition
- mount
- unpackfs
- machineid
- fstab
- locale
- keyboard
- localecfg
- users
- displaymanager
- networkcfg
- hwclock
- services-systemd
- packages
- shellprocess
- initramfscfg
- initramfs
- bootloader
- umount
```

NOTA

Cal que `shellprocess` vagi **després** de `users` (el mòdul que crea l'usuari real durant la instal·lació), per assegurar que ja hi ha algú amb qui iniciar sessió abans d'eliminar a l'umne.

10.7. Personalitza la fortalesa de la contrasenya de l'usuari

```
mkdir -p config/includes.chroot/etc/calamares/modules
nano config/includes.chroot/etc/calamares/modules/users.conf
```

```
---
userGroup:      users
defaultGroups:
- cdrom
- floppy
- sudo
- audio
- dip
```

```
- video
- plugdev
- netdev
- lpadmin
- scanner
- bluetooth
autologinGroup: autologin
sudoersGroup: sudo
setRootPassword: false
passwordRequirements:
  nonempty: true
  minLength: 1 # Password at least this many characters
  maxLength: -1 # Password at most this many characters
  libpwquality:
    - minlen=6
    - minclass=0
allowWeakPasswords: true
allowWeakPasswordsDefault: true
```

11. Construeix l'ISO

AVÍS

El procés de construcció pot trigar molt (des de 15 minuts fins a més d'una hora) segons la velocitat de connexió i la quantitat de paquets triats. **No l'interrompis mentre s'executa `lb build`.**

```
sudo lb build
```

O bé, si vols, pots generar un *log* de la construcció:

```
sudo lb build 2>&1 | tee build.log
```

Durant el procés, `live-build`:

1. Crea el sistema base amb `debootstrap`
2. Instal·la els paquets de totes les llistes definides
3. Copia els fitxers d'`includes.chroot`
4. Executa els *hooks*
5. Genera la imatge ISO final

Sortida:

```
...
...
...
[2026-08-11 14:40:52] lb chroot_sysfs remove
P: Begin unmounting /sys...
[2026-08-11 14:40:52] lb chroot_selinuxfs remove
```

```
[2026-08-11 14:40:52] lb chroot_proc remove
P: Begin unmounting /proc...
[2026-08-11 14:40:52] lb chroot_devpts remove
P: Begin unmounting /dev/pts...
P: Binary stage completed
P: Begin unmounting filesystems...
P: Saving caches...
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
[2026-08-11 14:40:52] lb source
P: Source stage disabled, skipping
P: Build completed successfully
ramon@debian:~/la-meva-distro$
```

Si tot ha anat bé, trobaràs el fitxer `live-image-amd64.hybrid.iso` a l'arrel del projecte.

```
ramon@debian:~/la-meva-distro$ ls -lh live-image-amd64.hybrid.iso
-rw-r--r-- 1 root root 2,6G 11 d'ag.    14:26
↪ live-image-amd64.hybrid.iso
ramon@debian:~/la-meva-distro$
```

12. Prova l'ISO

Sempre abans de gravar-la físicament, és recomanable provar-la en una màquina virtual:

```
qemu-system-x86_64 -m 2048 -boot d -cdrom live-image-amd64.hybrid.iso
```

O bé important-la com a ISO d'arrencada a [VirtualBox](#).

Pots [descarregar l'ISO de DistroThos Sid 1.0](#) que s'ha generat en aquest document des d'aquest enllaç

Les credencials d'usuari són alumne:alumne

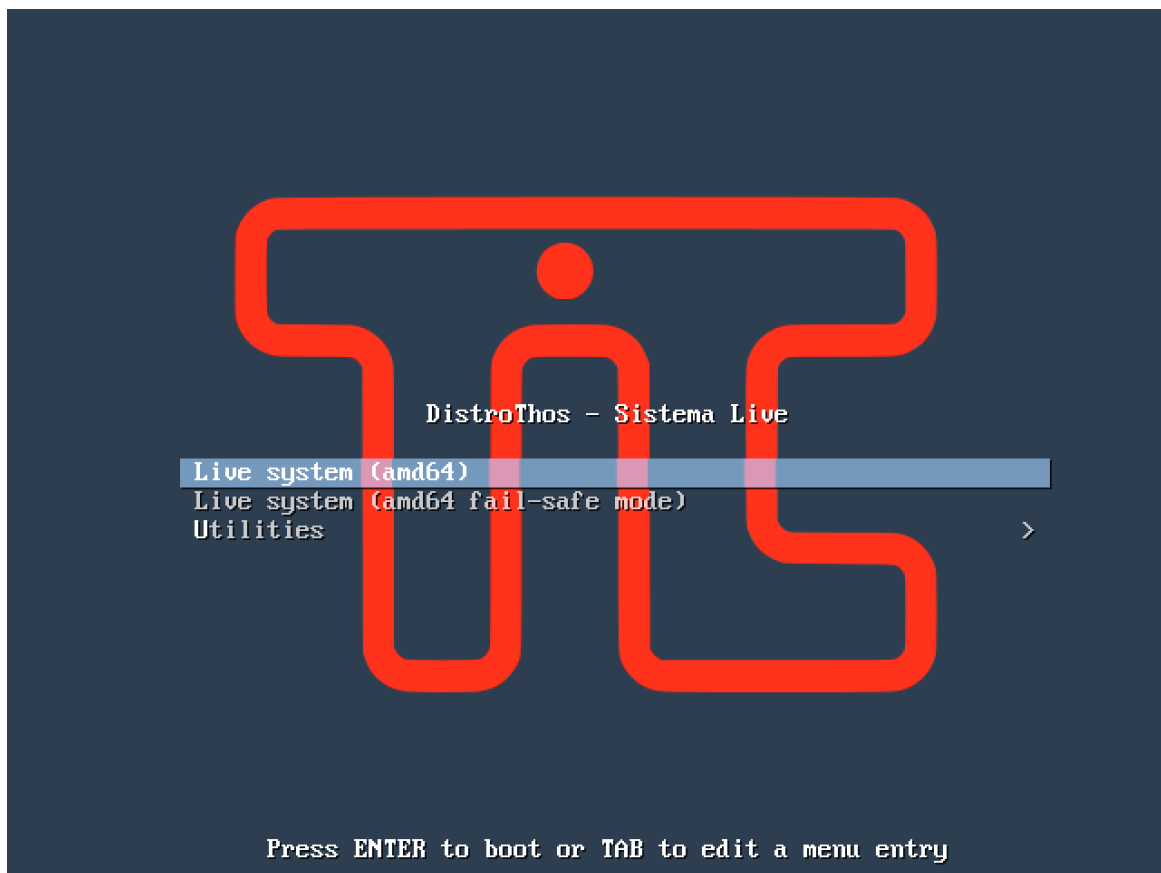


Figura 2: DistroThos live

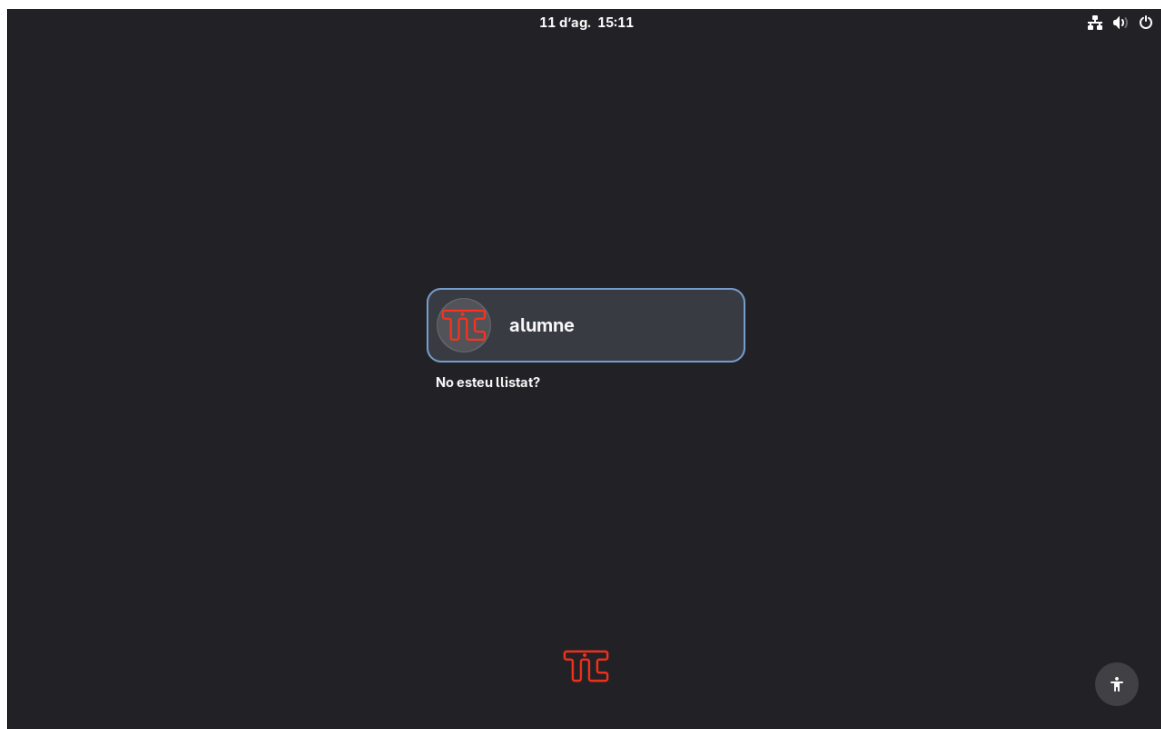


Figura 3: Login de l'usuari alumne

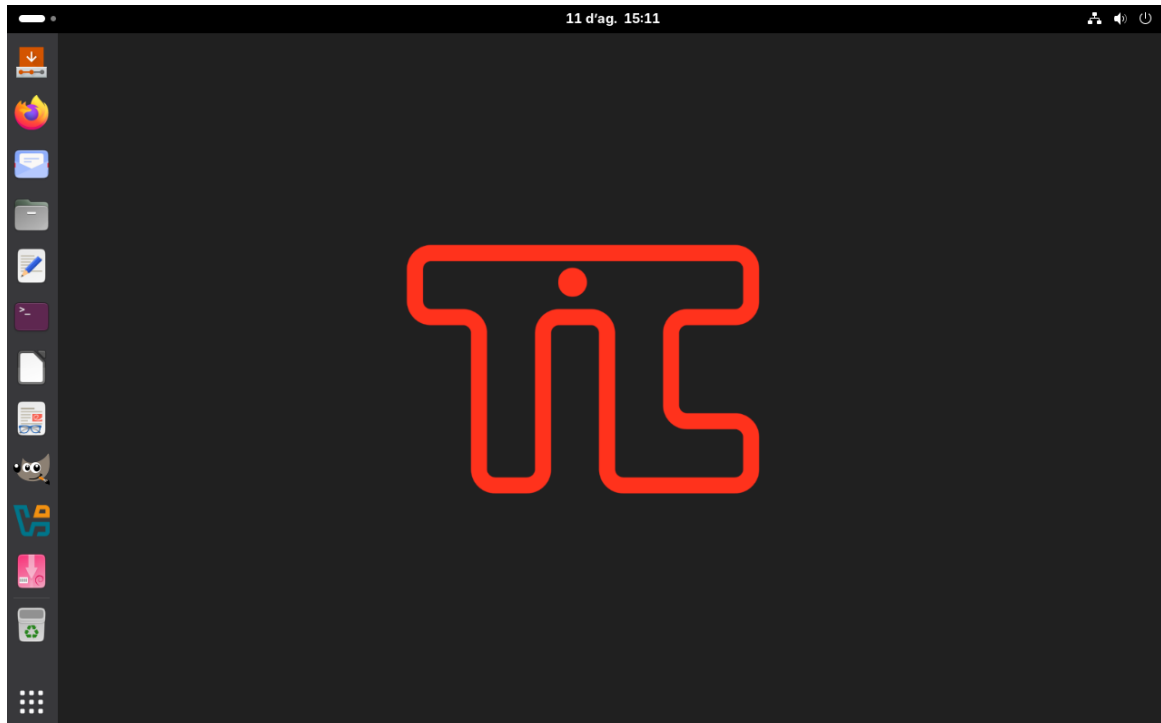


Figura 4: Escriptori

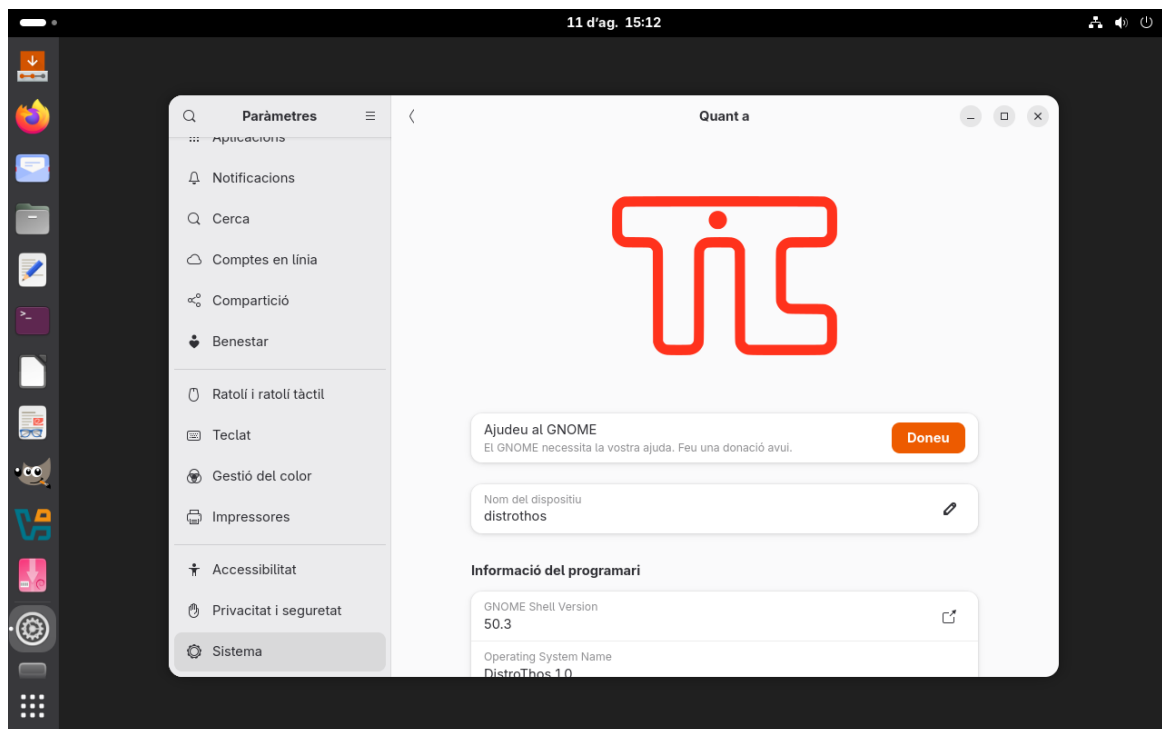


Figura 5: Quant a



Figura 6: fastfetch

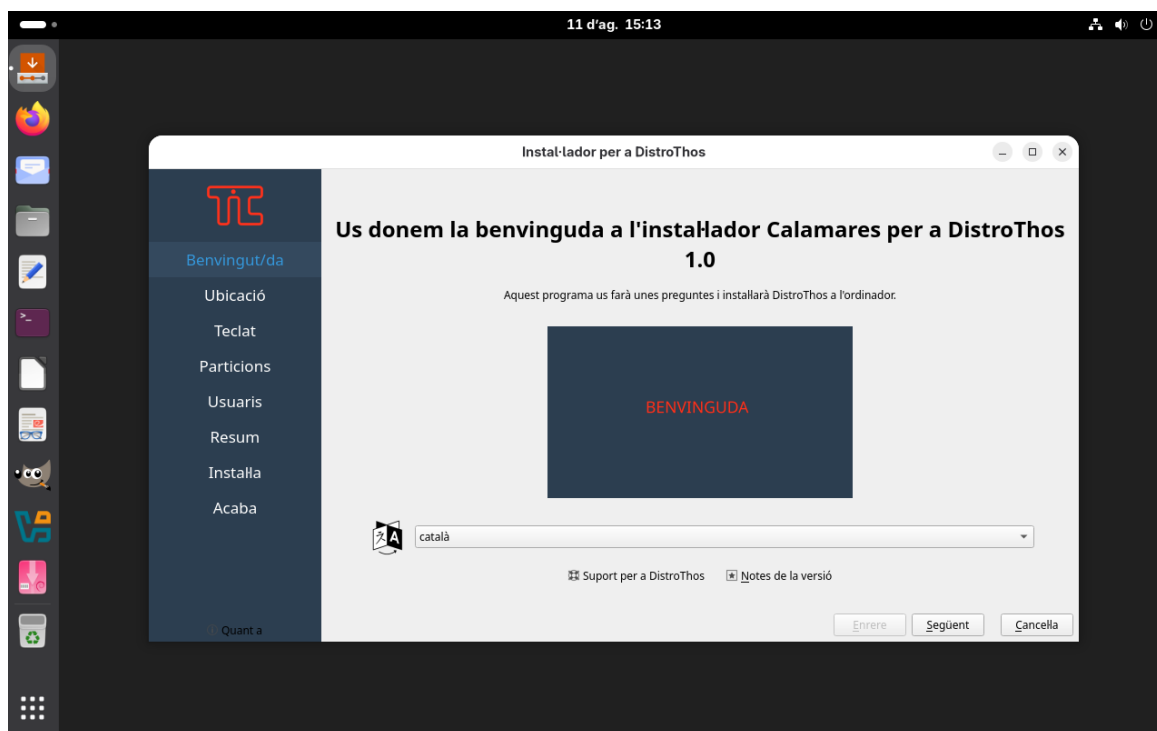


Figura 7: Instal·lador - Benvinguda

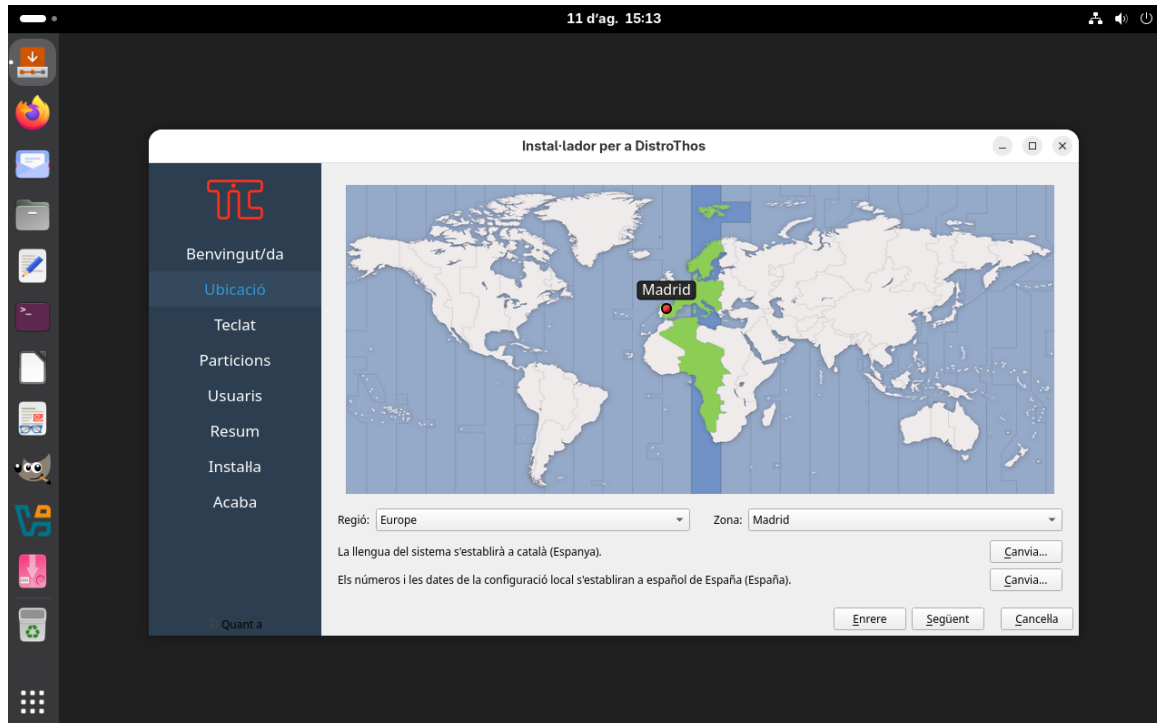


Figura 8: Instal·lador - Fus horari

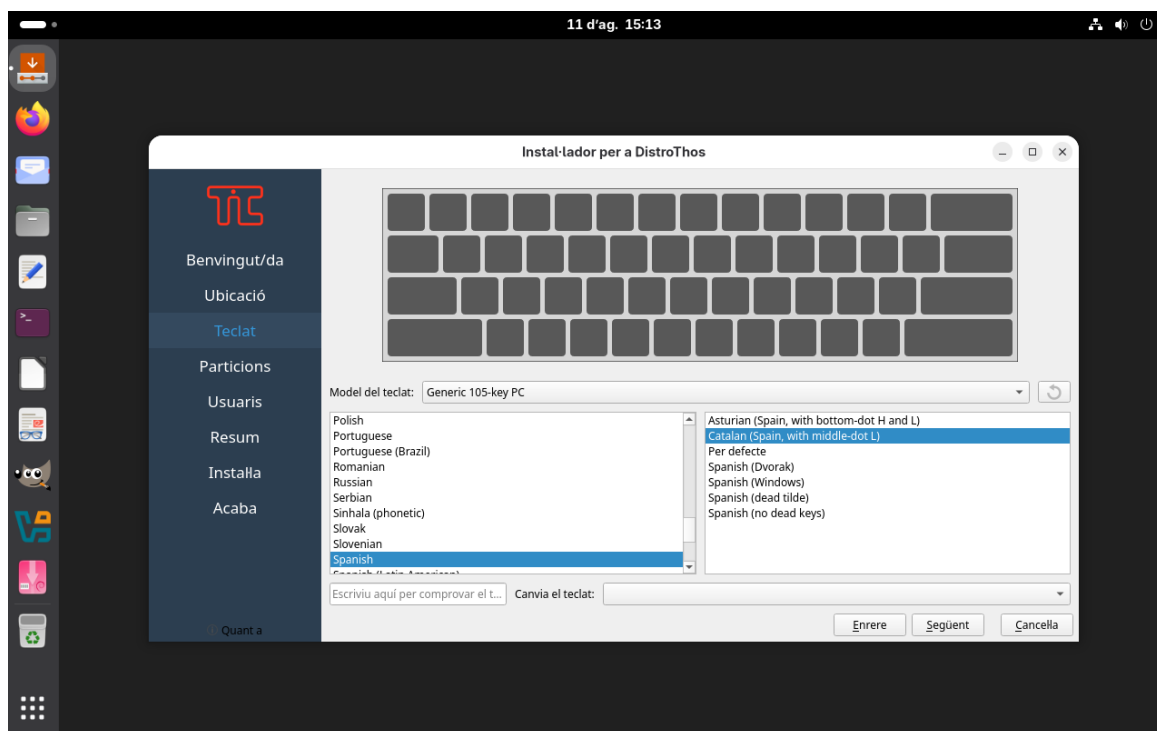


Figura 9: Instal·lador - Teclat

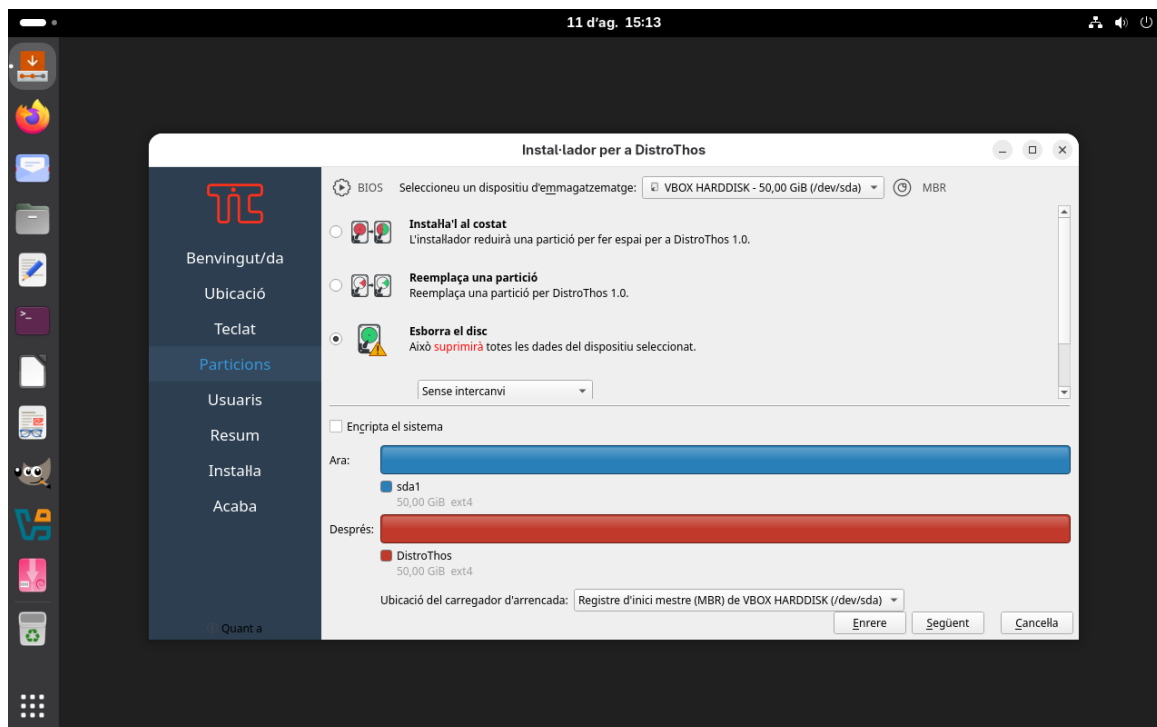


Figura 10: Instal·lador - Particions

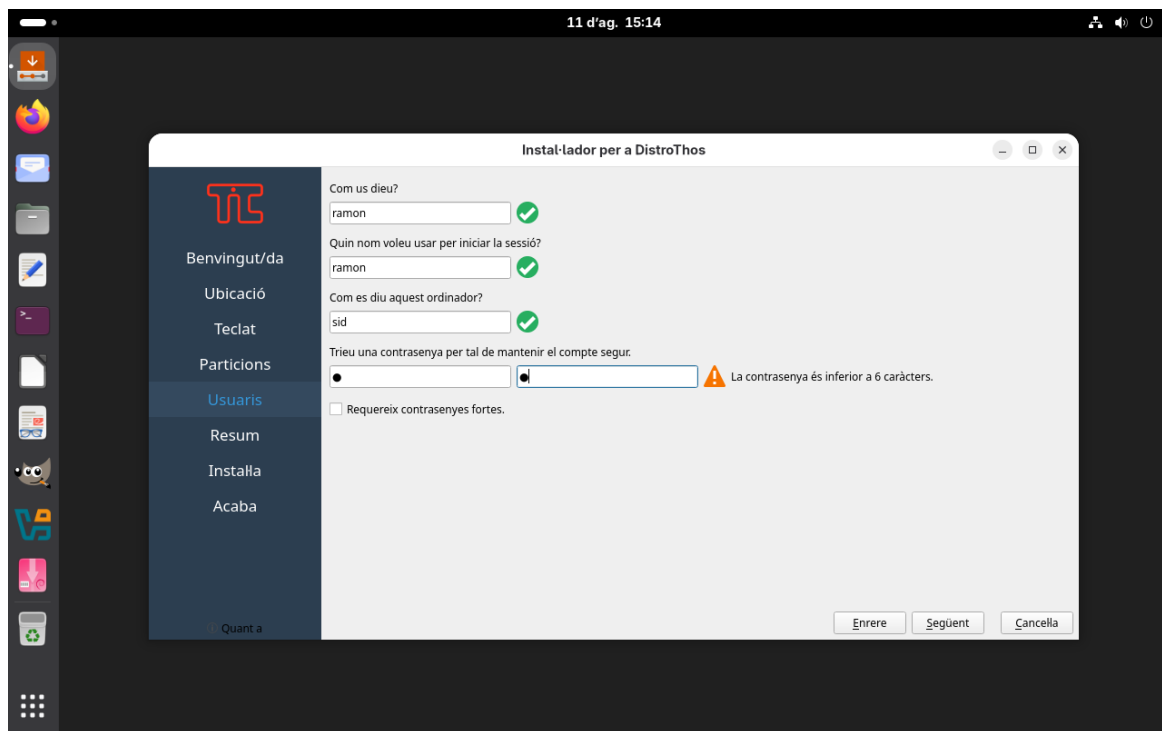


Figura 11: Instal·lador - Usuari

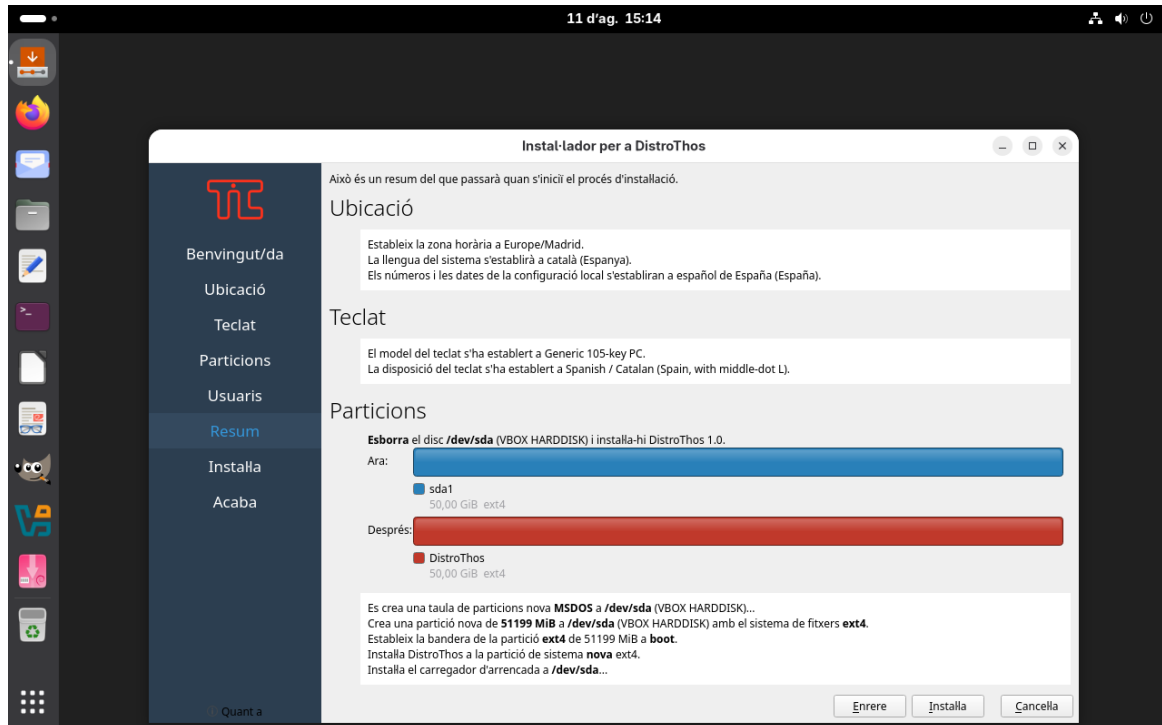


Figura 12: Instal·lador - Resum

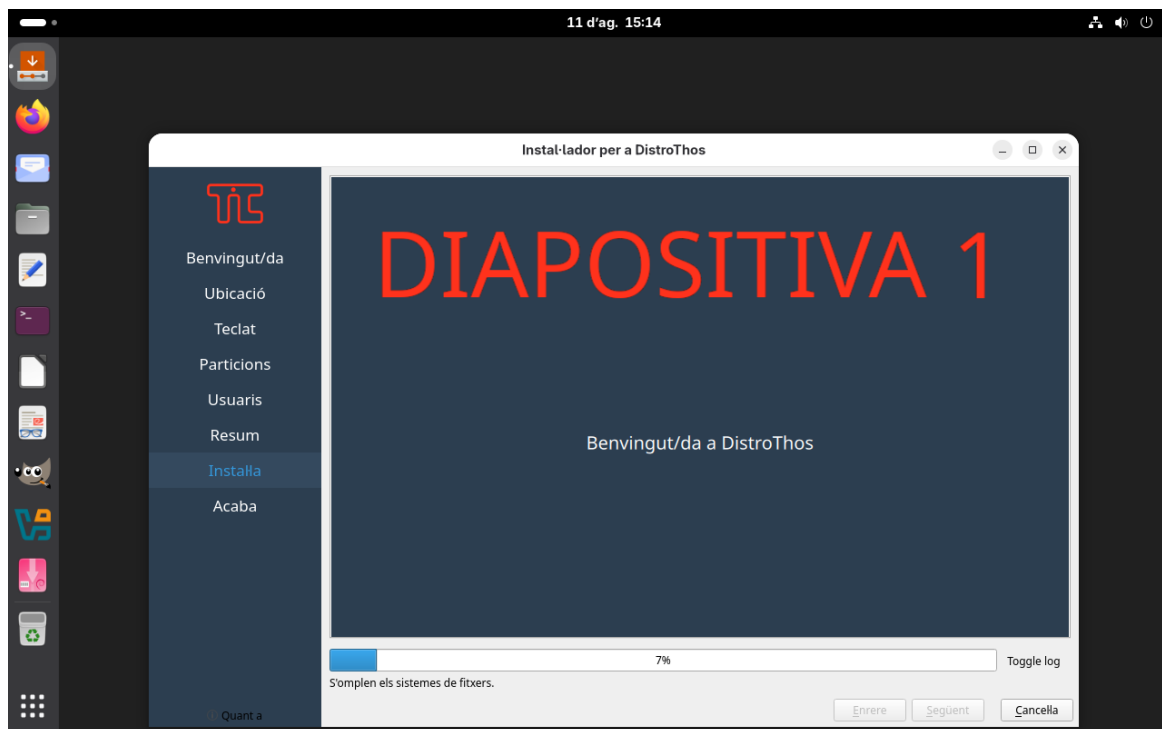


Figura 13: Instal·lador - Diapositiva 1

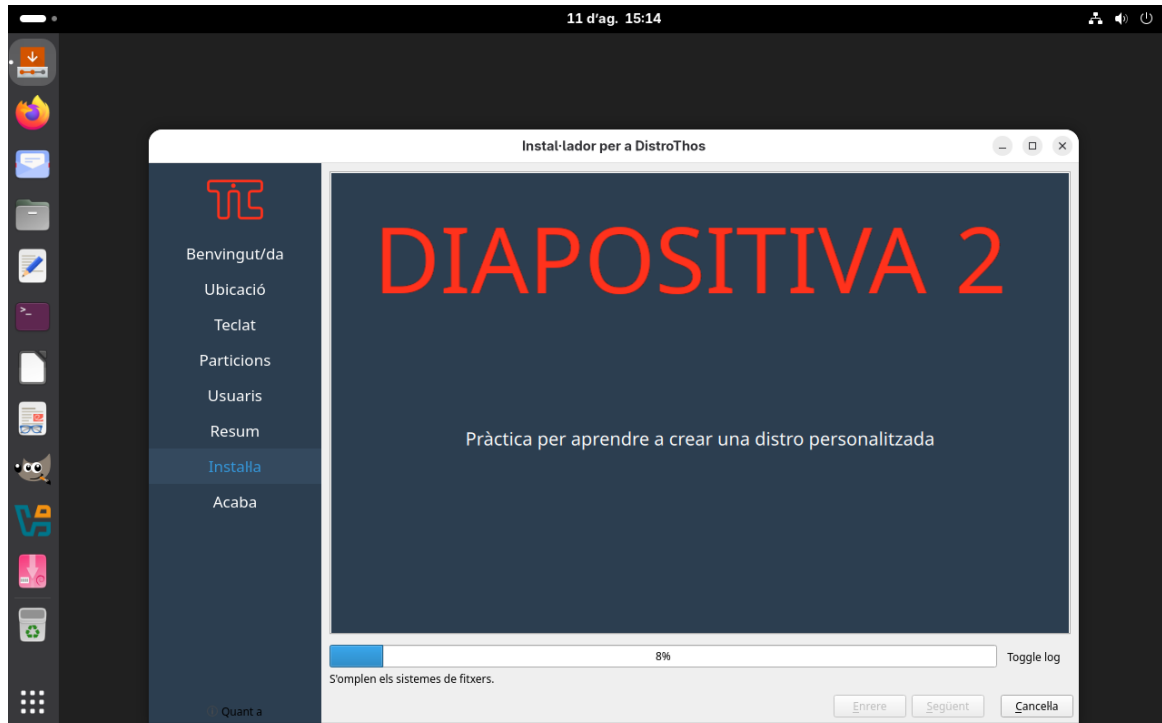


Figura 14: Instal·lador - Diapositiva 2

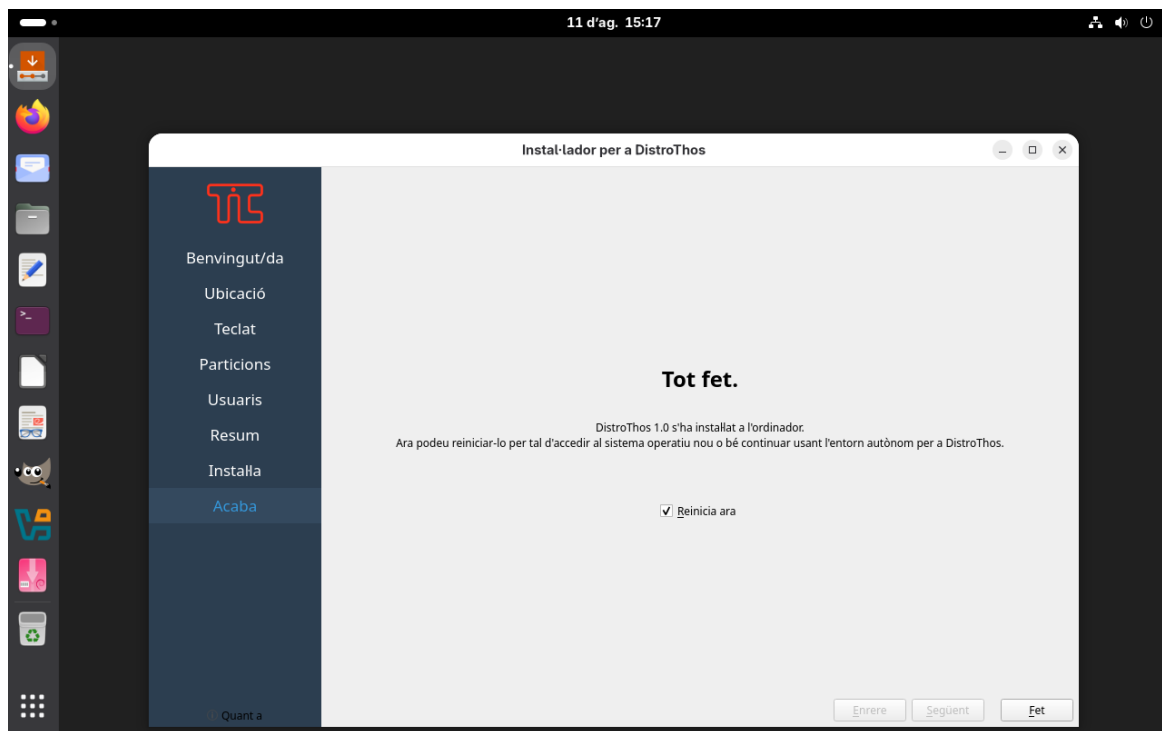


Figura 15: Instal·lador - Acaba

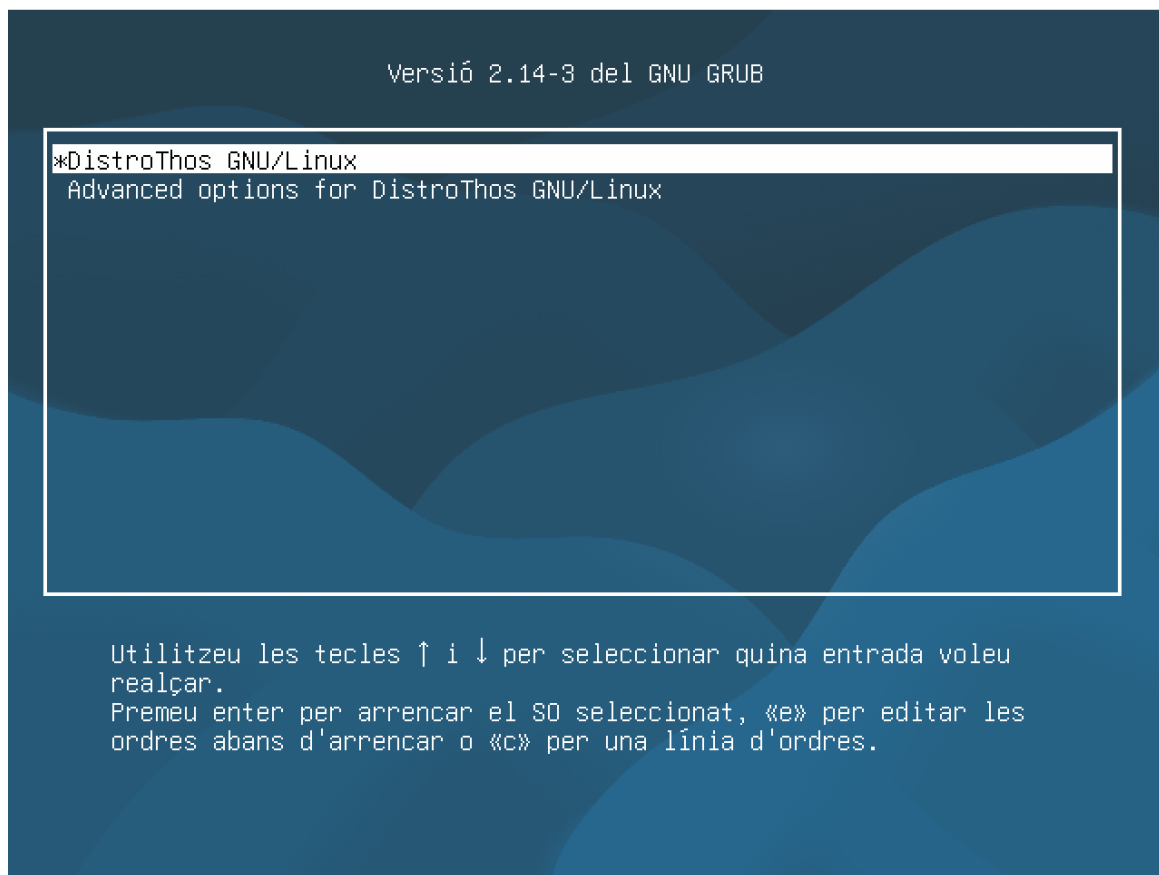


Figura 16: GRUB de DistroThos

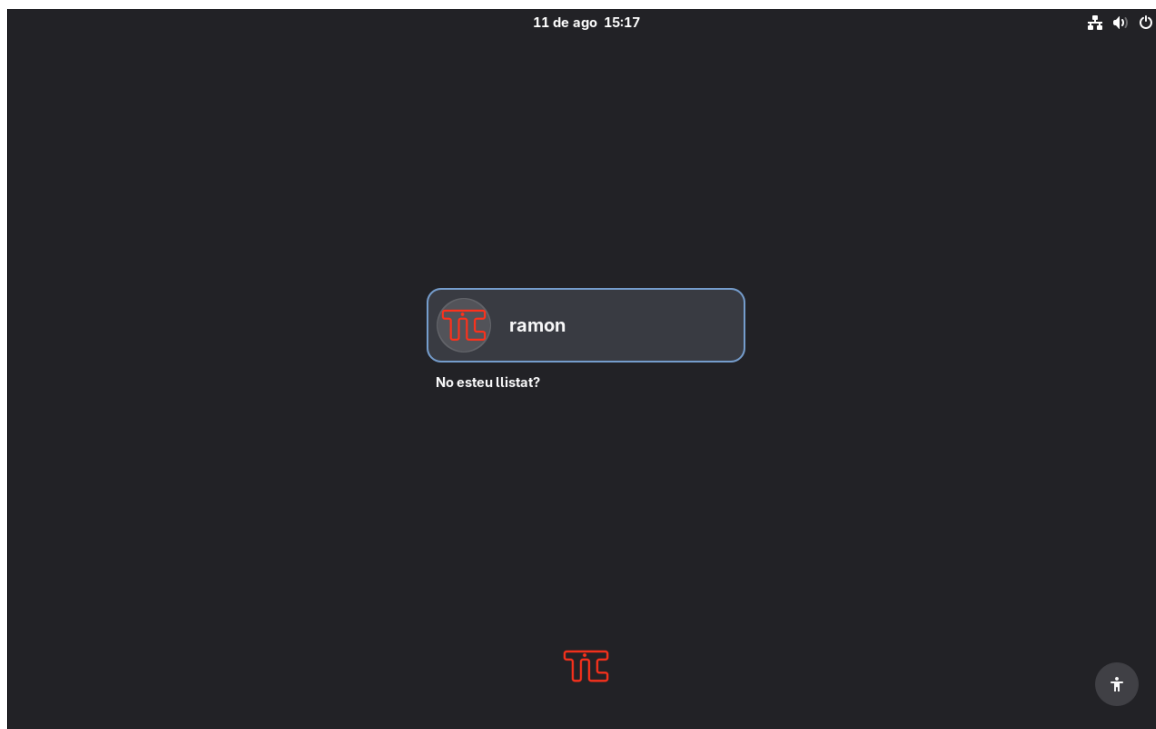


Figura 17: Login del nou usuari

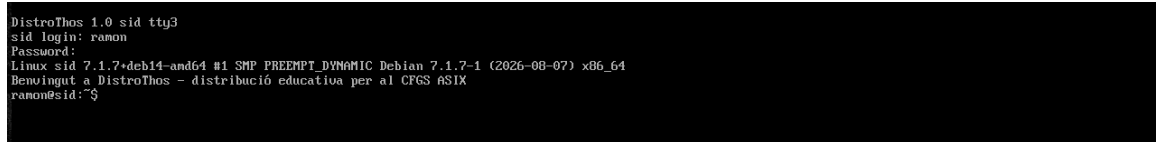


Figura 18: Inici de sessió a tty3

13. Neteja i reconstrueix

Quan es modifica la configuració (paquets, hooks, includes) cal netejar abans de tornar a construir:

```
sudo lb clean
```

Sincronitza:

```
lb config \  
  --distribution sid \  
  --distribution-chroot sid \  
  --distribution-binary sid \  
  --architectures amd64 \  
  --binary-images iso-hybrid \  
  --binary-images iso-hybrid
```

```
--archive-areas "main contrib non-free non-free-firmware" \
--firmware-chroot false \
--bootappend-live "boot=live components locales=ca_ES.UTF-8
↳ keyboard-layouts=es keyboard-variants=cat
↳ timezone=Europe/Madrid" \
--iso-application "DistroThos" \
--iso-publisher "Ramon López" \
--iso-volume "DistroThos"
```

Reconstrueix:

```
sudo lb build
```

Per a una neteja completa, incloent-hi la memòria cau de paquets descarregats:

```
sudo lb clean --purge
```

Ordre	Efecte
lb clean	Neteja parcial: elimina el chroot i la imatge generada
lb clean --purge	Neteja total: també elimina la memòria cau de paquets

Resum del flux de treball

```
lb config          ↳ Configuració inicial
package-lists/     ↳ Tria de paquets
includes.chroot/   ↳ Fitxers propis
dconf              ↳ Wallpaper i logo
desktop-base       ↳ Logo real a "Quant a"
config/bootloaders/ ↳ Personalitza el menú d'arrencada
hooks/live/        ↳ Automatització
debconf-set-selections ↳ Idioma, teclat i locale (català)
calamares          ↳ Instal·lador (opcional)
calamares/packages.conf ↳ Neteja calamares del sistema instal·lat
calamares/shellprocess.conf ↳ Elimina l'usuari live del sistema
↳ instal·lat
lb build           ↳ Construcció de l'ISO
lb clean           ↳ Neteja per reconstruir
lb config          ↳ Repetir després de lb clean (evita
↳ l'error "stage config")
```

ACTIVITAT

Construeix la teva ISO personalitzada, arrenca-la en una màquina virtual i documenta, amb captures de pantalla, els canvis respecte a una instal·lació estàndard de Debian (paquets afegits, usuari per defecte, wallpaper, etc.).

Referències

- Documentació oficial de Debian Live: <https://live-team.pages.debian.net/live-manual/>
- Manual de live-build: `man lb_config`, `man lb_build`
- Documentació oficial de Calamares: <https://calamares.codeberg.page/docs/documentation/>
- Guia de personalització de la marca (branding) de Calamares: <https://codeberg.org/Calamares/calamares/src/branch/calamares/src/branding/README.md>
- Dipòsit del projecte Calamares a Codeberg: <https://codeberg.org/Calamares/calamares>

Índex de figures

1	isolinux live	9
2	DistroThos live	34
3	Login de l'usuari alumne	35
4	Escriptori	35
5	Quant a	36
6	fastfetch	36
7	Instal·lador - Benvinguda	37
8	Instal·lador - Fus horari	37
9	Instal·lador - Teclat	38
10	Instal·lador - Particions	38
11	Instal·lador - Usuari	39
12	Instal·lador - Resum	39
13	Instal·lador - Diapositiva 1	40
14	Instal·lador - Diapositiva 2	40
15	Instal·lador - Acaba	41
16	GRUB de DistroThos	42
17	Login del nou usuari	43
18	Inici de sessió a tty3	43

Versions d'aquest document

- [HTML](#)
- [PDF](#)
- [ODT](#)
- [MD](#)

[Domini Públic \(CC0\)](#)