

The Open Source Project
Pau Cruz y Stephany Gamito



Institut Puig Castellar
Santa Coloma de Gramenet



**THE OPEN SOURCE
PROJECT**

The Open Source Project

Proyecto de desarrollo

CFGM Sistemas Microinformáticos y Redes

Stephany Gamito y Pau Cruz

2SMXA

2025/2026

Tutor: Rubén Arroyo

Fecha de entrega: 17/05/2026



Esta obra está sujeta a una licencia de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Copyright © 2026 The Open Source Project

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Índice

1. Introducción	1
1.1 Contexto	1
1.2 Justificación	2
1.3 Objetivos	2
1.4 Estrategia y planificación del proyecto	2
2. Descripción del proyecto	4
2.1 Análisis de requisitos de The Open Source Project	4
2.2 Tecnologías	4
2.3 Estructura del proyecto	5
2.4 Descripción de los componentes	8
2.5 Definición de las funcionalidades	13
3. Configuración de servicios	14
3.1. Kea (DHCP)	14
3.2. Bind9 (DNS)	16
3.3. Nextcloud (almacenamiento en la nube)	18
3.4. Matrix (Mensajería instantánea)	20
3.4. Gateway	26
4. Conclusiones	27
4.1 Conclusiones generales del proyecto	27
4.2 Consecución de los objetivos	27
4.3 Valoración de la metodología y planificación	28
4.4 Visión de futuro	28
4.5 Declaración del uso de inteligencia artificial	29
5. Glosario	30
6. Bibliografía	31
7. Anexos	32
7.1. Instalación de kea-dhcp4-server: configuración del archivo kea-dhcp4.conf	32
7.2. Instalación de Matrix: contenido de homeserver.yaml	33
7.3. Instalación de matrix: contenido de Caddyfile	34

Lista de figuras

Figura 1. Logotipo del proyecto.....	1
Figura 2. Logotipo de isardVDI.....	5
Figura 3. Logotipo de apache2.....	6
Figura 4. Logotipo de caddy.....	7
Figura 5. Logotipo de nextcloud.....	7
Figura 6. Logotipo de kea.....	7
Figura 7. Logotipo de matrix.....	8
Figura 8. Logotipo de la página web.....	8
Figura 9. Estructura de la red interna.....	9
Figura 10. Ventana que aparece al instalar matrix-synapse para definir el nombre del dominio.....	22
Figura 11. Mensaje que aparece al entrar en la URL del bastion de isard al acabar de configurar matrix.....	25
Figura 12. Sign-in de element donde se indica que la cuenta se aloja en nuestro servidor matrix.....	26

Resumen del proyecto

- *The Open Source Project es un proyecto que fomenta el uso de programas de código abierto como alternativa real a productos ofrecidos por grandes empresas como Meta o Google para garantizar una mayor privacidad y control sobre los datos de los usuarios.*
- *El objetivo del proyecto surge de las necesidades de tener programas libres de anuncios y suscripciones de pago, donde los datos no se alojan en servidores de terceros, y donde no se obtienen beneficios vendiendo los datos de los usuarios, ya que creemos en el control y privacidad total de los mismos.*
- *Hemos decidido reemplazar en su totalidad los productos más comunes, como Google Drive o Mega, por nextcloud; el DNS de Google o tu ISP por Bind9 y servicios de mensajería instantánea como WhatsApp o Discord por matrix.*
- *Principalmente, nuestro objetivo es crear un ecosistema completo donde los datos nunca salgan de las máquinas donde se alojan, eliminando vigilancia, anuncios y dependencia de terceros.*

Palabras clave

Código abierto

Autoalojado

Privacidad de datos

Soberanía digital

Alternativa a servicios de Google

Servicios descentralizados

Enfoque en privacidad

Independencia tecnológica

Abstract

The Open Source Project promotes the use of open-source software as a real alternative to products offered by large corporations like Meta or Google to ensure greater privacy and control over user data. The project's objective stems from the need for programs free of ads and paid subscriptions, where data is not hosted on third-party servers, and no profit is made from selling user data, as we believe in total control and privacy. We have decided to completely replace the most common products, such as Google Drive or Mega, with Nextcloud, Google DNS or your Internet provider's DNS with Bind9... Additionally, we will add our own DHCP server among other functions. Our goal is to create a complete ecosystem where data never leaves our servers, eliminating surveillance, ads, and third-party dependence.

Keywords

Open source

Self-hosted

Data privacy

Digital sovereignty

Google alternative

Decentralized services

Privacy-focused

Server independence

1. Introducción

The Open Source Project es una iniciativa que promueve el software de código abierto como respuesta ética, transparente y segura a las soluciones ofrecidas por grandes corporaciones. Frente a empresas como Meta o Google (que recopilan, comercializan y utilizan nuestros datos personales para un sinfín de usos, desde venderlos hasta entrenar sus propias IAs), nuestro objetivo es devolver el control a los usuarios mediante tecnologías transparentes, respetuosas con la privacidad y desarrolladas por la comunidad.

El proyecto permite desplegar una red local completa formada enteramente por software libre, eliminando la dependencia de servicios en la nube controlados por terceros. Para ello utilizaremos IsardVDI, una plataforma de virtualización de escritorios, donde configuraremos máquinas con aplicaciones y servicios libres. Además, emplearemos clientes finales para demostrar que esta infraestructura funciona de forma integral y autónoma.



Figura 1. Logotipo del proyecto

1.1 Contexto

Como bien sabemos, la información de los usuarios es prácticamente un diamante en bruto para las empresas, y, como hemos mencionado anteriormente, estas se dedican a recolectar datos de los usuarios, ya que la mayoría aceptan los términos y condiciones sin leer una sola línea, aprovechándose de ello, y sin que lo sepas, venden o utilizan toda aquella información recolectada para el beneficio propio de la empresa.

Al ver toda esta situación desde fuera, nos ha inspirado a salir adelante, plantar cara y ofrecer alternativas con funciones prácticamente idénticas, utilizando opciones de código abierto para asegurar que nuestros servicios sean totalmente transparentes y nadie obtenga un solo euro de tus datos de navegación, intereses o información personal.

1.2 Justificación

Nuestro objetivo es que nuestro proyecto permita a los usuarios tener más privacidad al no depender de empresas externas para gestionar sus archivos y su información. Lo vemos necesario, ya que hoy en día la privacidad es prácticamente nula, especialmente cuando dependes de terceros, por lo que queremos promover el uso de aplicaciones de código abierto que uno mismo puede montar en su casa, centro educativo o empresa, y que te permiten tener el control total sobre qué pasa con tus datos e información.

1.3 Objetivos

Nuestros objetivos son:

- Montar una infraestructura de red con los servicios necesarios para ser útil.
- Preservar la privacidad de los usuarios.
- No depender de empresas que alojan sus servicios en sus propios servidores.
- Promover una alternativa honesta, real y transparente a los usuarios.

1.3.1 Objetivo general

El objetivo se basa en montar una infraestructura de red con todo lo necesario para poder utilizarla en un centro educativo o empresa.

1.3.2 Objetivos específicos

Proporcionar una serie de aplicaciones en la red que aseguren la privacidad de los datos de los usuarios, como servicios de mensajería o clouds.

1.4 Estrategia y planificación del proyecto

Para este proyecto, nuestra idea es instalar servicios que ya vienen más o menos montados para la red (DHCP, DNS...) y para las aplicaciones web, descargarlas, instalar y configurarlas manualmente, en nuestro caso, Nextcloud y Matrix, y todo esto lo montaremos en los clientes de Isard. Creemos que es lo más apropiado, ya que hacer estas aplicaciones de cero es complicado e innecesario, por lo que hemos optado por utilizar servicios de red que hemos aprendido a utilizar este año en el módulo “serveis de xarxa” y parte de nuestros conocimientos del módulo de aplicaciones web del año pasado para Nextcloud y Matrix, aunque para nosotros, Matrix es algo nuevo, por lo que tendremos que aprender cómo funciona y cómo usarlo.

Para la página web del proyecto, utilizaremos mayormente HTML y CSS, lenguajes de programación los cuales estuvimos manejando un poco el año pasado y que ya hemos visto antes, aunque también incluiremos JavaScript u otros lenguajes dependiendo de las necesidades que vayamos viendo.

1.5 Metodología de trabajo

Optamos por la metodología Scrum. Creemos que es mejor ir dividiendo las tareas “por partes”, ya que centrarnos constantemente en un solo objetivo podría llevarnos a distracciones innecesarias y, por lo tanto, eso nos llevaría a no prestarle atención a tareas más importantes. Por lo tanto, escogeremos la base de nuestro proyecto y de ahí iremos añadiendo cosas más específicas para poder manejarlo óptimamente.

Para el seguimiento de las tareas utilizaremos el diagrama de Gantt, ya que no nos parece necesario utilizar una aplicación para organizar nuestras tareas porque sabemos cómo darle prioridad a las más importantes.

1.6 Estudio económico y presupuestario

Actualmente, no necesitamos invertir dinero en material físico para el proyecto. En cuanto a las herramientas utilizadas, tampoco tendremos gastos, ya que todo lo que utilizaremos es software de código abierto o proporcionado por nuestro centro educativo.

2. Descripción del proyecto

2.1 Análisis de requisitos de The Open Source Project

Para dar por válido el proyecto, debe cumplir con los siguientes requisitos:

- Permitir a los usuarios conectarse a la red.
- Asignar automáticamente una dirección IP al usuario.
- Dar acceso a internet al usuario.
- Dar acceso a los servicios y aplicaciones instalados, como Matrix, Nextcloud, etc.
- Poder entrar a la página web del proyecto.

2.1.1 Requisitos funcionales

- Proporcionar conexión a internet a los usuarios de la red.
- Proporcionar una serie de aplicaciones en la red que aseguren la privacidad de los datos de los usuarios, como servicios de mensajería o almacenamiento en la nube.
- Garantizar el buen funcionamiento de las máquinas de Isard y las conexiones a internet.

2.1.2 Requisitos no funcionales

- El sistema debe asignar direcciones IP a los dispositivos que se conectan a la red.
- El sistema debe soportar la carga de los usuarios que se conectan y utilizan sus servicios.

2.2 Tecnologías

2.2.1 Tecnologías escogidas

- HTML5
- CSS
- JavaScript
- PHP
- bash

2.3 Estructura del proyecto

IsardVDI

Es la base de todo, será donde nuestras máquinas y todo lo que necesitamos para el proyecto estarán alojados, por lo que podríamos llamarlo el núcleo.

Dentro de las máquinas, cada una será creada con un propósito en específico. El proyecto no será alojado en una sola máquina virtual; esto lo hacemos así porque si una máquina se daña, tendríamos que empezar de cero; por lo tanto, hemos optado por alojar cada programa dentro de una máquina distinta de Isard. Así, un fallo no nos cuesta el proyecto entero.



Figura 2. Logotipo de isardVDI

Además, hemos decidido escoger Isard y no VirtualBox porque podemos trabajar en el proyecto independientemente de dónde estemos localizados, algo que juega a nuestro favor si decidimos trabajar desde casa o en otro ordenador. Esta ventaja no la podríamos tener con VirtualBox, ya que aloja las máquinas en local. Si bien podríamos optar por utilizar la VPN de la que disponemos en el centro, no lo vemos conveniente, ya que a la hora de presentar nos será más fácil tenerlo todo dentro de nuestras cuentas de Isard.

El sistema operativo que utilizaremos para nuestras máquinas es Ubuntu Server 24.04.4 LTS, ya que es estable, simple y eficiente; tiene todo lo necesario para nuestro proyecto. Si bien recientemente ha salido la versión 26.04, no vemos necesario cambiarlo todo, ya que nos podría generar problemas, y la versión 24.04.4 LTS se adapta perfectamente a nuestras necesidades actuales.

En IsardVDI, disponemos de plantillas de Ubuntu Desktop y Ubuntu Server, por lo que no deberemos preocuparnos por hacer la instalación de la misma máquina muchas veces.

Apache2

Apache2 es el servidor HTTP que utilizaremos para montar las aplicaciones web. Su función será alojar la nube y la página web del proyecto, por lo que es una parte esencial. La nube y la página web estarán en distintas máquinas, por lo



Figura 3. Logotipo de apache2

que cada una tendrá su propio servidor Apache2.

Caddy

Al igual que Apache2, Caddy es un servidor HTTP que nos permite obtener certificados de la organización *Let's Encrypt* más fácilmente, permitiéndonos hacer nuestras páginas públicas; y gracias a la función *bastión* de isardVDI no tenemos la necesidad de comprar un dominio. Utilizaremos Caddy en Nextcloud y en Matrix, aunque en la cloud pondremos Caddy por encima de Apache2.



Figura 4. Logotipo de caddy

Nextcloud

Nextcloud es una aplicación web de código abierto con el objetivo de ser un servicio de almacenamiento de archivos en la nube. Nuestro objetivo es montarlo en una máquina de isard y permitir a los usuarios de la red acceder para tener su propio espacio en la nube.



Figura 5. Logotipo de nextcloud

Nextcloud no solo ofrece almacenamiento en la nube, sino que también contiene un servicio de mensajería, llamadas y videoconferencias, por lo que puede ser usado para más que un simple espacio en la nube.

Kea

Es el servicio DHCP que utilizaremos para nuestra red. Se tratará de una máquina que asignará direcciones IP a los demás clientes. Si bien configuraremos IPs estáticas para las máquinas que actúen como DNS, cloud y demás mediante su propio netplan, en el caso de los clientes, se les asignará una IP aleatoria.



Figura 6. Logotipo de kea

Bind9

BIND9 es un software que nos permite montar nuestro propio DNS. Un DNS (Domain Name System) es un servicio que traduce las direcciones web que nosotros introducimos a direcciones IP. Es decir, si en el buscador escribo elpuig.xeill.net, el DNS lo traducirá a 136.243.80.28.

Queremos montar nuestro propio servidor DNS por temas de privacidad y velocidad. Por experiencia, hemos podido comprobar que los DNS que ofrecen los proveedores de internet o los públicos como el 1.1.1.1 de Cloudflare o el 8.8.8.8 de Google no son muy rápidos ni seguros, por lo que tener el nuestro nos soluciona muchos problemas.

Matrix

Matrix es un servicio de mensajería instantánea libre que cualquiera puede montar en su propia máquina. Cuenta con chats privados, chats de grupo, comunidades y la posibilidad de hacer tanto llamadas como videollamadas.

Hemos decidido optar por una instalación “sencilla” de Matrix principalmente por la dificultad de permitir las llamadas y videollamadas, ya que tendríamos que montar más servicios y otro servidor aparte para permitir las conexiones. Por esto, hemos optado por que los usuarios utilicen Nextcloud para videoconferencias y Matrix para mensajería instantánea.



Figura 7. Logotipo de matrix

Página web

Nuestra página web será la imagen pública del proyecto, donde los usuarios encontrarán manuales explicativos sobre cómo funcionan los servicios que proporcionamos, nuestros valores e información sobre nuestra visión y objetivos, además de un formulario con el que los usuarios podrán suscribirse y recibir actualizaciones de The Open Source Project.



Figura 8. Logotipo de la página web

2.4 Descripción de los componentes



Figura 9. Estructura de la red interna

Para la web estamos usando HTML, CSS y JavaScript.

La estética de la página está inspirada en flores de cerezo, con colores pastel, predominando colores como el rosa. Al principio estaba destinada a ser predominante en naranja y el logotipo sería Tux, el pingüino de Linux, pero al final decidimos que no lo haríamos con esa interfaz, ya que los tonos naranjas nos recuerdan mucho a Ubuntu, y añadir a Tux lo haría parecer una copia de Linux, lo cual no era nuestro objetivo, porque queríamos que la web, que se trata de la cara de nuestro proyecto, tuviera un estilo propio; algo que visualmente llamara la atención y fuera recordable por su estética.

Al conversar sobre la estética que podríamos utilizar, decidimos que el rosa como color principal sería ideal, ya que es un color muy versátil, llamativo y fácil de combinar, y a ambos nos gustan las flores de cerezo, las cuales combinan perfectamente con este color.

Para acceder al código de la página web, dirigirse [aquí](#).

2.4.1 index.html

El archivo es básicamente la página web completa del proyecto. Tiene varias partes que se van viendo según navegas. Al entrar, lo primero que nos encontramos es una pantalla de bienvenida con un botón que pone "Entrar", que es la portada antes de acceder al contenido principal.

Una vez dentro, hay una barra de navegación fija en la parte superior que siempre está visible mientras te mueves por la página. Tiene enlaces que te llevan directamente a cada sección, y en móvil se convierte en un menú desplegable.

El resto de la página está organizada en secciones: una presentación del proyecto con datos destacados, un diagrama visual de cómo están conectados los servicios, tarjetas con los valores del proyecto, dos bloques desplegables con los objetivos y la lista de las siete herramientas que emplea el proyecto.

A continuación está la parte de documentación, con las guías técnicas enlazadas para instalar y configurar cada servicio. Casi al final hay un formulario para suscribirse a las novedades por email y los enlaces a nuestras redes sociales, Mastodon y GitHub. Por último, un pie de página con los enlaces legales: política de privacidad, términos de uso y licencia del proyecto.

2.4.2 style.css

Este es el archivo encargado de definir el aspecto visual de toda la página web. Aquí se decide, por ejemplo, qué colores se usan en cada parte del sitio, qué fuente se utiliza en el texto, cómo son los botones o cuán redondeadas son las esquinas de las tarjetas.

También controla cosas que el usuario ve al interactuar con la página: la pantalla de bienvenida que aparece al entrar, el efecto de texto que se va escribiendo letra a letra, la barra de menú que siempre se queda visible en la parte superior mientras bajas por la página y el botón para volver arriba rápidamente.

El resto del archivo define cómo se ven todas las secciones de la página principal: la cabecera, los bloques de estadísticas, las tarjetas de valores y tecnologías, el apartado de documentación y el formulario de suscripción por correo. Al final también incluye instrucciones para que todo se adapte correctamente a pantallas de móvil.

2.4.3 doc.css

Este archivo complementa al anterior, pero solo se usa en las páginas de documentación técnica. Tiene dos partes principales.

La primera define el aspecto del diagrama visual que muestra cómo están conectadas las máquinas del proyecto: un cuadro con fondo rosa donde

aparecen los distintos servicios representados como bloques con su nombre y su dirección de red.

La segunda parte controla el diseño de todas las páginas de guías técnicas: la barra superior con el botón para volver atrás, la cabecera con el nombre del servicio y los bloques de contenido donde aparecen los pasos a seguir, los comandos que hay que escribir y las notas explicativas.

2.4.4 pages.css

En este archivo se define el diseño visual de las páginas secundarias del proyecto, como la política de privacidad. Es independiente de los otros archivos de estilos porque estas páginas tienen una estructura diferente. Controla cómo se ve la barra de navegación superior, las tarjetas de contenido que aparecen con una pequeña animación al cargar la página, los títulos de cada apartado, los textos, las listas, los cuadros de aviso destacados y el botón de contacto por email.

También incluye las adaptaciones necesarias para que el diseño se vea bien en móvil y tablet.

2.4.5 main.js

El main.js es el cerebro de la página principal. Se encarga de hacer que todo funcione cuando el usuario hace clic, mueve la página o rellena un formulario.

Entre otras cosas, gestiona el efecto de texto que se escribe solo al entrar en la web, la transición animada entre la pantalla de bienvenida y el contenido principal, el menú que se despliega en móvil, el botón de volver arriba y los bloques de texto que aparecen con animación suave conforme vas bajando por la página.

También se encarga del formulario de suscripción: comprueba que el correo electrónico introducido sea válido, que el usuario haya aceptado la política de privacidad y, si todo es correcto, envía el email de confirmación. Si algo falla, muestra un mensaje de error.

2.4.6 dockea.html

Esta es la página de documentación del servicio que se encarga de asignar automáticamente una dirección de red a cada dispositivo que se conecta, para que no haya que configurarlo a mano. Hablamos de kea-dhcp4-server.

La página explica cómo instalarlo, cómo configurarlo para que reparta direcciones dentro de un rango concreto, cómo reservar siempre la misma dirección para dos dispositivos específicos y qué significa cada uno de los

parámetros de configuración. Al final indica cómo activarlo y comprobar que está funcionando.

[2.4.7 docmatrix.html](#)

Esta es la página de documentación de Matrix, el servicio de mensajería instantánea del proyecto, el cual no depende de ninguna empresa externa y nadie ajeno puede leer los mensajes.

La página está organizada en cinco pasos. El primero indica todos los paquetes necesarios para que Matrix funcione y cómo añadir el repositorio oficial para descargarlo. El segundo explica cómo preparar la base de datos donde se almacenará toda la información de Matrix. El tercero detalla cómo configurar el archivo principal de Synapse, *homeserver.yaml*, indicándole cómo conectarse a esa base de datos y demás. El cuarto paso explica la instalación de Caddy, que se encarga de recibir las conexiones de los usuarios y redirigirlas correctamente a Matrix, gestionando además la seguridad de la conexión de forma automática. El quinto y último paso indica cómo reiniciar todos los servicios para aplicar los cambios y cómo comprobar que todo funciona accediendo desde un cliente de mensajería como Element o Cinny. Al final hay una nota que aclara que este servidor está pensado para uso interno, por lo que los usuarios solo pueden comunicarse entre ellos dentro de la misma organización.

[2.4.8 docnextcloud.html](#)

Esta es la página de documentación de Nextcloud, el servicio de almacenamiento en la nube privado del proyecto, similar a Google Drive pero sin depender de ninguna empresa externa.

En la página se explica, en orden, cómo descargar el programa, cómo instalar el servidor web y la base de datos que necesita para funcionar, cómo preparar esa base de datos, cómo colocar los archivos en el lugar correcto y cómo asignarle los permisos adecuados. Al final indica la dirección web a la que hay que acceder para terminar la configuración desde el navegador.

[2.4.9 docisardvdi.html](#)

Esta página explica qué es IsardVDI y cuál es su papel en el proyecto. IsardVDI es la plataforma donde viven todas las máquinas virtuales, es decir, los ordenadores simulados dentro de un servidor real que ejecutan cada uno de los servicios del proyecto.

A diferencia de las otras páginas de documentación, esta no explica una instalación, sino que describe por qué se eligió esta herramienta, qué ventajas

tiene frente a otras alternativas, cómo están organizadas las redes internas y qué máquina virtual se encarga de cada servicio. Al final incluye un enlace a la documentación oficial de la plataforma.

[2.4.10 docbind9.html](#)

Esta es la página de documentación de Bind9, el servicio que se encarga de traducir los nombres de dominio a direcciones de red dentro del proyecto. Gracias a él, en lugar de tener que recordar números como 192.168.101.40, se puede acceder a los servicios escribiendo un nombre más fácil como cloud.tosp.local.net.

La página está organizada en cuatro secciones. La primera explica cómo instalar el programa. La segunda muestra cómo decirle a Bind9 qué nombres de dominio debe conocer y dónde encontrar su configuración. La tercera detalla el archivo donde se definen todos los nombres y a qué dirección corresponde cada uno: la página web del proyecto, Nextcloud y Matrix. La cuarta y última sección indica cómo comprobar que la configuración no tiene errores y cómo reiniciar el servicio para que los cambios tengan efecto.

[2.4.11 términos.html](#)

Esta página recoge las normas de uso de los servicios del proyecto. Explica qué ofrece el proyecto, cómo se puede usar correctamente y qué está prohibido hacer.

Está dividida en nueve apartados: la aceptación de las normas, la descripción de los servicios disponibles, las normas de uso responsable con la lista de comportamientos no permitidos, el compromiso con la privacidad, el tipo de licencia con el que se comparte el código y los contenidos, los avisos sobre la disponibilidad del servicio, los límites de responsabilidad del proyecto, cómo se gestionarán los cambios futuros en estas normas y los datos de contacto. Al final aparece la fecha de la última actualización y los nombres de las personas que desarrollaron el proyecto.

[2.4.12 privacidad.html](#)

Esta página explica qué datos recoge el proyecto de los usuarios y cómo los usa. Está escrita de forma clara y directa para que cualquier persona pueda entenderla. Está organizada en ocho apartados: quién es el responsable del proyecto y cómo contactarle, qué datos se recogen, para qué se usan, cuál es la razón legal para tratarlos, cuánto tiempo se guardan, qué derechos tiene el usuario sobre sus datos, qué medidas de seguridad se aplican y cómo se avisará si esta política cambia en el futuro.

2.5 Definición de las funcionalidades

Esta sección describe de forma conceptual cómo funcionan las principales funcionalidades que ofrece la solución desarrollada, detallando qué permite hacer cada una, cómo se lleva a cabo y su estado de implementación actual.

2.5.1 Kea (DHCP)

El servicio Kea DHCP permite que cualquier dispositivo que se conecte a la red 192.168.101.0/24 reciba automáticamente una dirección IP, una puerta de enlace y un servidor DNS sin necesidad de configuración manual por parte del usuario.

El servidor distingue entre dispositivos con IP reservada (identificados por su dirección MAC) y dispositivos genéricos, a los que asigna una dirección del rango 192.168.101.150–192.168.101.254. Esta es una funcionalidad implementada completamente en nuestra red privada.

2.5.2 Resolución de nombres de dominio (DNS)

Bind9 permite que los usuarios accedan a páginas mediante nombres de dominio legibles en lugar de tener que recordar direcciones IP.

El servidor resuelve internamente todos los registros definidos en la zona tosp.local.net. Debido a dificultades con los certificados en local, los servicios web son solo accesibles públicamente mediante la función bastión de IsardVDI. Implementada completamente (uso interno).

2.5.3 Almacenamiento en la nube privado (Nextcloud)

Nextcloud ofrece a los usuarios un espacio de almacenamiento de archivos en la nube equivalente a Google Drive, pero alojado en los propios servidores del proyecto. Solo pueden acceder los usuarios que creen los administradores.

Además del almacenamiento, Nextcloud incluye funciones de chat y videoconferencia, que complementan a Matrix para cubrir la comunicación dentro de la organización. Esta función se encuentra implementada completamente.

2.5.4 Mensajería instantánea privada (Matrix)

Matrix proporciona un sistema de mensajería instantánea interno para la organización, con soporte para chats privados y grupos, sin depender de ningún servidor externo. Los usuarios se registran y acceden mediante cualquier cliente compatible con Matrix, como Element o Cinny, apuntando a la máquina donde lo tenemos alojado.

El servidor está configurado únicamente para comunicación interna, sin integración con otros dominios. Implementada completamente sin conexión externa por decisión propia.

2.5.5 Acceso a internet para los clientes de la red (Gateway)

El gateway permite que todos los dispositivos de la red tengan acceso a internet. Esto se consigue habilitando el reenvío de paquetes (`ip_forward`) y aplicando NAT mediante iptables, de forma que las peticiones de los clientes internos salen con la IP del gateway y las respuestas vuelven correctamente. Implementada completamente.

2.5.6 Página web del proyecto

La página web es la cara pública del proyecto. Permite a cualquier usuario conocer los servicios ofrecidos, acceder a la documentación técnica de cada componente (Kea, Bind9, Nextcloud, Matrix, IsardVDI), consultar los valores y objetivos del proyecto y suscribirse por correo electrónico para recibir novedades.

El formulario de suscripción valida el formato del email y la aceptación de la política de privacidad antes de enviar la confirmación. La web también incluye páginas de política de privacidad y términos de uso. Implementada completamente.

3. Configuración de servicios

Una vez decidido todo lo que tenemos que montar, hemos ido a Isard y hemos empezado a crear las máquinas virtuales, poniéndolas todas en la red `puigcastellar5`, menos el gateway, que lo hemos puesto tanto en `puigcastellar1` como en `puigcastellar5` para poder tener conexión a internet y la página web, que únicamente estará en `puigcastellar1`. Todo lo relacionado con el gateway se explicará más adelante, donde se explica cómo lo hemos hecho funcionar.

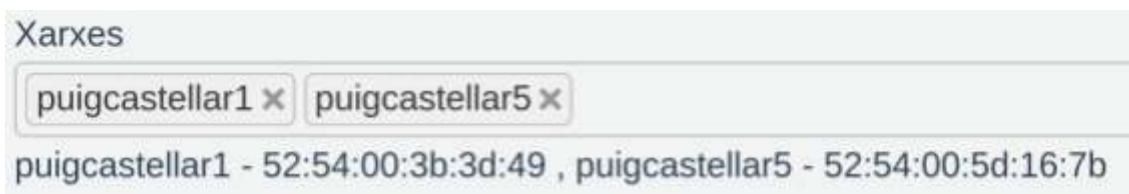


Figura 10. Interfaces de red seleccionadas en la máquina virtual de nuestro gateway alojado en isardVDI.

3.1. Kea (DHCP)

Kea es un servicio DHCP que asigna direcciones IP a los dispositivos que se conectan a la red. En nuestro caso, lo que queremos es mantener unas IPs fijas, que serán los clientes, y ofrecer un rango de IPs que se asignarán automáticamente a los dispositivos que se conecten y no formen parte de las IPs reservadas.

3.1.1. Instalación de kea-dhcp4-server

Antes de realizar la instalación, en nuestra máquina virtual de Isard, le hemos asignado la interfaz de red *puigcastellar5*, que es donde queremos montar nuestra infraestructura de red. No queremos tocar nada relacionado con *Puigcastellar1* con esta máquina, ya que esa red ya dispone de su DHCP, DNS, gateway y demás y no queremos causar ningún conflicto que pueda perjudicar a los usuarios que están utilizando esa red.

Antes de asignar IPs, para no lamentar ningún accidente, hemos hablado con Victor Carceler, quien nos ha explicado el funcionamiento de la red del instituto y la red de Isard. Este servicio de virtualización está ubicado en Alemania, pero mediante una VPN, podemos ubicarlo dentro de la red del instituto.

Victor nos enseñó las redes que podíamos utilizar. En un principio, queríamos quedarnos con 192.168.100.0/24, pero resulta que en esta red está ubicado Cirdan; por lo tanto, hemos cambiado a la 192.168.101.0/24. No habría ningún problema con montar nuestra infraestructura en la red 100, ya que nos encontramos en un entorno aislado de isardVDI, pero para no causar ninguna confusión, simplemente hemos optado por usar la red 101.

Empezamos por instalar Kea:

```
sudo apt update && sudo apt install kea-dhcp4-server -y
```

Seguido, editamos el fichero `/etc/kea/kea-dhcp4.conf`

Aplicamos la siguiente configuración, la cual se puede consultar en:

[Anexo 1 - instalación de kea-dhcp4-server: configuración del archivo kea-dhcp4.conf](#)

El contenido del archivo nos permite definir parámetros como el DNS que se les asignará a los hosts (192.168.101.20), el gateway que deben utilizar (192.168.101.10), el nombre del dominio, en nuestro caso, `network.tosp.local.net`, y en qué subred se aplicarán esos parámetros. Podemos ver todo esto definido aquí:

```
"subnet4": [
  {
    "subnet": "192.168.101.0/24",
    "match-client-id": false,
    "option-data": [
      {
        "name": "routers",
        "data": "192.168.101.10"
      },
      {
        "name": "domain-name-servers",
        "data": "192.168.101.20"
      },
      {
```

```
"name": "domain-name",  
"data": "network.tosp.local.net"  
}
```

También encontramos ajustes sobre la asignación de direcciones IP y su tiempo de validez:

```
"dhcp-socket-type": "raw"  
},  
"reservations-global": false,  
"reservations-out-of-pool": true,  
"valid-lifetime": 86400,  
"renew-timer": 43200,  
"rebind-timer": 75600,
```

El parámetro `dhcp-socket-type` lo dejaremos en `raw`, ya que es lo ideal para una red en la que se interactúa con dispositivos que no tienen una dirección IP asignada, que será el caso de los clientes que se vayan conectando a la red.

`reservations-global` se utiliza para definir si habrá dispositivos a los que se les asignará la misma IP sin importar la subnet a la que estén conectados.

`reservations-out-of-pool` permite que se puedan asignar IPs estáticas fuera del rango de la pool a los hosts.

`valid-lifetime` define el tiempo, en este caso segundos, por el cual la IP es válida hasta que tenga que volver a solicitar otra, es decir, renovarse.

`renew-timer`, en este caso la mitad del `valid-lifetime`, indica el tiempo que debe pasar después de haberle asignado la IP al host, para que vuelva a solicitar la IP al DHCP.

`rebind-timer` (aproximadamente un 80% del `valid-lifetime`) indica cuándo tiene que enviar un broadcast pidiendo renovar la IP a un DHCP disponible en la red en caso de que falle el DHCP que le asignó la anterior IP.

Una vez guardada la configuración y reiniciando Kea, si todo ha salido bien, ya se asignarán automáticamente las IPs a los hosts que se conecten a la red.

3.2. Bind9 (DNS)

BIND9 es un servicio DNS muy fácil de instalar que nos permite resolver dominios de manera más eficiente que el DNS que proporciona un ISP promedio al no estar obsoleto y saturado.

3.2.1. Instalación de bind9

Para instalarlo, hemos instalado el paquete bind9 con `apt install bind9`. Luego, nos hemos metido a la configuración, situada en el archivo `named.conf.local`, donde hemos añadido la configuración necesaria para establecer la zona.

```
usuario@tosp-dns:/etc/bind$ cat named.conf.local
//
// Do any local configuration here
//
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "tosp.local.net" IN {
    type master;
    file "network.tosp.local.net.hosts";
};
```

Lo que hemos hecho es añadir la zona `tosp.local.net`, que es el dominio que hemos definido para nuestro proyecto, que tendrá la configuración en el archivo `network.tosp.local.net.hosts` en el directorio `/var/cache/bind`.

En el archivo `/var/cache/bind/network.tosp.local.net.hosts` hemos añadido el siguiente contenido:

```
root@tosp-dns:~# cat /var/cache/bind/network.tosp.local.net.hosts
$TTL 300
@ IN SOA ns1.tosp.local.net. admin.tosp.local.net. (
    1 ; Serial
    10800 ; refresh
    3600 ; retry
    604800 ; expire
    60 ; minimum ttl
)

; nameservers
@ IN NS ns1.tosp.local.net.

; A Records - Servidores
ns1 IN A 192.168.101.20

; A Records - Servicios
@ IN A 192.168.101.1 ; tosp.local.net → página web
cloud IN A 192.168.101.40 ; cloud.tosp.local.net → nextcloud
matrix IN A 192.168.101.50 ; matrix.tosp.local.net → matrix
```

Nuestra idea inicial era hacerlo todo local y no montar en un dominio público ningún servicio, pero debido a dificultades que hemos tenido con los certificados y el acceso desde la red del centro sin la necesidad de utilizar un cliente de Isard, hemos decidido usar la función bastión, que nos crea un hash para poder acceder a lo que haya montado en esa página desde cualquier parte del mundo. Si bien hace que quede inutilizada la configuración que hemos montado en el DNS, sirve como demostración en caso de que se monten más servicios; en caso de que sean locales, ser conscientes de cómo podemos convertir esa IP en un dominio como hemos hecho anteriormente.

3.3. Nextcloud (almacenamiento en la nube)

Nextcloud es una aplicación web que te permite montar tu propia nube localmente, como si fuera Google Drive, pero con total privacidad.

Nuestra idea inicial era montarlo solamente en local, pero, como hemos mencionado anteriormente, debido a complicaciones con el acceso a estos, hemos decidido ponerle el bastión de Isard para acceder desde donde sea. A pesar de esto, solamente podrán acceder usuarios que creen los administradores, por lo que sigue siendo más seguro que cualquier otro servicio de almacenamiento en la nube.

Si bien hemos probado también OwnCloud, otra alternativa open source popular, hemos optado por Nextcloud por su interfaz gráfica más intuitiva y mayor variedad de funciones, ya que incluye chats y llamadas sin la necesidad de aplicar una configuración más avanzada como en Matrix.

3.3.1. Instalación de Nextcloud

Para instalarlo, hemos seguido las instrucciones que hemos sacado de [este repositorio de github](#).

Una vez creada la máquina y los repositorios actualizados, procedemos a descargar el zip con la última versión de Nextcloud.

```
wget  
https://download.nextcloud.com/server/releases/nextcloud-33.0.2.zip
```

Instalamos Apache2 (servidor HTTP) y MySQL (base de datos).

```
sudo apt install -y apache2 mysql-server
```

Instalamos paquetes de PHP necesarios para que Nextcloud funcione.

```
sudo apt install -y php libapache2-mod-php php-fpm php-common  
php-mbstring php-xmlrpc php-soap php-gd php-xml php-intl php-mysql  
php-cli php-ldap php-zip php-curl
```

En el caso de que hubiéramos instalado Owncloud, tendríamos que añadir un repositorio para tener una versión anterior de PHP, pero como Nextcloud es compatible con la actual, no será necesario.

Nos metemos en MySQL para crear la base de datos y el usuario:

```
mysql
CREATE DATABASE bdd;
CREATE USER 'usuario'@'localhost' IDENTIFIED WITH
mysql_native_password BY 'password';
GRANT ALL PRIVILEGES ON bdd.* TO 'usuario'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

Ahora, para instalar el Nextcloud en sí, movemos el zip de descargas /var/www/html con el comando mv y nos situamos en el mismo directorio con cd.

Descomprimos el zip con el comando unzip y lo borramos con rm; seguido, movemos todo el contenido del directorio que se ha creado al directorio html y borramos el directorio que se nos ha creado anteriormente.

Con esto ya podremos acceder a Nextcloud. Nos pedirá indicar dónde está la base de datos y las credenciales para acceder. Introducimos toda la información y ya tendremos todo listo.

Como hemos mencionado antes, hemos acabado haciendo las páginas públicas con el bastión de Isard. Esto lo hemos hecho con Caddy, que nos proporciona un certificado válido de Let's Encrypt para poder hacer nuestras páginas públicas. Ha sido fácil hacerlo; simplemente hemos instalado Caddy y añadido esto en su respectivo fichero de configuración:

```
root@tosp-nextcloud:/home/usuario# cat /etc/caddy/Caddyfile
dbf3bce9-de3a-4e28-822c.5764bb4a5b99.bastion.elmeuescriptori.cat {
    # Para otros clientes de NextCloud
    redir /.well-known/carddav /remote.php/dav 301
    redir /.well-known/calDav /remote.php/dav 301

    # Reverse proxy al apache
    reverse_proxy localhost:8080

    # Security headers
    header {
        Strict-Transport-Security "max-age=31536000;"
        X-Content-Type-Options "nosniff"
        X-Frame-Options "SAMEORIGIN"
    }
}
```

Los parámetros `redir` sirven para que los usuarios puedan utilizar clientes de terceros para Nextcloud, por ejemplo, el cliente para escritorio de Owncloud, que se puede utilizar para ambas aplicaciones.

El reverse proxy lo que hace es que Caddy le envíe las solicitudes al Apache, ya que Caddy es solamente un enlace hacia la nube que en sí está montada en el Apache.

Las cabeceras de seguridad fuerzan la conexión mediante HTTPS a la página y mejoran la seguridad de esta.

3.4. Matrix (Mensajería instantánea)

Matrix es un servicio de mensajería instantánea libre y descentralizado. Si bien tienen sus propios servidores en matrix.org, puedes montar tu propio Matrix, como hemos hecho nosotros, donde puedes manejar todos tus datos y dependes de ti mismo para tu privacidad.

Hemos elegido Matrix, ya que es un servicio bastante conocido en la actualidad; si bien muchas personas no le dan preferencia al ser un servicio que no tanta gente utiliza y que mucha gente ya está acostumbrada a redes como Discord o WhatsApp, estas se quedan con tus datos y los venden para obtener beneficios, mientras que en Matrix todo se queda donde está hospedado y el peligro es mucho más bajo.

3.4.1. Instalación de Matrix

Matrix es una de las cosas que más tiempo nos ha consumido en este proyecto, pero para nosotros ha valido la pena, ya que creemos que sería una muy buena opción para un centro educativo o empresa que quiere tener una manera rápida y segura de establecer contacto.

Hemos seguido [estas instrucciones](#) para poder montar nuestro propio servidor Matrix.

Hemos empezado por desplegar la máquina de Isard donde alojaremos Matrix, como siempre, un Ubuntu 24.04 server. Le pondremos la interfaz de red `puigcastellar5`, y configuraremos el netplan con una IP estática, en este caso, `192.168.101.50`.

Una vez dentro, nos convertimos en sudo, actualizamos los paquetes e instalamos los necesarios para Matrix:

```
apt update && apt install -y lsb-release wget apt-transport-https  
build-essential python3-dev libffi-dev python3-pip python3-setuptools  
sqlite3 libssl-dev virtualenv libjpeg-dev libxslt1-dev libicu-dev  
libpq5 libpq-dev postgresql postgresql-contrib debian-keyring  
debian-archive-keyring apt-transport-https
```

Para empezar ahora sí con la instalación de Matrix, introducimos los siguientes comandos en la terminal:

```
wget -O /usr/share/keyrings/matrix-org-archive-keyring.gpg  
https://packages.matrix.org/debian/matrix-org-archive-keyring.gpg  
echo "deb  
[signed-by=/usr/share/keyrings/matrix-org-archive-keyring.gpg]  
https://packages.matrix.org/debian/ $(lsb_release -cs) main" | tee  
/etc/apt/sources.list.d/matrix-org.list
```

Instalamos Matrix con `apt install -y matrix-synapse-py3`. Es importante, antes de instalarlo, actualizar la lista de paquetes, ya que si no lo hacemos, nos saldrá que no se ha podido encontrar el paquete.

Mientras se instala, nos preguntará por el dominio de nuestro servidor. Hemos puesto `tosp-matrix`, ya que es como queremos que aparezcan los nombres de usuario de las cuentas en nuestro servidor. Es importante que, si se cambia esto en el futuro, todas las cuentas que tenían el nombre anterior quedarán completamente inutilizadas.

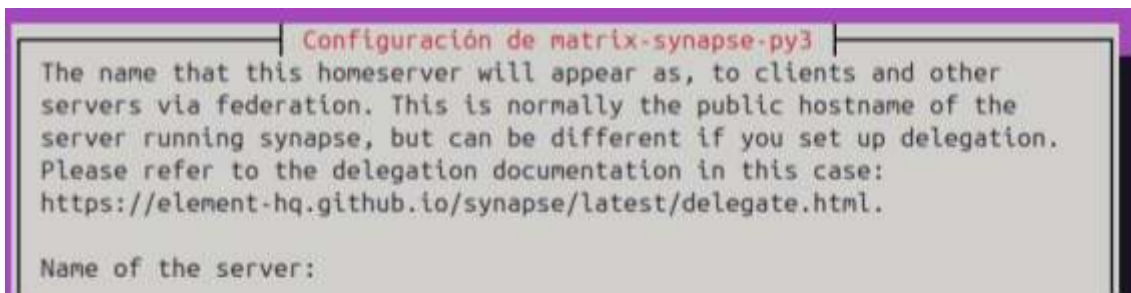


Figura 10. Ventana que aparece al instalar matrix-synapse para definir el nombre del dominio

Entramos a la base de datos con `sudo -u postgres bash` y creamos el usuario de esta. En nuestro caso hemos puesto el mismo que el del manual. Hay que tener en cuenta que al ejecutar este comando nos pedirá también crear una contraseña para el usuario; es importante recordarla.

```
createuser --pwprompt synapse_user
```

Creamos la base de datos, la cual llamaremos *Synapse*.

```
createdb --encoding=UTF8 --locale=C --template=template0  
--owner=synapse_user synapse
```

Salimos de PostgreSQL y editamos el fichero `/etc/postgresql/12/main/postgresql.conf` para configurar la base de datos.

Buscaremos dentro de este el parámetro `'listen_addresses ='` al cual, después del `=`, le pondremos `'localhost'`. Quedará tal que así:

```
# - Connection Settings -  
listen_addresses = 'localhost'          # what IP address(es) to listen on;
```

Guardamos y salimos del fichero.

Ahora entraremos a `/etc/postgresql/12/main/pg_hba.conf` para activar la autenticación mediante contraseña para que el usuario que hemos creado pueda conectarse a la base de datos.

Al final del archivo, añadimos esta línea.

```
local  synapse          synapse_user          md5
```

El archivo quedará tal que así:

```
# Allow replication connections from localhost, by a user with the  
# replication privilege.  
local  replication     all                    peer  
host   replication     all                    127.0.0.1/32         md5  
host   replication     all                    ::1/128              md5  
local  synapse         synapse_user          md5
```

Salimos del archivo para ahora añadir la configuración de synapse en nuestro archivo `homeserver.yaml`, el archivo principal de configuración de Matrix, aunque antes de tocar nada, borraremos las primeras líneas que aparecen:

```
database:  
  name: sqlite  
  args:  
    database: /var/lib/matrix-synapse/homeserver.db
```

Añadimos estas líneas en el fichero:

```
database:  
  name: psycopg2  
  args:  
    user: synapse_user  
    password: <contraseña>  
    database: synapse  
    host: localhost  
    cp_min: 5  
    cp_max: 10
```

El resultado final del `homeserver.yaml` se puede consultar en:

[Anexo 2 - Instalación de Matrix: contenido de homeserver.yaml](#)

Ahora vamos a configurar el proxy inverso.

Un proxy inverso es un servidor intermedio que se sitúa delante de un servidor web y recibe las peticiones de los clientes, en vez de recibirlas directamente el servidor. En este caso, usaremos Caddy.

Todavía no hemos instalado Caddy, ya que el manual nos recomienda añadir los repositorios oficiales de Caddy antes de instalarlo. Lo haremos con los siguientes comandos:

```
curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/gpg.key' | sudo gpg  
--dearmor -o /usr/share/keyrings/caddy-stable-archive-keyring.gpg
```

```
curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/debian.deb.txt' |  
sudo tee /etc/apt/sources.list.d/caddy-stable.list
```

Hacemos un apt update e instalamos Caddy.

```
apt update && apt install caddy
```

Vamos a editar el archivo de configuración de Caddy, ubicado en /etc/caddy/Caddyfile

Borramos todo el contenido y añadimos la siguiente configuración, que se puede consultar en el siguiente anexo:

[Anexo 3 - Instalación de Matrix: Contenido de Caddyfile](#)

Hecho, ya podemos recargar los servicios para comprobar que todo está en orden y Matrix funciona.

```
systemctl restart caddy.service  
systemctl restart postgresql.service  
systemctl restart matrix-synapse
```

Si todo ha salido bien, al entrar al link del bastión, nos saldrá esto:

[**matrix**]

It works! Synapse is running

Your Synapse server is listening on this port and is ready for messages.

To use this server you'll need [a Matrix client](#).

Welcome to the Matrix universe :)

matrix.org

Figura 11. Mensaje que aparece al entrar en la URL del bastion de isard al acabar de configurar matrix

Ya tenemos listo matrix, ahora falta comprobar que realmente funciona, para ello, iremos a un cliente de matrix a hacernos una cuenta. En nuestro caso usaremos Element, pero hay muchos más que se pueden usar, uno bastante conocido es Cinny.

Iremos a app.element.io, seleccionamos registrarnos y en *host account on*, introducimos la URL del bastión tal que así:

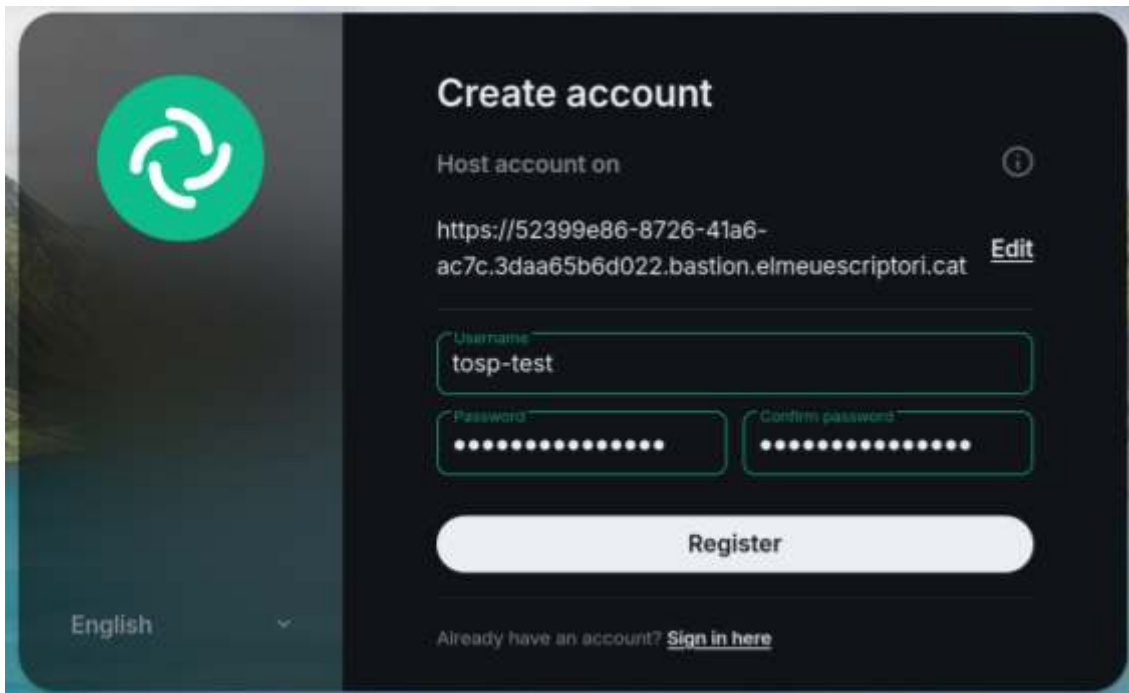


Figura 12. Sign-in de element donde se indica que la cuenta se aloja en nuestro servidor matrix

Ya tenemos nuestra cuenta creada. Ahora podemos empezar chats con otras personas con cuentas de Matrix creadas en el mismo dominio que esta, ya que, por desgracia, no permite comunicarse con cuentas en otros dominios (ejemplo: matrix.org)

Si bien esto puede verse como un punto negativo, lo hemos hecho a propósito, ya que este servidor de Matrix está hecho para poder hablar entre las personas de una organización o empresa, no con personas externas, por lo que no supone un problema para nosotros.

3.4. Gateway

Un gateway es esencial en una red, ya que permite que los hosts se puedan conectar a internet y enviar peticiones a sitios web. Configurarlos es muy sencillo, ya que solamente se necesita una máquina virtual y unos pocos comandos para hacerlo funcionar.

3.4.1. Configuración del gateway

Primero crearemos una máquina virtual con Ubuntu Server en isardVDI, a la cual le añadiremos dos interfaces de red: *puigcastellar1* y *puigcastellar5*. La encendemos, iniciamos sesión y nos metemos en netplan para configurar las tarjetas de red.

```
network:
  ethernets:
    enp1s0:
      dhcp4: true
    enp2s0:
      dhcp4: false
      addresses: [192.168.101.10/24]
      nameservers:
        addresses: [192.168.101.20]
  version: 2
```

Lo que hacemos aquí es dar una IP por DHCP a la interfaz *enp1s0*, ya que es la interfaz que va a *puigcastellar1*, y no nos interesa ponerle una IP fija para no causar problemas en esta red, que ya dispone de su infraestructura propia. En la tarjeta de red de *puigcastellar5*, pondremos una IP fija y el DNS que tenemos montado en nuestra red. No es necesario añadir una ruta a internet, ya que utiliza la ruta que recibe por DHCP en la primera tarjeta de red.

Ahora vamos a configurar los parámetros necesarios para que esta máquina sirva como gateway. Es muy simple, solamente tenemos que activar el ip forward en la máquina para que los paquetes puedan salir de esta hacia internet.

```
sysctl -w net.ipv4.ip_forward=1
```

Además, activaremos NAT. Es esencial para que quien ha recibido el paquete sepa a quién responder, cambiando la IP del remitente por la de *enp1s0*. Es decir, si yo envío un paquete de esta red a la red de clase, el router de clase sabe que tiene que responder a nuestro gateway.

```
iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE
```

Y con esto, tenemos configurado el gateway. Ahora, los hosts de la red tienen acceso a internet.

4. Conclusiones

4.1 Conclusiones generales del proyecto

Con este proyecto nos hemos demostrado que es perfectamente viable construir una infraestructura de red completa y funcional utilizando exclusivamente software libre, sin depender de ningún servicio comercial ni de terceros.

A lo largo del desarrollo hemos podido montar desde cero servicios de red esenciales como un DHCP y un DNS, así como aplicaciones web un poco más complejas como Nextcloud y Matrix, aprendiendo a integrarlos en un ecosistema coherente y autónomo.

Académicamente, el proyecto nos ha permitido poner en práctica de forma conjunta conocimientos adquiridos en distintos módulos: configuración de servicios de red, administración de servidores Linux, instalación de aplicaciones web y desarrollo de páginas web. Haber afrontado problemas reales, como las dificultades con los certificados TLS o la configuración de Matrix, y haberlos resuelto de forma autónoma ha sido, para nosotros, la parte más importante del proyecto.

A nivel personal, el proyecto refuerza nuestra mentalidad de que la privacidad digital no es un lujo, sino algo que cualquier persona, empresa o centro educativo puede conseguir con las herramientas adecuadas y los conocimientos necesarios.

4.2 Consecución de los objetivos

Montar una infraestructura de red con los servicios necesarios para ser útil: alcanzado.

Se han desplegado y configurado correctamente un servidor DHCP (Kea), un servidor DNS (Bind9) y un gateway con acceso a internet.

Preservar la privacidad de los usuarios: alcanzado.

Todos los servicios están autoalojados y los datos nunca salen de sus máquinas. Nextcloud y Matrix garantizan que tanto los archivos como los mensajes de los usuarios permanecen bajo control propio.

No depender de empresas que alojan sus servicios en sus propios servidores: alcanzado.

Se han sustituido completamente Google Drive por Nextcloud, WhatsApp/Discord por Matrix y los DNS comerciales por Bind9, sin recurrir a ningún servicio de pago ni de terceros.

Promover una alternativa honesta, real y transparente: alcanzada.

La página web del proyecto documenta de forma pública y detallada cómo instalar y configurar cada servicio, con el objetivo de que cualquiera pueda replicar la infraestructura por su cuenta.

Ofrecer servicios de almacenamiento en la nube y mensajería instantánea: alcanzado.

Nextcloud y Matrix están operativos y accesibles incluso desde fuera de la red local gracias a la función bastión de IsardVDI.

4.3 Valoración de la metodología y planificación

La metodología Scrum nos ha resultado adecuada para la naturaleza del proyecto. Dividir el trabajo en bloques independientes (primero la infraestructura de red, luego las aplicaciones web y finalmente la página) nos permitió avanzar de forma ordenada y detectar problemas en cada componente sin que afectaran al resto.

El diagrama de Gantt fue suficiente como herramienta de seguimiento dado el tamaño del equipo, aunque en algún momento nos vimos obligados a replanificar algunas tareas, especialmente las relacionadas con Matrix, que consumió más tiempo del previsto debido a la complejidad de su configuración y a la necesidad de resolver errores muy específicos con poca documentación disponible. A pesar de esto, al final, la planificación fue realista y el reparto de tareas entre los dos miembros del equipo funcionó bien, aunque en momentos de mayor dificultad o falta de información fue necesario hacer más hincapié entre ambos para poder resolver el problema.

4.4 Visión de futuro

El proyecto tiene margen de crecimiento en varias direcciones: En primer lugar, sería deseable habilitar la federación en Matrix para permitir la comunicación con usuarios de otros servidores, lo que requeriría configurar un servidor TURN/STUN para las llamadas y gestionar correctamente los certificados de dominio público.

En segundo lugar, Nextcloud admite una gran cantidad de aplicaciones adicionales que no hemos podido explorar, como calendarios compartidos, gestión de tareas o edición colaborativa de documentos, que convertirían la plataforma en una alternativa aún más completa a Google Workspace.

También sería interesante añadir un sistema de monitorización de los servicios (como Grafana, el cual hemos visto este año en el módulo de servicios de red) para detectar caídas o problemas de rendimiento en tiempo real, y explorar la posibilidad de automatizar el despliegue de la infraestructura mediante herramientas como Ansible para facilitar la replicación del proyecto en otros entornos. Estos no han podido llevarse a cabo por falta de tiempo, pero son

opciones muy interesantes, las cuales hemos explorado y tenemos inspiración para utilizar en el futuro.

Finalmente, a largo plazo, el proyecto podría evolucionar hacia una oferta más estructurada para centros educativos y pequeñas empresas que quieran adoptar infraestructuras libres sin tener que configurarlo todo desde cero.

4.5 Declaración del uso de inteligencia artificial

Respecto a la parte práctica.

Este proyecto se ha desarrollado, en parte, con el apoyo de herramientas de inteligencia artificial para comprender el funcionamiento de algunos componentes de los servicios y entender el funcionamiento de los lenguajes de programación que se han utilizado a lo largo del desarrollo tanto de los servicios como de la página web.

No obstante, se ha optado en todo momento por el uso de manuales y los conocimientos adquiridos en clase. Bajo ninguna circunstancia se ha tenido iniciativa de recurrir a la inteligencia artificial como principal fuente de información. Se ha hecho uso de la información pública que disponemos en foros y manuales hechos por usuarios a la hora de realizar cualquier consulta.

A pesar de ello, sea por falta de información o errores humanos, no hemos podido hacer funcionar algunos componentes para los cuales no encontrábamos ninguna solución en sus foros ni documentación ya que se trataban de errores muy específicos, por lo que es aquí donde hemos recurrido a la inteligencia artificial para obtener fuentes de información o posibles soluciones que podemos tratar de comprender y poner a prueba para seguir adelante.

Respecto a la memoria.

La memoria de nuestro proyecto ha sido redactada por los miembros del equipo, absteniéndose así de textos *copia y pega* generados por IA.

A pesar de esto, se ha utilizado inteligencia artificial para ofrecer una mejor expresión a la hora de explicar conceptos sobre el funcionamiento de la web, los servicios de red y las aplicaciones web, debido a que algunos componentes nos resultaban difíciles de comprender y expresar sus funciones. No obstante, las ideas, decisiones, tecnologías y reflexiones finales son propias y no ha habido influencia ninguna respecto a la inteligencia artificial, tanto en la parte práctica como en la memoria.

5. Glosario

Apache2: Servidor HTTP de código abierto utilizado para alojar aplicaciones web como Nextcloud y la página del proyecto.

Bind9: Software de código abierto que permite configurar un servidor DNS propio para traducir nombres de dominio a direcciones IP.

DNS (Domain Name System): Sistema que traduce las direcciones IP (por ejemplo, 136.243.80.28) a nombres de dominio (por ejemplo, elpuig.xeill.net).

Caddy: Servidor HTTP y proxy inverso que gestiona automáticamente los certificados SSL mediante Let's Encrypt.

DHCP (Dynamic Host Configuration Protocol): Protocolo que asigna automáticamente una dirección IP y otros parámetros de red (DNS, NTP...) a los dispositivos que se conectan a la red que tiene asignada el dhcp.

kea-dhcp4-server: Servidor DHCP moderno de código abierto utilizado en este proyecto para la asignación de direcciones IP.

Federación (Matrix): Capacidad de un servidor Matrix para comunicarse con usuarios registrados en otros servidoresMatrix.

Gateway: Máquina que actúa como punto de salida de una red local hacia internet.

IP estática: Dirección IP asignada de forma fija a un dispositivo. No cambia con el tiempo ni con cada nueva conexión a la red.

IsardVDI: Plataforma de virtualización de escritorios de código abierto utilizada en este proyecto para alojar todas las máquinas virtuales.

Let's Encrypt: Autoridad de certificación gratuita que emite certificados para habilitar conexiones HTTPS.

Matrix Synapse: homeserver descentralizado y de código abierto para mensajería instantánea. Synapse es la implementación de referencia del servidor.

Nextcloud: Aplicación web autoalojable de código abierto para el almacenamiento de archivos en la nube.

Proxy inverso: Servidor intermediario que recibe las peticiones de los clientes y las redirige al servidor correspondiente.

PostgreSQL: Sistema de gestión de bases de datos de código abierto, utilizado como base de datos de Matrix.

Software libre / Código abierto: Software cuyo código fuente es público y puede ser usado, modificado y redistribuido libremente por cualquier persona.

TLS (Transport Layer Security): Protocolo de cifrado que garantiza la seguridad de las comunicaciones en internet, habilitando las conexiones HTTPS.

Ubuntu Server: Variante oficial de Ubuntu. Distribución de linux orientada a servidores, utilizada como sistema operativo base en todas las máquinas virtuales del proyecto.

6. Bibliografía

- [1] Nextcloud. Nextcloud Documentation. <https://docs.nextcloud.com> (consultado en febrero de 2025)
- [2] Internet Systems Consortium. Kea DHCP Administrator Reference Manual. <https://kea.readthedocs.io> (consultado en enero de 2025)
- [3] ISC. BIND 9 Administrator Reference Manual. <https://bind9.readthedocs.io> (consultado en enero de 2025)
- [4] Element Synapse - Matrix homeserver implementation. <https://element-hq.github.io/synapse/latest/> (consultado en marzo de 2025)
- [5] Caddyserver. Caddy Documentation. <https://caddyserver.com/docs> (consultado en marzo de 2025)
- [6] IsardVDI. IsardVDI Documentation. <https://isard.gitlab.io/isardvdi-documentation> (consultado en enero de 2025)
- [7] Canonical Ltd. Ubuntu Server Guide. <https://ubuntu.com/server/docs/> (consultado en enero de 2025)
- [8] Element. Element Web - Matrix Client. <https://element.io> (consultado en marzo de 2025)
- [9] Creative Commons. Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya. <https://creativecommons.org/licenses/by-nc-nd/3.0/es> (consultado en enero de 2025)
- [10] Somnath Das, *Setup Matrix Synapse Home-server*. Medium, 18 de junio de 2023. <https://medium.com/@dassomnath/setup-matrix-synapse-home-server-ba54e20f8290> (consultado en marzo de 2026)

7. Anexos

7.1. Instalación de kea-dhcp4-server: configuración del archivo kea-dhcp4.conf

```
{
  "Dhcp4": {
    "interfaces-config": {
      "interfaces": [
        "enp2s0"
      ],
      "dhcp-socket-type": "raw"
    },
    "reservations-global": false,
    "reservations-out-of-pool": true,
    "valid-lifetime": 4000,
    "renew-timer": 1000,
    "rebind-timer": 2000,
    "subnet4": [
      {
        "subnet": "192.168.101.0/24",
        "match-client-id": false,
        "option-data": [
          {
            "name": "routers",
            "data": "192.168.101.10"
          },
          {
            "name": "domain-name-servers",
            "data": "192.168.101.20"
          },
          {
            "name": "domain-name",
            "data": "network.tosp.local.net"
          }
        ]
      },
      {
        "pool": "192.168.101.150-192.168.101.254"
      }
    ],
    "reservations": [
      {
        "hostname": "host1.network.tosp.local.net",
        "hw-address": "52:54:00:10:82:ce",
        "ip-address": "192.168.101.100"
      },
      {
        "hostname": "host2.network.tosp.local.net",
        "hw-address": "52:54:00:7e:98:64",
        "ip-address": "192.168.101.101"
      }
    ]
  }
}
```

```
    ],  
    "loggers": [  
      {  
        "name": "*",  
        "severity": "DEBUG"  
      }  
    ]  
  }  
}
```

7.2. Instalación de Matrix: contenido de homeserver.yaml

```
server_name: "matrix-tosp"  
public_baseurl:  
"https://52399e86-8726-41a6-ac7c.3daa65b6d022.bastion.elmeuescriptori.  
cat"  
serve_server_wellknown: true  
pid_file: "/var/run/matrix-synapse.pid"  
listeners:  
  - bind_addresses:  
    - "::1"  
    - "127.0.0.1"  
  port: 8008  
  resources:  
    - compress: false  
      names:  
        - client  
        - federation  
  tls: false  
  type: http  
  x_forwarded: true  
database:  
  name: psycopg2  
  args:  
    user: tosp-admin  
    password: PASSWORD  
    database: synapse  
    host: localhost  
    cp_min: 5  
    cp_max: 10  
log_config: "/etc/matrix-synapse/log.yaml"  
media_store_path: /var/lib/matrix-synapse/media  
signing_key_path: "/etc/matrix-synapse/homeserver.signing.key"  
trusted_key_servers:  
  - server_name: "tosp-matrix"  
  
registration_shared_secret: "PASSWORD"  
  
enable_registration: true  
enable_registration_without_verification: true  
rc_registration:  
  per_second: 9999  
  burst_count: 9999
```

7.3. Instalación de matrix: contenido de Caddyfile

```
52399e86-8726-41a6-ac7c.3daa65b6d022.bastion.elmeuescriptori.cat {  
  # Matrix client API  
  reverse_proxy /_matrix/* localhost:8008 {  
    header_up Host {host}  
    header_up X-Forwarded-For {remote_host}  
    header_up X-Forwarded-Proto {scheme}  
  }  
  
  # Synapse client endpoints (for SSO, registration, etc.)  
  reverse_proxy /_synapse/client/* localhost:8008 {  
    header_up Host {host}  
    header_up X-Forwarded-For {remote_host}  
    header_up X-Forwarded-Proto {scheme}  
  }  
  
  # Fallback for everything else (including .well-known)  
  reverse_proxy localhost:8008 {  
    header_up Host {host}  
    header_up X-Forwarded-For {remote_host}  
    header_up X-Forwarded-Proto {scheme}  
  }  
}
```