



TREBALL DE FI DE GRAU

TÍTULO DEL TFG: Desarrollo de una Oficina de Ingeniería Remota: Análisis y Implementación de una Solución en el Contexto del Teletrabajo

TITULACIÓN: Grado en Ingeniería Telemática

AUTOR: Rodrigo Sampedro Casis

DIRECTOR: Antonio Marzoa Domínguez

FECHA: 23 de octubre de 2023

Título: Desarrollo de una Oficina de Ingeniería Remota: Análisis y Implementación de una Solución en el Contexto del Teletrabajo

Autor: Rodrigo Sampedro Casis

Director: Antonio Marzoa Domínguez

Fecha: 23 de octubre de 2023

Resumen

Actualmente el mundo tecnológico sufre una expansión sin precedentes debido a la cuarta revolución industrial^[1] y el acceso o actualización de las pequeñas y medianas empresas a las tecnologías de la información^[2]. Esto ha supuesto la llegada de una amplia oferta de empleos de escasa inversión material, alta capacitación curricular y la posibilidad de deslocalización de los empleados, favoreciendo el crecimiento de lo que se ha venido a conocer como “la ingeniería en remoto”^{[3] [4]}. Los empleados que han entrado en este nuevo ambiente de trabajo, tienen herramientas similares debido a su condición de “remoto” mayoritario, independientemente de si pertenecen a una gran empresa o una startup, sean freelances o subcontratados por consultoras externas.

Paralelamente, tras la pandemia de COV-19^[5], se ha evidenciado la necesidad y facilidad del trabajo en remoto así como la conciliación del home office. Sin embargo, ha creado la falacia del ingeniero con un ordenador, capaz de trabajar desde casa con compañeros que no se conocen físicamente, así como la vuelta irremediable a la oficina por pérdida de productividad en muchas empresas^[6].

Este TFG analiza, evalúa e implementa una solución a la creación de una oficina de desarrollo de ingeniería en remoto, explotada y utilizada por el propio autor del mismo, detallando los problemas encontrados y las soluciones halladas e implementadas durante la construcción de esta.

En los anexos se resume el estado de arte de los principales puntos de interés, especialmente en el ámbito tecnológico, así como se presenta una posible guía para la puesta en marcha de una startup tecnológica a partir del trabajo realizado, y desde un punto de vista técnico, puesto que los requisitos son tanto aplicables para un trabajador en remoto, como para un trabajador por cuenta ajena, o bien aplicable al desarrollo inicial de una empresa con pocos trabajadores.

El objetivo principal es obtener un set mínimo de recursos que permitan de una manera genérica, sin un coste excesivo, permitir un correcto desarrollo de la actividad, sorteando problemas y facilitando la creación de herramientas de trabajo colaborativo.

Asimismo se ha utilizado esta plantilla como base de proyectos personales y como implementación activa en varias empresas pyme de familiares y amigos del propio autor.

Title : Development of a Remote Engineering Office: Analysis and Implementation of a Solution in the Context of Home Office

Author: Rodrigo Sampedro Casis

Advisor: Antonio Marzoa Domínguez

Date: October 23, 2023

Overview

At the moment, the technological world is undergoing an unprecedented expansion due to the fourth industrial revolution[1] and the access or update of small and medium-sized enterprises to information technologies[2]. This has led to the emergence of a wide range of jobs with low material investment, the requirement of high curriculum qualification, and the possibility of employee delocalization, promoting the growth of what has come to be known as "remote engineering"[3] [4]. Employees who have entered this new work environment use similar tools due to their predominantly "remote" condition, regardless of whether they belong to a large company or a startup, or being freelancers, or being subcontracted by external consulting firms.

In parallel, the COVID-19 pandemic[5], the need and ease of remote work, as well as the balance of home office, have become evident. However, it has created the fallacy of the engineer with a computer, capable of working from home with colleagues they have never met physically, as well as the inevitable return to the office due to a loss of productivity in many companies[6].

This Bachelor's Degree analyzes, evaluates, and implements a solution for creating a remote engineering development office, used by the author on his own, detailing and explaining the problems and solutions undertaken during its construction.

In the appendices, the state of the art of the main points of interest regarding the project is summarized, especially in the technological field, and a possible guide for launching a technology startup from a technical perspective is presented since the requirements are applicable to both remote and in-house workers, or even to the initial development of a company with few employees.

The main goal is to obtain a minimal set of resources that, in a fairly generic manner and without excessive cost, enable the proper development of activities, overcoming challenges, and facilitating the creation of collaborative work tools.

Furthermore, this template has been used as the basis for personal projects and actively implemented in several family and friend-owned SMEs.

a Chloe promotora de la idea

ÍNDICE GENERAL

Introducción	1
0.1. Objetivos	1
0.2. Requisitos	1
CAPÍTULO 1. Oficina física	5
1.1. Requisitos	5
1.1.1. Elementos físicos	5
1.1.2. Setup informático	5
1.1.3. Conectividad	6
1.2. Tabla de Requisitos	6
1.3. Diseño Práctico	8
1.3.1. Circunstancias e histórico	8
1.3.2. Creación de un espacio de trabajo	8
1.3.3. Dimensiones y distribución	9
1.3.4. Electricidad y Red	10
1.4. Construcción Oficina	10
1.4.1. Acondicionamiento terraza	10
1.4.2. Casetas y armario	11
1.4.3. Presupuestos planificación	12
1.4.4. Resultados de construcción	14
CAPÍTULO 2. Servicios y herramientas	17
2.1. Software, hardware y gestión.	17
2.2. Ansible	18
2.3. Contenedores y Docker Service	19
2.4. Docker Compose, Kubernetes, Docker Swarm	21
2.5. Arquitectura de nuestros Servicios	22
2.6. Aprovisionamiento y Securización	23
2.7. Servicios	24

2.8. Múltiples Docker compose y limitaciones	26
2.9. Docker Backups	26
2.10. Automatizaciones y scripts	27
2.11. Legalidad	28
2.11.1. Clasificación de datos e inventario	28
2.11.2. Consentimiento y mecanismo de acceso	29
CAPÍTULO 3. Redes y Casuísticas	31
3.1. VPN e intranet	31
3.1.1. Objetivos de VPN	31
3.1.2. Tipo de enlaces VPN	31
3.2. Casuísticas de interés	33
3.2.1. Interconexión de redes privadas por VPN	33
3.2.2. Múltiples capas de VPN	34
3.2.3. DNS filtro y espejos	34
3.2.4. Exposición externa vía VPS	35
3.3. Docker y automatización de redes	36
3.3.1. Automatizaciones	36
3.3.2. Reverse Proxy y HTTPS	36
3.3.3. Dns automatizado	37
3.3.4. Interconexión de Red docker, VPN y gateway	37
3.4. Caso desarrollado	39
3.5. Generalización de Casos	40
CAPÍTULO 4. Desarrollo de software	43
4.1. Dinámica de trabajo (equipo)	43
4.1.1. Agile	43
4.1.2. Scrum	44
4.1.3. Generalización y profesionalidad	45
4.2. Git, CI/CD y contenedor	45
4.2.1. Git y repositorios de código	45
4.2.2. CI/CD	46
4.2.3. Opinión de un usuario con experiencia	47
4.3. CI basado en docker	47

4.4. CI/CD dentro de nuestra nube	48
4.5. Setup Software Local	48
4.5.1. Automatización Local	48
4.5.2. Herramientas de desarrollo basadas en docker	48
4.5.3. Dot files y configuraciones portables	49
CAPÍTULO 5. Conclusiones	51
5.1. Conclusiones de la aplicación en Elenkar	51
5.2. Conclusiones Personales	52
5.3. Trabajo futuro	52
Bibliografía	53
APÉNDICE A. Teletrabajo	65
A.1. Contexto Semántico	65
A.2. Objetivo del teletrabajo	66
A.3. Histórico	67
A.3.1. Pre pandemia	67
A.3.2. Pandemia	67
A.3.3. Post Pandemia	68
A.3.4. Opinión Personal	68
A.4. Ofertas laborales	69
A.5. Teletrabajo overemployed	70
A.6. Caso de estudio	70
APÉNDICE B. Requisitos y oficina física	71
B.1. Definición y tipos de requisitos	71
B.2. Requisitos físicos	72
B.2.1. Área de Trabajo	72
B.2.2. Otros elementos	72
B.3. Setup informático	73
B.3.1. ¿Cuál es tu prioridad?	74
B.3.2. Pantalla, comodidad y opinión	74

B.3.3. Hardware y recursos	76
B.3.4. Periféricos	76
B.3.5. Gestión del setup	77
B.4. Abastecimientos auxiliares	78
B.4.1. Suministro Eléctrico	79
B.4.2. Climatización	80
B.4.3. Acceso a Internet	80
B.4.4. Circunstancias e histórico	84
B.4.5. Nueva habitación	88
B.4.6. Caseta Oficina	89
B.4.7. Mejoras de Terraza	95
B.5. Evaluaciones, mejoras y correcciones	96
B.5.1. Humedades y condensación	96
B.5.2. Bomba de calor/frío	97
B.5.3. Paneles solares y SAI	98
B.5.4. Domótica y alarma	99
APÉNDICE C. Software y Nube	101
C.1. Software	101
C.1.1. Proveedor, mantenimiento y gestión	101
C.1.2. Licencias y tipo de software	101
C.2. Elección de Servidor	103
C.2.1. Servidor Físico vs Servidor virtual vs Cloud Externo	103
C.2.2. Securización	105
C.2.3. Setup y abastecimiento	105
C.3. Servicios MVP	107
C.3.1. Servicio de Comunicación	108
C.3.2. Servicio de interconectividad	109
C.3.3. Servicio de almacenamiento	110
C.3.4. Servicio de documentación	111
C.3.5. Servicio de Repositorios y CI	111
C.3.6. Servicio de web externa	112
C.3.7. Servicio de MetaDatos y contraseñas	112
C.3.8. Servicio de Autenticación Centralizada	113
C.4. Tabla de servicios públicos - externos	114
C.5. Servicios Dockerizados Shelf Host	117
C.6. Contenedores y relaciones extendidas	119

C.6.1.	Docker multi-capa	120
C.6.2.	SystemD service con docker-compose	123
C.6.3.	Backups y restauración de servicios	125
C.6.4.	Estructura de ficheros	126
APÉNDICE D. Redes		129
D.1. Conceptos básicos		129
D.1.1.	VPN y encapsulado	130
D.1.2.	Protocolo y tecnología	130
D.1.3.	Enrutado	132
D.1.4.	Nat Masquerade	134
D.1.5.	DNS	134
D.1.6.	Proxy	135
D.2. Seguridad		136
D.2.1.	VLAN	137
D.2.2.	Filtrado Mac	137
D.3. Encendido Remoto		137
D.3.1.	Wake on LAN	138
D.3.2.	Clock - Scheduler wake up	138
D.3.3.	Smart switch, grid	138
D.4. Elementos Usados		138
D.4.1.	Conexión	138
D.4.2.	Switch	139
D.4.3.	Router	141
D.4.4.	Servidor autocrático	142
D.4.5.	Camaras IP	145
D.4.6.	Smart Devices	145
APÉNDICE E. Pruebas de concepto y MVP		147
E.1. Pruebas de Concepto		147
E.1.1.	Comunicación	147
E.1.2.	Ejemplo web	148
E.1.3.	Ejemplo wiki	149
E.1.4.	Cloud Storage	151
E.1.5.	Reverse Proxy y https	154
E.1.6.	Ticketing	155
E.1.7.	SSO Ldap y Keycloak	156
E.1.8.	VPN	157

E.1.9. CI / CD	160
E.2. Wireguard LAN-VPN-LAN	162
E.2.1. Nat masquerade hace red docker	162
E.2.2. Script en cliente	162
E.3. Mínimo Viable Producto	164
E.4. Aprovisionamiento básico Ansible	169
APÉNDICE F. Proyectos de aplicación	171
F.1. Elenkar	171
F.2. Casos menores, emprendimiento	172
F.3. Mi uso personal	172

ÍNDICE DE FIGURAS

1.1 Diseño caseta, distribución mobiliario	9
1.2 Montaje eléctrico.	10
1.3 Estructuras de aluminio.	11
1.4 Caseta iluminada	11
1.5 Caseta mobiliario.	12
1.6 Panorámica caseta migrada.	12
1.7 Setup en mayo, vista satélite pre pandemia, pandemia, actualidad.	14
2.1 Ansible diagrama funcionamiento[143][144].	18
2.2 Ansible ejemplo de ejecución desde proyecto VScode	19
2.3 Maquina virtual vs contenedores[145].	20
2.4 Docker compose, swam y kubernetes logo images[146].	21
2.5 Diagrama de interacción.	22
3.1 Casuística de interés de VPN - VPS.	32
3.2 Diagrama interconexión de redes por VPN[147].	33
3.3 VPN multi-salto, multi-capa.	34
3.4 Diagrama DNS proxy[157].	34
3.5 Conexión a recursos detrás de CG-NAT / Firewall o NAT.	35
3.6 Reverse proxy diagram.	35
3.7 Diagrama de interconexión de redes docker y servicios.	38
3.8 VPNs, relación entre ellas VPS y clientes.	39
4.1 Diagrama metodología AGILE.	43
4.2 Diagrama funcionamiento SCRUM-AGILE[149].	44
4.3 Diagrama flujo de trabajo en git[150].	46
A.1 Coworking[151] vs home office[152]	65
B.1 Cubículo biblioteca UPC (campus nord)	72
B.2 Silla y mesas, productos Amazon.	73
B.3 Porta monitor, raton ergonómico, productos Amazon.	73
B.4 Portátil, mini pc y torre, google images.	74
B.5 Monitores, formas y geometrías [153].	75
B.6 Evolución tecnológica monitores ultimos 30 años.	75
B.7 Conectividad de monitores, amazon products.	78
B.8 Enchufes e instalación eléctrica.	79
B.9 Diagrama de conexión.	80
B.10Setup pre pandemia.	85
B.11Setup pandemia.	85
B.12Terraza en pandemia.	86
B.13Setup post pandemia.	87
B.14Setup 2023.	87

B.15Terraza planos, pendiente y zonas inúndales.	89
B.16Terraza cerramientos planificados con pendientes zonas inúndales marcadas.	90
B.17Casetas división espacial.	90
B.18Limpieza, pintura y trabajos previos.	91
B.19Montaje de sistema eléctrico y cableado Ethernet.	92
B.20Panel de aluminio lacado con aislante térmico.	92
B.21Montaje paneles aluminio y soportes.	93
B.22Vista de perfiles y techo con aislamiento.	93
B.23Mobiliario y setup en caseta.	94
B.24Instalación eléctrica en caseta y cerramientos.	94
B.25Regulación térmica y lumínica.	95
B.26Mejoras externas de terraza que afectan a la caseta.	96
B.27Mejoras externas de terraza II que afectan a la caseta.	96
B.28Impermeabilización con pintura y silicona.	97
B.29Paneles solares, tejado y frontal chimenea.	98
 C.1 Tipos de licencias mas comunes[25].	102
C.2 Tipos de servicios en servidores[154].	103
C.3 Comunicaciones mas utilizadas.	108
C.4 Clientes centralizadores de comunicaciones.	109
C.5 Intranet ventajas.[49]	110
C.6 Opensource alternativas a almacenamiento en la nube.	111
C.7 Gitea junto a drone opción mas común y actual.	112
C.8 Gestor de metadatos y contraseñas.	113
C.9 Single Sign on, diagrama[155].	113
C.10Prueba de concepto[105] SSO Keycloak con openldap.	114
C.11Docker in docker vs Docker out of docker.	120
C.12Docker in docker diagrama de ejecución de comandos[156].	121
C.13Docker out of docker diagrama de ejecución de comandos[156].	122
C.14Sysbox [35] diagrama de funcionamiento.	123
C.15Backup diagrama de secuencia.	125
C.16Restore diagrama de secuencia.	126
C.17Estructura de un Servicio A.	127
C.18Estructura completa y ejemplos de include, merge o makeFile.	127
 D.1 Diagrama capa OSI [120].	129
D.2 Encapsulado túnel paquete de VPN en red real.	130
D.3 Diagrama interconexión de redes por vpn.	132
D.4 Diagrama resolución de queries dns proxy[157].	134
D.5 Ngix-rever proxy + Let's encrypt dockerized[158].	136
D.6 Cables Cat con sus diferentes trenzados y apantallamientos[159].	139
D.7 Imagen TP-LINK TL-SG108 Switch 8 Puertos (PCCcomponent imágenes).	140
D.8 Imagen TP-Link LS1005G Switch 5 Puertos (PCCcomponent imágenes).	140
D.9 Imagen Mercusys MS105G Switch 5 Puertos (PCCcomponent imágenes).	140
D.10Router portable GL-MT300N-V2 (amazon products).	142
D.11Placa Rasberry pi (amazon products imagen).	143
D.12Placa Orange Pi PC 3 (Aliexpress products imagen).	143
D.13Placa Banana Pi M2+(Aliexpress products imagen).	144

D.14EQ12 Pro y Chuwi LarkBox X (Aliexpress products imagen). 144

ÍNDICE DE TABLAS

1.1 Tabla de requisitos físicos, oficina.	7
1.2 Presupuesto Caseta Física	13
2.1 Docker services más de 10 personas.	24
2.2 Servicios docker nube MVP.	25
B.1 Comparación Tecnologías de acceso	82
B.2 Setup pre pandemia tabla comparativa	85
B.3 Setup Pandemia tabla comparativa	86
C.1 Tabla comparativa opciones servidor.	104
C.3 Tabla servicios públicos para externalización.	115
C.4 Tabla servicios dockerizados.	117
C.5 Tabla servicios dockerizados Extras.	119
D.1 Tabla enrutado elemento solo conectado a la red naranja.	132
D.2 Tabla enrutado desde A.	133
D.3 Tabla enrutado desde un elemento solo conectado a la red verde.	133
D.4 Tabla enrutado desde B.	133

BLOQUES DE CÓDIGO

C.1	SystemD /etc/systemd/system/docker-compose@.service	123
C.2	SystemD /etc/systemd/system/custom-service	124
C.3	Crone line /etc/crontab, actualizacion imagenes	124
C.4	SystemD /etc/systemd/system/docker-compose-reload.timer	124
C.5	SystemD /etc/systemd/system/docker-compose.reload.service	124
C.6	Fichero de configuracion daemon.json	125
E.1	docker-compose.yml Rocket Chat prueba de concepto.	147
E.2	docker-compose.yml Wordpress prueba de concepto.	148
E.3	docker-compose.yml PineDocs prueba de concepto.	149
E.4	docker-compose.yml MediaWiki prueba de concepto.	149
E.5	docker-compose.yml BookStack prueba de concepto.	150
E.6	docker-compose.yml SeaFile prueba de concepto.	151
E.7	docker-compose.yml Samba prueba de concepto.	152
E.8	docker-compose.yml NextCloud prueba de concepto.	152
E.9	docker-compose.yml Ngix proxy y Let's Encrypt prueba de concepto.	154
E.10	docker-compose.yml Planka prueba de concepto.	155
E.11	docker-compose.yml SSO LDAP-Keycloak prueba de concepto.	156
E.12	docker-compose.yml OpenVPN con UI.	157
E.13	docker-compose.yml Wireguard prueba de concepto.	158
E.14	docker-compose.yml Wireguard Easy prueba de concepto.	159
E.15	docker-compose.yml Gitea-Drone CI prueba de concepto.	160
E.16	Ejemplo de configuracion servidor Nat masquerade	162
E.17	Ejemplo de configuracion servidor para homenets	163
E.18	docker-compose.yml MVP prueba de concepto.	164

INTRODUCCIÓN

Este TFG esta centrado en el concepto de teletrabajo y mas específicamente en los requisitos, herramientas y conceptos que se aplican en un trabajo remoto profesional. El caso de estudio se centra en la solución realizada a un problema logístico y familiar, para habilitar un espacio adecuado para el teletrabajo y las herramientas necesarias tanto desde el punto de vista del trabajador como desde la infraestructura en la nube de una empresa.

0.1. Objetivos

Los objetivos de este TFG son:

- Planificar y solventar la necesidad de un espacio de teletrabajo (para este autor) y facilitar el home-office o flexwork, para la pareja del autor, debido a la incorporación de un nuevo miembro a la familia (Chloe).
- Investigar, recopilar y resumir un estado del arte tanto infraestructura (física) como herramientas e infraestructura de software (virtuales).
- Definir los requisitos y un ejemplo de implementación de una oficina física, para la realización del teletrabajo de este autor. Implementarlo y evaluar los resultados o puntos de mejora y comparativas con oficinas / setups previos.
- Definir e implementar, una nube con los servicios mínimos para realizar teletrabajo, de especial interés para autónomos, pequeñas y medianas empresas o startups tecnológicas.
- Simplificar y automatizar el proceso, con el objetivo de poder usar el resultado práctico de este TFG, como elemento genérico o producto fácilmente desplegable sin necesidad de conocimientos profundos.

0.2. Requisitos

La definición de trabajo en remoto, difiere según el porcentaje de jornada laboral realizada; la ubicación desde dónde se realiza; la topología y forma de la empresa y el estatus jurídico del trabajador. En el anexo A, se desarrollan los diferentes tipos o escalas de teletrabajo, sin embargo la casi totalidad de desarrollo técnico de **este documento se centra en la modalidad de “home office” de un 60-100%**, es decir, trabajar desde casa e ir 1-2 días por semana o en ocasiones específicas. Por consiguiente una parte importante de este documento se centra en los requisitos y caso práctico de ‘una oficina física de desarrollo tecnológico’ para teletrabajar.

¿Qué entendemos como una oficina de desarrollo tecnológico? Todos aquellos trabajos que se involucran en el desarrollo (mantenimiento o creación) de productos/servicios tecnológicos. Esto está intrínsecamente relacionado con software y una de las siguientes necesidades más demandadas por pymes (pequeña y mediana empresa):

- Desarrollo de software: sin incluir los nuevos productos, existe una necesidad de digitalizar todo proceso en papel, manual o burocrático así como actualizar viejos software para su uso extendido en 'la nube'; o el digital twin[13] que permite representar elementos físicos en el mundo digital y realizar simulaciones.
- Infraestructura en la nube: permitir el acceso des-localizado, el escalado y la flexibilidad de gestión al tener recursos en una nube, tanto como servicio como el soporte asociado.
- Infraestructura tecnológica (redes & hardware), acceso, actualización y puesta en marcha tanto de servidores, seguridad, sensores, oficinas o elementos para el teletrabajo.
- Análisis de datos: existe un nuevo mercado basado en la gran cantidad de datos y el minado y procesado de los mismos con fines optimizadores o como subproducto derivado.
- Electrónica, sensores, impresión 3D y prototipado: puede resumirse en llevar las nuevas tecnologías low cost al mercado tradicional, permitiendo un grado tecnológico en pequeñas empresas o explotaciones, anteriormente solo utilizado en grandes empresas.
- Ciberseguridad y temas legislativos: los nuevos riesgos y deberes implica que una actividad se digitalice.

¿Qué significa esto para los requisitos de nuestra oficina de desarrollo? Básicamente que un componente muy significativo obligatorio debe estar centrado en la infraestructura física del trabajador y la gestión o creación de una nube digital por parte de la empresa.

Independientemente de que tipo de trabajador y donde sea el lugar de la actividad ¿Qué necesita un técnico para desarrollar un trabajo en remoto?

- Un espacio físico, internet, luz, hardware, temperatura, control de horarios ... que nombraremos requisitos físicos.
- Software y plataforma de comunicación directa, ya sean salas, teléfono, herramientas informáticas (correo, videoconferencia, red social corporativa, página web), organizar/almacenar/monitorizar el trabajo realizado.
- Redes, Vpn, capas de seguridad de aislamientos, seguridad.
- Wiki, documentación, guías o código de buenas prácticas, que no solo faciliten la producción de los proyectos sino que eviten problemas estructurales y promuevan un mantenimiento ágil así como definen la dinámica de trabajo desde el punto de vista humano.

Durante los diferentes apartados de este documento, se irán desglosando los diferentes temáticas de interés iniciando desde un punto de vista genérico dando paso a un caso específico para finalmente exponer la implementación de "mi oficina" realizada por este autor, y la nube digital asociada. El documento principal introduce y desarrolla las utilidades y conceptos utilizadas así como los resultados obtenidos pero la gran parte del

contexto, razonamientos y detalles o comparativas se predisponen en los anexos adjuntos o en algunos casos en documentos externos no adscritos a esta memoria.

Este TFG esta segmentado en 3 partes, capítulo **1**, anexos **A** y **B**, focalizados en la perspectiva del trabajador. Capítulo **2** y **3** y anexos **C**, **D**, **E** focalizados en la infraestructura necesaria desde el punto de vista técnico-empresarial. Y capítulo **4** centrado en las dinámicas de trabajo en grupo, buenas prácticas o metodología en desarrollo de software.

Finalmente en el anexo **F** las conclusiones obtenidas en casos reales al implementar la base proporcionada por la 'oficina física y virtual', se evalúan y mencionan varios sub-proyecto personales así como la aplicación en varios negocios reales.

CAPÍTULO 1. OFICINA FÍSICA

Este tema define los requisitos teóricos de la construcción de una oficina en remoto, contrastada con dos setups, uno mi habitación de estudio reconvertida en oficina durante la pandemia de 2020-21 véase [B.4.4.2.](#) y el diseño implementación y uso de “mi oficina-virtual” durante 2022-23. Los requisitos se catalogan en 7 categorías (véase anexo [B.1.](#)), **mínimo, adecuado y óptimo** respecto a la calidad del requisito; y **básico, profesional, profesional pro y business** en relación a el coste.

1.1. Requisitos

En este apartado se define la base adecuada para una oficina en casa o también útil como base elemental de oficina para una startup tecnológica.

1.1.1. Elementos físicos

Entendemos como mínimo indispensable un pc o laptop para realizar la tarea, sin embargo esto es “la falacia estudiantil”. Realizar un trabajo profesional de manera continuada (8 horas), sin interrupciones y concentrado no debe llevarse a cabo en espacios públicos o familiares. Las malas condiciones posturales así como climatización o la inestabilidad de servicios claves como internet degradan la calidad final del trabajo[\[14\]](#) y contribuyen psicológicamente en una actitud negativa del trabajador[\[15\]](#).

Está claro que un elemento implícito es el espacio de trabajo, una habitación, un cubículo, un coworking, su principal función es permitir instalar en el el setup informático quien es el verdadero centro de la oficina. Pero otros aspectos como tener una iluminación y ventilación adecuada, así como facilitar la organización de apuntes, herramientas o periféricos necesarios durante la jornada laboral son tan importantes como el propio setup.

Finalmente se ha definido espacialmente, el doble del tamaño de un cubículo de las bibliotecas de UPC como el espacio adecuado y una habitación dedicada como óptima.

1.1.2. Setup informático

En el anexo [B.3.](#) mencionan las principales corrientes en la selección y montaje de un setup. En mi opinión y como conclusión se recomienda encarecidamente un setup basado en laptop, aunque puede utilizarse un pc o mini pc. Es requisito indispensable un monitor extendido de 24’-27’, aunque se recomienda el uso de dos pantallas ([B.3.2.](#)). El uso de periféricos tales como ratón/teclado, auriculares con micrófono, todos ellos conectados a través de un elemento centralizador de interconexión de pantallas, pc/laptop.

Respecto al hardware existe una evolución histórica[\[16\]](#), cuellos de botella y estrategias a la hora de planificar la compra y mantenimiento de los diferentes componentes del hardware, como resumen en el punto [B.3.3.](#), detalla las estrategias a utilizar en la selección de hardware sin la comprensión del mismo.

En base a las estrategias de selección de hardware, en el contexto actual (2023) la prioridad es la utilización de las nuevas plataformas RAM DDR5 enlazadas con las últimas series de procesadores AMD Ryzen o Intel; ram adecuada de 16 GB, 32 GB óptimo, discos duros SSD o NVMe de 512 GB - 1 TB.

Focalizando en modalidades 8 cores (adecuado), es decir, procesadores Ryzen 5/7, o Intel i5/i7 que como configuración final puede presupuestarse entre 900-1200€ + mantenimiento 100-150€ en 2026.

Pero siguiendo una estrategia 'b', pueden conseguirse ofertas significativas para obtener un hardware bastante equivalente basado en ddr4 en el rango 500-700€ y 100€ de mantenimiento en 2025, y planificar un reemplazo por un hardware nuevo de 600-800€ en 2027.

Por último existe una extensión de la estrategia 'b' basada en mini pc, la cual aunque degrada la movilidad y el mantenimiento, permite obtener un hardware sin GPU equivalente por 250-400€ y un mantenimiento 100-150€ en 2025.

Respecto a las tarjetas gráficas deben ser adecuadas a las necesidades del trabajador. Un programador puede usar gráfica cpu-embedded, pero actividades gráficas o renderizado requieren de GPU dedicadas. En caso de optar por GPU integradas, los procesadores AMD Ryzen tienen una clara ventaja sobre Intel tanto en performance, como en calidad-precio.

Para un mayor detalle así como entender la selección y mantenimiento del setup recomendamos encarecidamente la lectura del anexo B o guía externa para hardware [16].

1.1.3. Conectividad

Toda infraestructura asumida en el espacio de trabajo, son los llamados abastecimientos auxiliares (véase anexo B.4.), ya que sin ellos no es posible realizar el trabajo o en algunos casos degrada tanto la eficacia como la comodidad del trabajador.

Internet o conectividad es el requisito indispensable, ya que una oficina puede existir pero no ser en remoto. Otra cosa es definir las propiedades de la conexión, como solución general el requisito de una conexión con un mínimo de **15 / 5 Mbps (down/up)**, con **delays** no mayores a **75 ms** y tasas baja de **jaming 30 ms**, es decir, un servicio de ADSL céntrico, cable coaxial, conectividad 3G/4G/5G urbana o fibra estándar cumplen o superan las características requeridas.

De especial interés el análisis comparativo de la tabla B.4.3.4. que cataloga aquellas tecnologías no aptas, problemáticas o poco apropiadas por calidad precio, como son Wimax, internet por satélite o ADSL en zonas rurales.

1.2. Tabla de Requisitos

En la tabla 1.1 resume los diferentes requisitos necesarios de una oficina física para teletrabajar. Aquellos que comienzan con "+" hace referencia a extender en anterior requisito con más elementos. Las categorías o elementos marcados indicados con "*" son de uso obligatorio. Véase un acoplamiento generalizado entre prestaciones y costes, que no re-

presenta la realidad de mercado, sino una tendencia intencionada para buscar las mejores prestaciones en base a un precio límite.

Tabla 1.1: Tabla de requisitos físicos, oficina.

	Mínimo - Básico	Adecuado - Profesional	Óptimo - Profe.Pro	Óptimo - Business
Espacio de trabajo*	cubículo UPC	escritorio personal	habitación / co-working	despacho dedicado
Iluminación*	luz blanca oficina	+luz natural	+regular luz natural	+regular luz artificial
Ventilación*	manual	automático	aclimatado	aclimatado con renovación
Hardware Base*	pc-torre	mini-pc laptop	laptop especializado	hardware sobre dimensionado o exclusivo
CPU* cores	4 cores 2-3 años	8 cores 1-2 años	8-16 cores 1-2 años	12/16/32 cores ¡1 año
RAM*	ddr4 8-12 GB	ddr4 16 GB ddr5 8-16 GB	ddr4 32 GB ddr5 16-32 GB	ddr5 32 GB
Disco Duro*	SSD 256 GB 100 - 200 MB/s	SSD 512 GB - 1000 GB 400-600 MB/s	SSD mayor 1 TB 600-1000 MB/s	NVMe 1-2 TB 1000 - 7000 MB/s
Pantallas*	x1 21'-24', HD	x2 21'-24', HD-FHD	x2 27' FHD-2K	x2-x3 27'-34' 2K-4K
Periféricos	webcam micro* auriculares* ratón*	HD cam* HD micro Auriculares HF ratón-teclado inalámbrico*	+cam bloqueable +auriculares con cancelación de ruido +pad táctil o bolígrafo táctil	+cam HD cancelación de ruido y micrófono estéreo +software eliminación de sonido externos
Elemento centralizador	-	Cable VGA/HDMI Hub usb	dock-station* extender usb-vga-hdmi	hub-station usb-c
Elemento de seguridad	-	lector de tarjeta con pin	lector de huellas	llave usb con cierre de lector huellas
Elemento opcionales	alfombrilla ratón pomodoro	+reposamuñecas +reposapiés +monitor elevado	+monitores ajustables +ratón ergonómico	+VR
Conectividad*	ADSL / 3G urbano	ADSL2 / VDSL / 4G / cable coaxial Fibra 100 MB/s	Fibra >300 MB/s 5G	StarLink
Extras	almacenaje estanterías y cajones	impresora / escáner / teléfono ip	kit electronica soldador impresora 3D pizarra	projector alexa / google home / sensoring hub
Seguridad	cajonera con llave	+ip cam	+habitación con llave	+sensores de puerta o presencia

La gran mayoría de los requisitos de un setup eficiente para un uso profesional deben de estar enmarcados en opciones “**adecuadas-profesionales**” u “**óptimo - profesional-pro**”. En concreto el caso de estudio de este trabajo se centra en un enfoque “óptimo” siempre que el precio sea “profesional”, o degradado a “adecuado” por la mejora de rela-

ción calidad-coste.

1.3. Diseño Práctico

En este apartado se plantea un problema real, “crear un espacio y acondicionarlo” para permitir la instalación de un setup real asumiendo la solución teórica de los requisitos anteriores.

1.3.1. Circunstancias e histórico

En el anexo [B.4.4.](#), se especifican las condiciones generales, pre pandémica, durante la pandemia y post pandemia. Podemos resumirlas en “siempre he contado con una habitación dedicada como despacho compartida con mi pareja”, dicha habitación tenía un uso estudiantil así como para hobbies. Varios elementos, tales como pantalla, conexión de internet o espacio dedicado, hacían inviable su uso real como lugar de teletrabajo.

La remodelación (véase [B.4.4.2.](#)) y mejora durante y post pandemia la convirtió en un lugar apto para teletrabajar (monopolizado por mi), sin embargo dicha remodelación estaba dirigida a una función más familiar (habitación para niños). En 2022 se descarta completamente el uso de la habitación puesto que para finales de ese año se espera la incorporación de un nuevo miembro a la familia. Las nuevas circunstancias (véase [B.14](#)) obligan a la búsqueda de un espacio adecuado y reconstrucción completa de la habitación de teletrabajo, diseñada bajo los parámetros del punto 1.2 de este trabajo. Así como la planificación de una nube privada aplicable tanto a mi oficina para proyectos personales, como al trabajo de mi pareja para flexibilizar o facilitar el trabajo remoto con la nueva incorporación familiar.

1.3.2. Creación de un espacio de trabajo

En el anexo [B.4.5.](#) se especifican posibles soluciones espaciales, así como las circunstancias detalladas del problema. Tras evaluar las diferentes posibilidades finalmente la solución es crear un espacio de trabajo utilizando una caseta de aluminio a medida hermética.

Esta solución elimina una zona mal drenada y hundida de la terraza y que no es útil debido a una respiradero-chimenea comunitario así mismo se combina con armario de aluminio en otra zona infrautilizada para realojar aquellos elementos que ocupaban el espacio de la caseta.

Por lo tanto, debido a ser la única solución aceptable como material de construcción se usa aluminio y paneles sándwich con cámara térmica, lacados en blanco del mismo tipo que los utilizados por la comunidad de vecinos para la separación entre terrazas, apropiadamente anclados a muro, paredes y apropiadamente herméticos con silicona, especialmente suelos.

1.3.3. Dimensiones y distribución

El acceso a la caseta, así como la iluminación natural, se realizan mediante una puerta corrediza y una ventana acristalada también corredera, típicas de cerramientos de aluminio.

Con el objetivo de maximizar la capacidad, y el movimiento el techo será prácticamente plano, con una inclinación de 5º para drenar el agua. Por normativa comunitaria, dichas aguas deben caer a mi terraza, no se puede exponer nada diferente al panel de aluminio en los tramos de alféizar de la fachada.

Por lo tanto las dos únicas paredes útiles son las que disponen de la ventana (Sur-Oeste) y puerta (Oeste-Norte), y por consiguiente las otras dos paredes son las indicadas para almacenamiento y setup fig.1.1.

Aunque no es lo mas deseado una 'iluminación natural lateral' en las pantallas, se dedujo que la predisposición más útil es un Setup en 'L', utilizando la pared Oeste-Norte como base del setup y la ventana como base de la 'L' generando una mesa extendida para tareas manuales como soldadura, herramientas o disponer de un espacio extra y permitiendo la colocación de almacenaje en la pared norte (la más extensa).

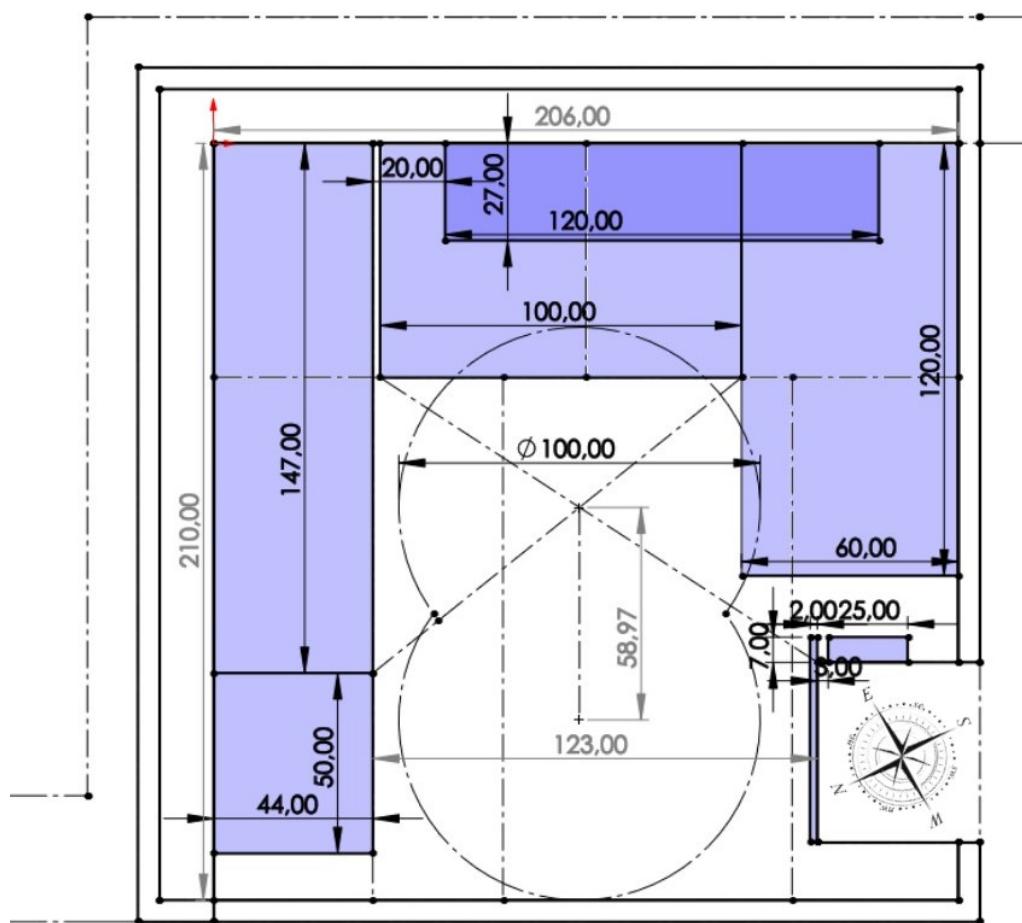


Figura 1.1: Diseño caseta, distribución mobiliario

Dicha pared tiene de suelo a techo, mobiliario con cajoneras y estanterías, quedando el segundo nivel apoyado directamente sobre el medio alféizar. En el centro de la caseta

se ha generado un espacio de 1-1.3 metros ideal para una silla giratoria con ruedas que accede fácilmente a toda la oficina y no entorpece el movimiento.

Por último tanto el espacio de entrada como los recovecos entre la chimenea-respiradero y el aluminio, son ampliamente funcionales debido a la puerta corredera y el uso de las pizarras este espacio es ampliamente usado como lugar de reflexión y pensamiento.

1.3.4. Electricidad y Red

La conexión eléctrica así como la conectividad se basa en una instalación nueva, extendiéndose desde la existente durante el proceso de puesta a punto previo a la instalación de armario y caseta. Con el objetivo de facilitar y separar las instalaciones, se dispondrá de un cableado de $2.5mm^2$ y PIA asociado únicamente para la caseta y otro para enchufes e iluminación de terraza.

Inicialmente se contaba con un PLC de 10 Mbps que permite conexiones a través de la red eléctrica de iluminación de la terraza, sin embargo se ha instalado un cableado directo de ethernet desde la vivienda con el fin de maximizar dicha conexión a 1 Gbps.

1.4. Construcción Oficina

Este punto resume la construcción de la caseta presupuesto y tiempos aunque su detalle y pormenorizado se encuentra en el anexo B.

1.4.1. Acondicionamiento terraza

Previamente a la instalación de la caseta así como del armario se realizaron un conjunto de tareas de acondicionamiento y mantenimiento previos que están descritos en anexo B.4.6.1. Los principales cambios fueron:



Figura 1.2: Montaje eléctrico.

1. Limpieza de paredes (barrido y agua a presión), alféizares con productos ácidos y suelos (con ácido clorhídrico diluido y detergente) para eliminar cal y abrir los poros.

2. Pintar la terraza y aplicaciones borada en juntas desgastadas entre baldosas.
3. Colocación de cajas y canalizaciones eléctricas en la terraza, permitiendo diferenciar, iluminación terraza, alimentación caseta y cable de red.

1.4.2. Caseta y armario

El material de cortado así como montaje fue presupuestado y construido por una empresa local de aluminio, cerramientos, puertas y ventanas, que es la utilizada por la comunidad de vecinos para la instalación de las separaciones de medianera. En apenas 5 días la empresa encargada terminó la instalación completa de los cerramientos de aluminio, previos a la Semana Santa fig.1.3 En los días posteriores una vez secadas las juntas se continuó



Figura 1.3: Estructuras de aluminio.

con la instalación eléctrica tanto de armario como caseta, con puntos de luces, interruptores, enchufes y cajas internas fig.1.4. Así como un primer aislamiento de paredes y suelo con capa transparente anti-humedad previa a la colocación de mobiliario y alfombras. El



Figura 1.4: Caseta iluminada

mobiliario principalmente ikea o leroy merlin, se fundamenta en almacenamiento o estanterías apropiadamente seleccionadas para optimizar el espacio a un coste competitivo. La mesa y el elevador de pantallas son elementos estandarizados de ikea.



Figura 1.5: Casetas mobiliario.

1.4.3. Presupuestos planificación

La gran mayoría de trabajos realizados a excepción de los montajes de aluminios se han realizado por este autor. Debido a ello la realización de las tareas no solo se ven limitadas por la secuencialidad de las mismas o necesidad de esperas como secado o asentamiento, sino que también se debe tener en cuenta que está mayoritariamente realizadas de viernes-domingo más días sueltos 4-6 h dedicados entre semana. Como contrapartida el coste real es “nulo” ya que no he asignado un precio real hora, pero como queda reflejado en la tabla de presupuesto con un tiempo personal muy monopolizado por la construcción.

Por ejemplo, aunque las tareas previas a la instalación de la caseta no superan los 15 días de trabajo continuado se realizaron durante Febrero-Marzo, casi 40 días laborales y 20 festivos.



Figura 1.6: Panorámica caseta migrada.

En la siguiente tabla se muestran los tiempos y costes de las tareas a realizar, que están dispuestas en orden cronológico de realización.

Tabla 1.2: Presupuesto Caseta Física

Elemento	Descripción	Tiempo	Coste Material
Limpieza Alféizar	Barrer, frotar, aplicar producto limpieza especial, aclarado	4 h	6€
Limpieza Pared	Barrido y agua a presión	1d	0€
Limpieza Suelo	Barrido, fregado, limpieza con salfumán diluido	1d	10€
Pintar Terraza	pintado y limpieza posterior de suelo	2d	35€
Boradas y juntas	Aplicación y limpieza de borada en juntas.	4d	15€
Cajas y canalización	Colocación de cajas y canalizaciones eléctricas	3d	25€
Cableado Eléctrico	Pasar cableado eléctrico, interconexión de cajas y puntos de luz.	2d	40€
Cableado Ethernet	Adecuar cableado desde azotea comunitario, eliminar antena Wimax, pasar cableado por terraza.	1d	15€
Armario Aluminio Aislado	Montaje y sellado de armario por empresa.	2d	1500€
Casetas Aluminio	Montaje de laterales, montaje alféizar, montaje tejado, sellado. Montaje ventana y puerta.	4d	4500€
Luces, enchufes y tubos led	Instalación de cableado, enchufes, interruptores, cajas internas y tubos led de iluminación.	3d	60€
Mobiliario	Compra y montaje de mobiliario ikea	2d	450€
Mesa	Montaje de mesa customizada y eleva pantallas	5h	80€
Extras mobiliario	Separadores, protectores, cubetas ikea etc...	1d	60€
Pizarra	Montaje de pizarras en la chimenea-respiradero	4h	25€
Estor	Montaje y anclaje de estor interior difuminador de luz en ventana.	2h	20€
Mosquitera	Instalar mosquitera plástica fácilmente reemplazable o reinstalable.	1h	4€
Persiana exterior	Compra instalación y sujeción de persiana alacantina externa de pvc.	2d	40€
Cámara Seguridad	IP cam con IA de reconocimiento personas con movimiento automatizado 360°	2h	35€
Alfombra Oficina	Alfombra robusta, oficina, con grosor extra aislante.	2h	80€
Mudanza	Mover el antiguo setup, herramientas, libros etc a la nueva oficina. Mover Impresora 3D, periféricos.	3d	-
Router	Router wifi para caseta y alrededores de terraza.	1h	25€
Switch	Switch Giga ethernet para conexiones por cable de elementos de la caseta	1h	20€
Google Home	Elemento interactivo de smart home dedicado a terraza y oficina	1h	18€

Smart things	Luces de terraza configurables (color, intensidad) accionadas por Alexa/Google assistant. Termómetro, interruptor inteligente.	1h	32€
Cámaras Seguridad	Cámaras de Seguridad Exterior de Terraza y Balcón, + Switch-Router que provee la conectividad a la terraza y las cámaras	2h	55€
Pinguino Calefactor	Pinguino, bomba calor/frió/deshumidificador para 15m ²	2h	350€
Total		35d	7500€

1.4.4. Resultados de construcción

Finalmente en mayo la caseta esta completamente terminada y 100% funcional, siendo utilizada tanto en horario de oficina en mi trabajo, hobbies, formaciones y la elaboración de esta tesis. Así como la ejecución entre Febrero-Abril y un coste final de 7600-7800€, se puede asumir una finalización en plazo así como costes adecuados a lo planificado, no superando desvíos de 3-5%.

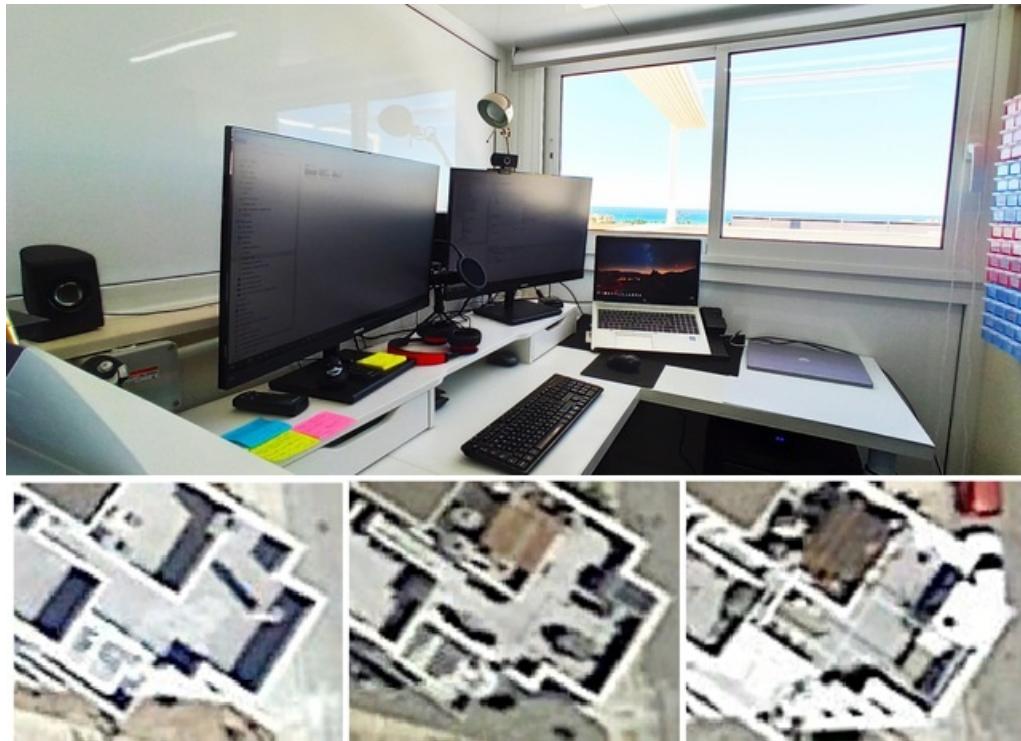


Figura 1.7: Setup en mayo, vista satélite pre pandemia, pandemia, actualidad.

Véase una explicación al pormenor en el anexo B los puntos B.4.4. como la causa y comparativa con setups anteriores, referentes a la construcción B.4.6.1., B.4.6.2. así como

otros puntos menos resaltables tales como iluminación [B.4.6.4.](#) y un conjunto no despreciable de fallos, mejoras y correcciones durante 2023 ([B.5.](#)) que evidencian la continuación del proyecto mas allá de su desarrollo principal.

Por otra parte este proyecto de bricolaje técnico, ha servido de base para la aplicación en otros ámbitos personales del conocimiento técnico/práctico adquirido, como la instalación del sistema eléctrico y solares en un almacén agrícola de mi padre; el cableado ethernet de mi casa, casa de mis padres, suegra, oficinas y en general cualquier amigo que me ha pedido ayuda con cableado ethernet y cámaras de seguridad. Así como muchos elementos del setup, y especialmente la guía[\[16\]](#) y a previsión de mantenimiento para la selección de hardware con fines profesionales a largo plazo.

Finalmente como cliente y promotor de este espacio, me siento completamente satisfecho al pasar una media entre 9-13 horas de mi día a día en dicho espacio y obtener un aislamiento familiar perfecto y separación de entorno profesional y personal.

CAPÍTULO 2. SERVICIOS Y HERRAMIENTAS

Una vez resuelto los temas propiamente físicos, es necesario evaluar las diferentes posibilidades, softwares y estrategias para permitir realizar un trabajo virtualmente. Este tema muestra los diferentes puntos de interés a la hora de generar una nube y sus diversas herramientas necesarias para el día a día.

2.1. Software, hardware y gestión.

En el Anexo [C.1.](#) se especifica el razonamiento tanto de las licencias como del tipo de software base que se utiliza en esta oficina virtual.

El precepto es el uso de **Software libre, auto gestionado**, generalmente basado en versiones **LTS(Long Time Support) soportadas por comunidades** que dependiendo de un análisis coste/precio se auto gestiona o externaliza parcialmente.

Por ello aunque el 98 % de la utilidades y necesidades pueden ser cubiertas por software libre desde el SO (Sistema Operativo) linux, tareas ofimática y especialmente en desarrollo de software, se entiende que una licencia retail tanto de windows, como programas genéricos de ofimática puede no superar los 3-10€ siendo permanentes y fácilmente asequibles, especialmente cuando hablamos de pc/laptop desktop y no de servidores.

Respecto a que auto gestionar y que externalizar, adoptaremos un enfoque puramente económico, minimizando los costes en base a nuestro conocimiento técnico. Por ello la principal externalización son el dominio y servidor de correo, junto a uno/dos VPS (virtual private server), como punto externo de conectividad para aplicaciones esenciales véase anexo [C.3.](#), donde también se detalla el aprovisionamiento y securización anexo [C.2.3..](#)

Requisito	Caso seleccionado	Coste
Hardware cloud base	VPS (2 cpu, 2GB/44G ram, 40 GB disk, red 250/500 Mbps)	4-6 € / mes (dependiente de promoción) [27]
Hardware cloud productos	VPS (2/3 cpu, 4GB ram, 80 GB disk, red 500 Mbps)	8-12 € / mes (dependiente de promoción) [28]
Hardware Desarrollo	power hardware wake up on demand (localhost)	20-40 € / mes coste eléctrico
S.O (base) server	Linux LTS: Debian, Alma o Rocky	0€ LTS community supported
Cloud platform	Ansible & docker & docker-compose/swarm	0€ LTS community supported
Services	Principalmente servicios auto alojados mediante dockerización	0€ community supported

2.2. Ansible

Ansible[29] es uno de los software más utilizados actualmente como herramienta devops, y es la principal herramienta de automatización de este trabajo. Ansible es un software de aprovisionamiento y orquestación, es decir, permite gestionar configuraciones y tareas orientadas a estado de una manera simple y centralizada.

La herramienta utiliza SSH para comunicarse desde un nodo central, utiliza YML como lenguaje descriptivo y JSON como salida. Permite usar una gran variedad de módulos para realizar tareas simples. El objetivo de la herramienta es generar grupos o jerarquías

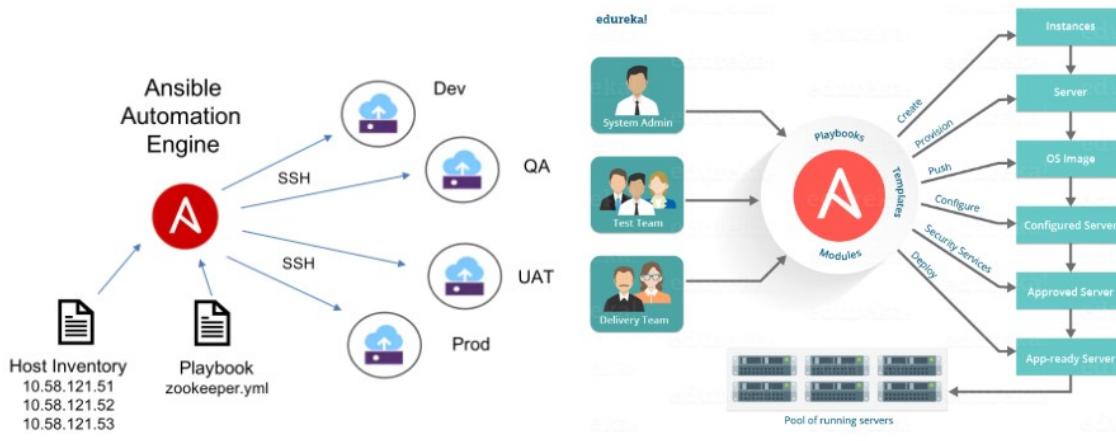


Figura 2.1: Ansible diagrama funcionamiento[143][144].

de servidores a los que conectarse vía SSH con las credenciales apropiadas. Nótese que no es necesario de ningún software previo en dichos servidores únicamente un servicio SSH con la apropiada credencial de login. Una vez conectado ejecuta el comando o las tareas asociadas al script de ejecución (copiar configuraciones, ejecutar comandos, verificar estados). Estas tareas están sujetas a estado, es decir, el YML de la tarea define el estado final que se desea y el script chequea si se cumple o sino ejecuta la tarea para obtenerlo. Las tareas se ejecutan por grupo y/o tag y la ejecución se produce en paralelo en todas las máquinas.

El resultado final es que tras la ejecución del script, recibimos un output coloreado no solo con las descripción de las tareas verificadas, ejecutadas y resultado de las mismas sino con la posibilidad de acceso a un log detallado en formato json (véase fig.2.2).

El verdadero poder de los módulos de Ansible reside en su gran abanico de opciones y softwares soportados. Permitiendo gestionar host virtualizadores o API de proveedores para crear máquinas, dispositivos de red. Ejecutar o acceder la casi totalidad de las funciones de linux y windows server. Configuración y ejecución de software de amplio uso como docker[31], kubernetes[38], git[138], npm, gestores de repositorios o librerías, y la retroalimentación del output de ciertas tareas para la designación de tag o grupo de la máquina ejecutora o como input condicional para otras tareas.

Finalmente Ansible también permite la agrupación de las tareas y ficheros relacionados como rol en una estructura jerárquica de ficheros. Por lo tanto, automatizar una tarea como 'rol'. Esta estructura facilita su puesta en marcha, su actualización y su publicación en Ansible Galaxy (repositorio de roles de ansible), o lo que es lo mismo re-utilizar la gran

```

! ejemplo-apache.yaml ×
ansible-playbooks > ! ejemplo-apache.yaml
1 ...
2 - hosts: pruebascomandos
3   become: yes
4   tasks:
5     - name: Install Apache
6       apt: name=apache2 state=latest
7     - name: Check Apache is running & start on boot
8       service: name=apache2 state=started enabled=yes
9 ...

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS
luis@luis-UX370UAR:~/Escritorio/ansible-playbooks$ ansible-playbook ejemplo-apache.yaml
PLAY [pruebascomandos] ****
TASK [Gathering Facts] ****
ok: [34.71.75.10]
ok: [35.238.41.3]

TASK [Install Apache] ****
changed: [35.238.41.3]
changed: [34.71.75.10]

TASK [Check Apache is running & start on boot] ****
ok: [34.71.75.10]
ok: [35.238.41.3]

PLAY RECAP ****
34.71.75.10      : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
35.238.41.3      : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

Figura 2.2: Ansible ejemplo de ejecución desde proyecto VScode .

cantidad de publicaciones de rol, evitando tener que crear un trabajo desde cero o tener acceso a innumerables ejemplos reales.

Debido al acoplamiento del caso real (Elenkar) y la sensibilidad de ciertos datos, sensibles desde el punto de vista de seguridad hacia futuros ataques, solo se han desvelado parcialmente las partes mas genéricas de dichos script, véase anexo E.4. como ejemplos.

2.3. Contenedores y Docker Service

Cuando se necesita ejecutar una aplicación o servicio se necesitan tres cosas principales, un SO que lo soporte, las librerías externas que pueda usar el software y la configuración e instalación customizada dentro del SO que lo ejecuta.

A excepción de software auto empaquetado como el de Mac OS o aplicaciones snap, la gran mayoría de softwares utilizan librerías (dll, lib) para su funcionamiento (“reusar es de pobres, pero eficiente”). La gestión de las librerías puede conllevar a un problema cuando múltiples aplicaciones del SO requieren de librerías que entran en conflicto, o dejan de ser soportadas tras una actualización o los requisitos de máquinas virtuales de ejecución Python, Java, Php, Node, requieren de diferentes versiones en paralelo.

La necesidad de gestionar servicios pseudo duplicados que actúan sobre un mismo puerto (apache, tomcat, node) o el uso compartido de varias aplicaciones del mismo host-web. No solo hace complejo su gestión sino vulnerable ya que afecta a todos los elementos de manera transversal. La caída del servicio base o la vulneración de seguridad de una aplicación puede exponer el resto de ejecuciones del SO al compartir sistema de archivos, procesos y memoria; y solucionar este problema por VM (Máquinas virtuales) por servicio es poco eficiente y económico costoso.

Un contenedor, no es más que un formato estandarizado que incluye la aplicación, sus

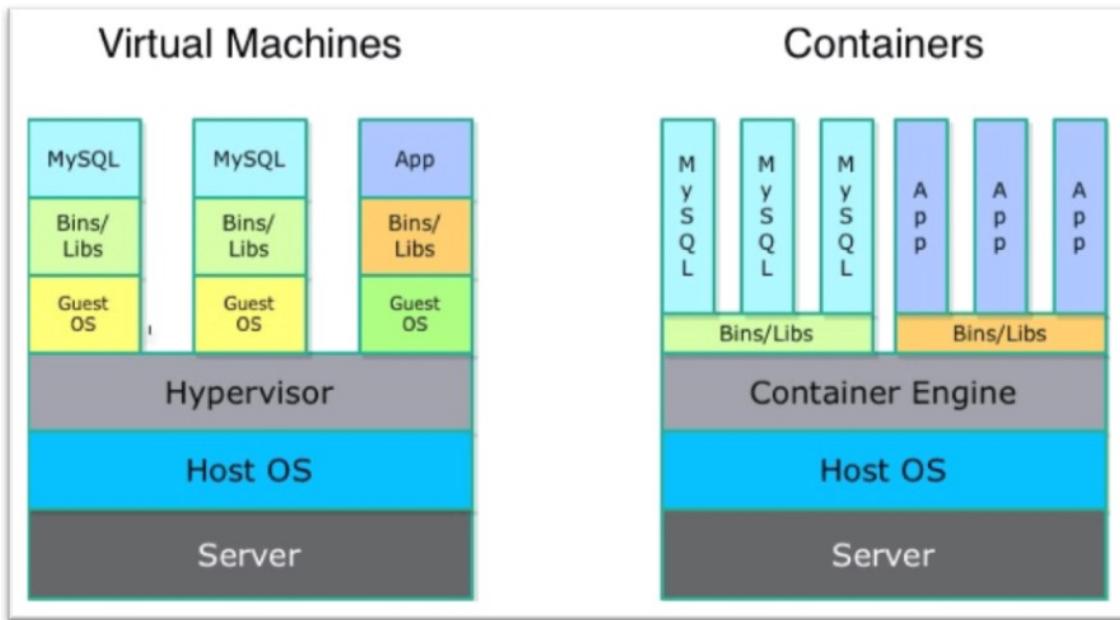


Figura 2.3: Maquina virtual vs contenedores[145].

dependencias y una base de sistema operativo, para una ejecución independiente, aislada y ligera de la aplicación. A diferencia de una Virtual machine, el contenedor no emula hardware, sino que ejecuta la aplicación directamente en el kernel del sistema hospedante, sin embargo utiliza la abstracción a 'nivel de espacio de usuario' (similar chroot, cgroups y namespaces[109]) para que el sistema de archivos, los recursos, los procesos estén aislados en el contenedor. Por otra parte, los contenedores usualmente contienen un único hilo de ejecución que es arrancado al iniciar el contenedor y cuyos cambios no son persistentes, sólo temporales mientras el contenedor está en ejecución.

Docker[31] es un software de código abierto usado como servicio de contenedores. Permite crear, gestionar, abstraer y automatizar el uso de contenedores permitiendo definir redes, puertos, arrancar imágenes, acceder a sus logs o mapear carpetas persistentes dentro del container.

Las imágenes de un container se crean en base a capas (como fotos del sistema de archivos), de tal manera que múltiples imágenes de aplicaciones diferentes pueden compartir capas. Si tengo múltiples imágenes de contenedores, que ejecutan software java, pero el SO (alpine) y la máquina virtual de java son iguales únicamente las capas diferentes, dependencias y aplicación serán propias de cada container. De igual manera la diferencia entre las imágenes y los container en ejecución es una última capa temporal que contiene los cambios del contenedor durante su ejecución. Entendiendo esta propiedad; es factible ejecutar múltiples containers de una misma imagen en paralelo, es decir, permite el escalado horizontal de recursos.

Por último las imágenes se pueden almacenar en repositorios como Docker Hub, que sirven a su vez no solo para arrancar contenedores sino como base para imágenes más complejas, permitiendo desarrollar tus imágenes únicamente dependientes de la capa de aplicación y delegando en comunidades el mantenimiento de aquellos elementos que no son el focus de nuestro producto.

Las principales mejoras de usar docker son:

- Aislamiento, seguridad, gestión simplificada centralizada de múltiples aplicaciones dockerizadas en una única Virtual machine / servidor.
- Rapidez, simplicidad y sencillez en despliegue sin necesidad de instalación.
- Administración del sistema, sin conocimiento previo o mantenimiento sobre la aplicación.
- Consistencia e Independiente de plataforma, misma ejecución en test, local o producción. Facilidad en pruebas o despliegues continuos (CI/CD).
- Modularidad y escalabilidad horizontal, perfecto para entornos de microservicios o ambientes distribuidos.
- Repetibles, replicables y versionables, permite facilitar las migraciones o flexibilizar entornos y desarrollos muy ágilmente.

Por ultimo se debe entender que con el énfasis de la automatización, muchos de nuestros servicios dockerizados necesitan acceder a información de otros servicios dockerizados, por lo que en este documento se utilizan aproximaciones de docker complejas (dind o dood) ve ase Anexo [C.6.1](#).

2.4. Docker Compose, Kubernetes, Docker Swarm

Existen múltiples orquestadores sobre docker. Un orquestador de container es una capa extra de software que interacciona con nuestro engine de contenedores. Su principal ventaja es la configuración o el acceso a una interfaz más humana que el command line del engine.

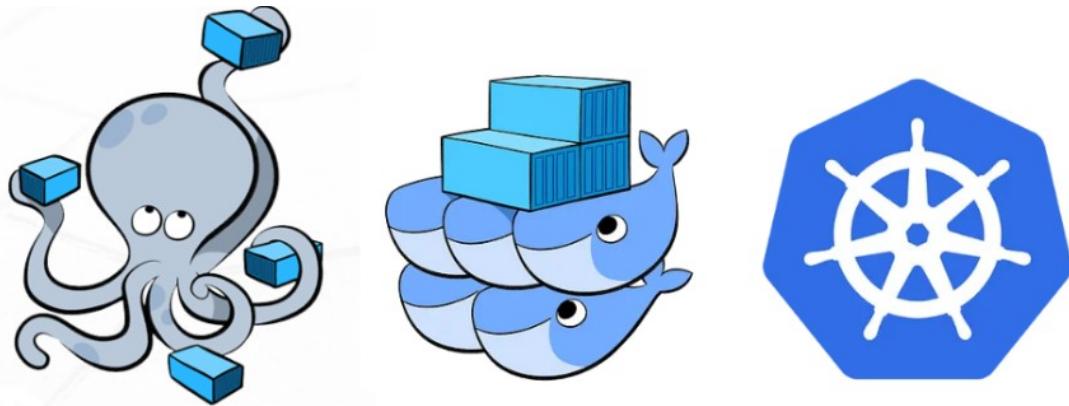


Figura 2.4: Docker compose, swam y kubernetes logo images[\[146\]](#).

Sin embargo otras tareas como la definición de relaciones entre contenedores, sincronía y monitorización también son añadidas a la propia funcionalidad del engine. El mejor ejemplo es Docker-Compose[\[36\]](#), que permite configurar fácilmente en un formato YML multiples contenedores, redes, volúmenes, condiciones y relaciones entre ellos en un único fichero, permitiendo su arranque y gestión. Docker-Compose es el orquestador utilizado

en este documento debido a su facilidad y el foco de una única máquina (VPS) o servidor autocrático, que actúa de manera aislada.

Sin embargo, si fuera necesario, Docker-Swarm^[37] es un orquestador que incluye las funcionalidades de docker-compose y genera un cluster de trabajo, permitiendo unificar múltiples máquinas con servicio docker instalada como un único cluster. Facilitando no solo las herramientas de un cluster tales como master-slave, backups y balanceo de carga, sino añadiendo más herramientas no incluidas en docker-compose, como escalado, monitorización, ingress network point y encriptaciones .

Por último Kubernetes^[38] es un framework de containers, open source creado por google, que gestiona de una manera más profunda y profesional la creación y uso de un cluster de contenedores en la nube. Aunque es mi herramienta diaria de trabajo como desarrollador de software, no será usada en este trabajo puesto añade un complejidad y trabajo adicional no necesario para el nivel de requisitos de nuestros servicios donde **no se espera una gran demanda y se tiene una restricción de recursos escasos**.

2.5. Arquitectura de nuestros Servicios

La arquitectura base de la oficina virtual se fundamenta en la conjunción de tres pilares:

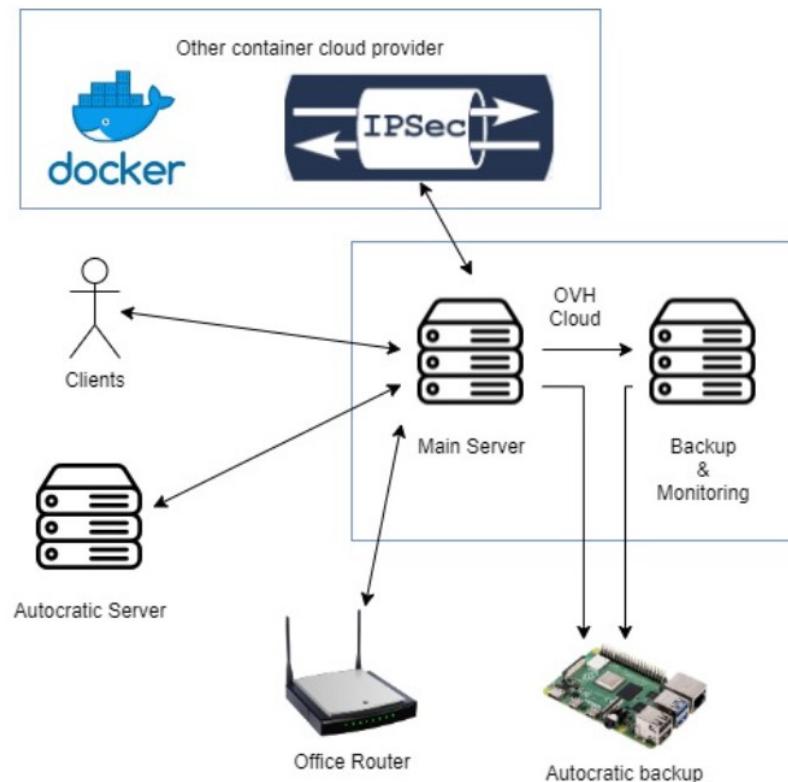


Figura 2.5: Diagrama de interacción.

- Uso de Linux LTS segurizado y automatizado que instala docker como principal servicio, minimizando y facilitando el mantenimiento de la máquina.

- El uso de docker y orquestadores de docker para proveer de los diferentes servicios necesarios y la interacción de los mismos.
- El uso de Ansible para el aprovisionamiento del servidor, la seguridad y configuración de los servicios, así como la conectividad y configuraciones de elementos conectados al VPS por una VPN o túneles.

Los servicios públicos tales como web publica, proxy o vpn son expuestos directamente.

El servidor principal, permite conexiones reversas a los servidores-granja de producto o servidores autócratas re-enrutando el tráfico. Un servidor secundario sirve como gestor de mantenimiento, monitorización y backup del principal, así como un backup autocrático en local con raspberry pi almacena el histórico de backups y configuraciones.

En todo momento, la idea es poder crecer en una nube externalizada o local, así como mantener en todo momento un control de backup. El proxy permite exponer servicios hospedados en nube externa o local a través del servidor principal VPS. La VPN permite conectar a servicios no expuestos a internet, pero si en la intranet.

Todo ello fácilmente gestionable por ansible, docker-compose que junto a los backups de configuraciones y volúmenes permiten replicar, migrar o restaurar todo en cuestión de minutos.

2.6. Aprovisionamiento y Securización

El aprovisionamiento principal es configuración, creación de grupos, permisos, eliminación de todo software innecesario del servidor, añadir docker, actualizaciones desatendidas y la autorización de claves ssh sobre un usuario administrativo de deploy.

La principal baza de seguridad de este proyecto se basa en la simplicidad, aislamiento y el uso de docker como elemento separador entre aplicaciones.

Por lo tanto, blindando del firewall y servicio SSH, la seguridad se basa en la ausencia de "accesos" y la "compartimentación" de elementos. A excepción de los servicios públicos dockerizados y SSH no hay más servicios expuestos en la ip pública del VPS aislando y protegiendo el SO. Así mismo aquellos servicios que no requieren de ser expuestos públicamente, se utilizan internamente vía vpn - intranet.

Por lo general se puede resumir la securización del server en los siguientes conceptos:

1. Purga de elementos vulnerables o no usados del sistema.
2. Denegación de todo puerto no autorizado. Firewall ofensivo con denegación de ip.
3. Configuración securizada de ssh. Solo permitidos non-root, vía claves asimétricas y valores default cambiados.
4. Control con filelog, logwatch, logcheck, accounting para auditorias.
5. Log rotate, y mecanismo de mitigación de ataques como fail2ban o denyhosts.
6. Exponer el mínimo de servicios externamente y usar VPN para el resto.

7. Scripts de control, sudo limitado y notificaciones de procesos de superusuarios.
8. Usar VPN e integrar una nodo (raspberry pi o similar) con NIDS (Network Intrusion Detectors System) como prevención en redes locales interconectadas.
9. Usar imágenes docker oficial o auto-ensambladas, que siguen buenas prácticas o escáneres de seguridad (docker hub lo incluye).

Véase rol del Ansible de segurización [160] y el anexo C.2. .

2.7. Servicios

En este apartado se indican los servicios seleccionados como nube MVP(Minimum Viable Product), en el anexo C.3. se detallan razonamiento, posibles candidatos y una argumentación sobre cada uno de los principales servicios. En la tabla 2.1 y 2.2, indican como obligatorios los marcados con un a **. En el caso de que la nube se aplique a un grupo de trabajo de 10 o mas personas se incluyen los siguientes servicios:

Tabla 2.1: Docker services más de 10 personas.

Tipo de Servicio	Herramienta	Detalles	Tipo y Coste
Comunicación Interna >10 trabajadores	Rocket Chat[78]	Requiere recursos y mantenimiento continuado, es decir, servidor dedicado	Interno VPS dedicado 3-5€ / mes
Autenticación Centralizada	KeyCloak[86] con LDAP[85]	Utilizar un LDAP que suministra información a KeyCloak y automatizar la carga de LDAP	Interno 0€ VPS principal

Es importante recordar que en las pruebas de concepto anexo E.1. no se ha automatizado el proceso de configuración de interacción entre los servicios, en el caso de autenticación.

El tuning de los plugins o las third parties, debe hacerse manualmente durante la primer arranque de la nube. Este proceso puede automatizarse pero requiere un conocimiento específico de cada servicio y tiempo dedicado excesivo, por lo que se ha priorizado la prueba de múltiples servicios y la selección del más conveniente, ya que una vez configurado apropiadamente, puede usarse el primer backup como producto pre configurado de la nube.

Finalmente recordar que no es necesario el uso de todos los servicios, es decir, la empresa u oficina virtual debe hacer uso en función de sus necesidades, tanto en tamaño de grupo de trabajo, criterios de seguridad y monitorización etc ... deben ser coherentes con las necesidades.

Tabla 2.2: Servicios docker nube MVP.

Tipo de Servicio	Herramienta	Detalles	Tipo y Coste
Correo Electrónico *	Don Dominio[40]	Externo 10 € / año 10 cuentas 3 GB o 20 € / año 15 cuentas 10 GB.	Externo 10 € / Año
Comunicación Externa*	Teams, Zoom, Whatsapp, Telegram	Centralizadas desde cliente Fermi.	Externa, 0€
Comunicación Interna	Teams / Slack	Es un elemento excesivamente costoso, en recursos. Se usara servicio gratuito limitado.	Externa, 0€
Almacenamiento*	Nextcloud[57]	Requiere recursos, es decir, servidor VPS dedicado o via intranet a un server autocriptico.	Interno 3-5€ / mes
Aplicación ofimática	Only Office / collabora[58]	Hosteado junto a next cloud en un server dedicado.	Interno incluido nextcloud server
Wiki*	Bookstackapp[60] / MediaWiki[59]	Ligero y útil para desarrollo.	Interno 0€ VPS
VPN*	Wire Guard[88]	Punto externo de conexión intranet, generación de múltiples VPN.	Interno 0€ VPS
Repositorio de Code	Gitea[68]	Si es publico en servidor VPS principal, si es privado puede estar en un server autocriptico.	Interno 0€ VPS
CI/CD	Jenkins[69] / Drone [70]	Si es publico en servidor VPS principal, si es privado puede estar en un server autocriptico.	Interno 0€ VPS
Web Service*	WordPress[72] / Static web[75]	Simplemente como pagina web principal y/o blog	Interno 0€ VPS
Ticketing	Taiga[92]	Simple y eficiente board de scrum-agile.	Interno 0€ VPS

2.8. Múltiples Docker compose y limitaciones

Para la puesta en marcha tanto de servicios externos, así como servicios no expuestos debemos de entender que muchos de ellos tiene bases de datos o servicios auxiliares, es decir por ejemplo el servicio de web puede ser un único container con todo (db,wordpress,web server, php) o puede ser múltiples container aportando cada uno su propio sub-servicio. Por lo tanto ya sea por catalogación o por simplicidad no es práctico la generación de un único docker-compose con todos los servicios.

Lo óptimo es la definición de servicios o grupos de servicio bajo un único docker-compose, facilitando la tarea de instaurar un servicio systemD[110] de Linux, que arranque los servicios junto con el Sistema operativo. Otro elemento útil es el uso de “Múltiples file call”[111] en casos complejos o de elementos comunes permitiendo importar sub-ficheros yml para conformar un docker-compose final o sobre-escribir un fichero con otro [112]. Par aun mayor detalle véase anexo C.6..

La principal limitación de docker-compose es su aplicación en un único host, lo que limita su uso a entornos de desarrollo/testing o como nuestro caso aquellas herramientas que no requieren de una alta disponibilidad y demanda, ya que no implementa de manera nativa ningún tipo de escalado o balanceo de carga. Sin embargo si utilizamos versiones superiores a la V3, es plausible deployar directamente ficheros docker-compose en cluster de docker swarm previamente inicializados facilitando la evolución hacia un escalado horizontal de recursos.

2.9. Docker Backups

Como se ha mencionado en el anterior punto, se ha priorizado la configuración manual de plugins o interrelaciones entre los servicios, sobre la “auto-creación por script” ya que requiere de un tiempo y esfuerzo sobre dimensionado para el objetivo de este trabajo, que es tener una nube, pero no auto-configurar ágilmente.

Sin embargo sí es objetivo que esta nube se pueda construir y destruir o realizar una restauración ágilmente, es decir, se definen y configuran las relaciones entre servicios, se realiza un backup que puede ser usado como base de construcción para nuevos despliegues (siempre y cuando se mantenga la selección de servicios e interrelación). Por lo tanto la generación de backups periódicos para posibles restauraciones, así como el despliegue de backups como nuevas nubes es un elemento obligatorio.

Toda la información que los servicios desplegados contienen se fundamenta en las capas variable de los contenedores y en los volúmenes persistentes, es decir, con el apropiado backup de volúmenes, los contenedores son capaces de re-arrancar con todo apropiadamente configurado.

Muchas veces los servicios dependen de otros servicios auxiliares tales como bases de datos, en estos casos dependiendo de si se aplicado una política de centralización en una única base de datos o en múltiples, puede hacerse el backup “por servicio” exportando sus schemas, o realizarse directamente un backup del volumen de la ddbb.

En todo caso el procedimiento siempre sigue una pauta marcada, que únicamente es influenciada por los recursos que disponemos en el VPS:

1. Parada de servicios, excepto ddbb o similares.
2. Backup de ddbb o servicios auxiliares por script.
3. Apagado de todo servicio y realización de backup de volúmenes.
4. Reiniciado de servicios.

El punto crítico es la capacidad de procesado así como el espacio limitado en disco.

Existen múltiples puntos de vista a la hora de realizar el backup usualmente se utilizan métodos de diferenciado binario con checksum (similar a git) y compresión para generar layers de cambio de las que se almacena un número máximo de veces.

Existen software especializados también dockerizados como autorisc[101] o duplicati[102], sin embargo el punto principal de este backup en el VPS es minimizar el uso de recursos por ello se sobreentiende que verdaderamente el backup es una sincronización en red, utilizando software como rsync[104], que sincroniza los folders de los volúmenes con un servidor remoto y es allí donde se realiza el proceso de backup propio y almacenado del último backup en un medio donde la capacidad no es un problema, ni existen limitaciones de cpu.

Entiéndase también que la realización del backup es automática-global, aunque se permita por la estructura de carpetas y subcarpetas la realización parcial de backup manualmente.

2.10. Automatizaciones y scripts

Una vez seleccionados los servicios necesarios para nuestra nube y su prueba de concepto validada, es necesario la realización de automatizaciones para su uso diario y gestión:

- Ansible y script de despliegue, son principalmente scripts de ansible que deben permitirnos un despliegue rápido desde cero de todos nuestros servicios. Dicho despliegue debe validar y llamar a script de securización y aprovisionamiento si las máquinas elegidas, si no están apropiadamente predispuestas.
- Apagado, encendido de la plataforma, haciendo énfasis en la automatización de muchos servicios o grupos de servicios como servicio de systemd[107], incluyendo la inicialización de los mismo al reiniciar la máquina linux (véase anexo C.6.2.).
- Cron, script o ansibles de backups, purgado de datos y chequeos de seguridad y actualización. Su principal función es verificar la seguridad y consistencia de las máquinas linux, gestionando un apagado para la realización de backups, limpieza y reinicio de la plataforma principalmente en horario nocturno (3-5 am).
- Validación de ecosistema, los anteriores apartados hacen referencia a máquinas exclusivamente centradas en VPS o servidores, pero es necesario una validación y gestión del cloud en todas las características añadidas, redes, conexiones vpn a otros servidores o entornos de trabajo.

2.11. Legalidad

Este apartado indica únicamente una tendencia o recomendaciones a la hora de evaluar los requisitos y riesgos legales de nuestra nube. Como ciudadanos europeos o empresas que trabajen dentro de el área económica europea, debemos tener claro 2 pilares fundamentales:

- Cumplir la carta de derechos fundamentales de la UE y directivas o reglamentos europeas de protección de datos[114], ya que estas incluyen las leyes nacionales[113] de los diferentes estados en temas de protección de datos, derechos u obligaciones en el mundo digital europeo.
- Servidores europeos o equivalentes, es decir, muchos de los requisitos necesarios se adquieren al usar un proveedor europeo con sede en Europa, o en países homologados para datos europeos. Este punto es especialmente conveniente, debido a que raramente se utilizan nubes lejanas geográficamente debido a la degradación de propiedades de red como el ping, availability, bw que degradan la conexión.

La recomendación como desarrollador que impera, para evitar la complejidad y el sobre esfuerzo de tratar con datos sensibles, es "**utiliza única y exclusivamente aquellos datos realmente necesarios**", haciendo especial énfasis en evitar aquellos datos de especial sensibilidad como religión, tendencia sexual, etnia, afiliaciones políticas o sindicales, genéticos, biométricos identifica-torios, relativos a historial de salud o datos traceable.

La segunda recomendación es no vincular datos personales con otros conjuntos de datos almacenados, facilitando la eliminación de los mismos en el plazo de conservación de los datos de carácter personal.

La tercera recomendación es, segmenta, clasifica tus datos o modifícalos degradando su sensibilidad. Degrado o hacer anonimo un dato significa implicar tener un dato sensible como la MAC, IP u otros identificadores que por su carácter único y identificativo es necesario. Se pueden transformar vía hash o checksum obteniendo un segundo valor que no tiene la clasificación de datos personal pero mantiene su carácter único y traceador para nuestro aplicativo.

2.11.1. Clasificación de datos e inventario

Es importante inventariar y clasificar los datos utilizados en la nube o software. Se recomienda una clasificación a tres niveles:

1. Confidencial: un dato que requiere no ser expuesto y por lo tanto no puede almacenarse en claro. Limitando el acceso al valor.
2. Activo critico: es un dato confidencial con integridad y 'availability', es decir, permite verificar su no modificación y un procedimientos de acceso garantizado.
3. Datos personales identificables, es un activo critico de carácter personal, por lo que incluye una clausula de tiempo de almacenamiento y eliminación el tiempo máximo legal.

2.11.2. Consentimiento y mecanismo de acceso

Cuando obligatoriamente necesitamos un dato sensible, es muy importante que el tratamiento se base en el consentimiento del interesado, el responsable deberá ser capaz de demostrar que aquel consintió el tratamiento de sus datos personales. Por lo tanto el consentimiento debe ser inequívoco y explícito, ya que tiene la responsabilidad jurídica de demostrar que el usuario ha dado su permiso. Como ejemplo **no se puede poner pre-marcado por defecto la aceptación de política de privacidad**.

Por otra parte en todo momento existen unos procedimientos[115] para atender las demandas de los clientes sobre sus derechos, y dichos procedimientos deben ser visibles, accesibles y sencillos.

CAPÍTULO 3. REDES Y CASUÍSTICAS

Este tema resume y expone ejemplos llevados a cabo en la prueba de concepto de redes para la oficina virtual. Es de especial interés para comprender el detalle realizado la lectura del Anexo [D](#).

3.1. VPN e intranet

Primeramente hemos de entender que siempre existe una “red común”, por lo tanto ya sea como freelance en mi oficina en casa, o la sede física de una pequeña pyme o una gran oficina en un edificio de cristal en el centro metropolitano, o una red en un cloud siempre existe una **red principal**.

3.1.1. Objetivos de VPN

Normalmente los servicios internos están accesibles dentro de esta red y bloqueados externamente. Por lo tanto tenemos múltiples intereses para implementar una VPN o virtual private network, comúnmente llamadas intranet, que nos permita el acceso a esta red interna:

- Conexión directa con otros elementos de la VPN, es la funcionalidad básica de una red, permite interconectar pc, impresoras o varias personas, pero usando una interfaz virtual, es decir, cada uno usando como soporte su LAN-real. **Especialmente interesante para utilizar elementos de control remoto** o monitorización.
- Conexión entre redes, en muchos casos puede ser de interés que múltiples LAN (múltiples sedes) sean también accesibles entre sí mediante la red principal sin necesidad de ser expuestas a internet o alquilar un cable-túnel a un proveedor de telecomunicaciones.
- DNS, monitorizado y filtrado. Especialmente útil para monitorizar. Usualmente las redes tienen su propio dns, que aparte de añadir servicios interno, permite filtrar llamadas dns o implementar otros mecanismo útiles anti spam[\[116\]](#).
- Salida predeterminada. En algunos casos nos interesa que la IP desde la que salimos sea nuestra red principal, ya sea por temas de legalidad, accesos concedidos u otros servicios que filtran por ubicación o ip.

Para entender adecuadamente cómo funcionan las VPN hay que refrescar conceptos básicos de Redes véase anexo [D](#) tales como 'NAT', 'Enrutado', 'DNS', 'proxy reverse', así como elementos utilizados en nuestro despliegue. .

3.1.2. Tipo de enlaces VPN

En base a la topología o complejidad o tecnología utilizada existen diferentes tipos de enlaces, en este trabajo asumimos siempre VPN a niveles L2 o L3 de la capa OSI[\[120\]](#),

es decir equivalentes a una red cableada.

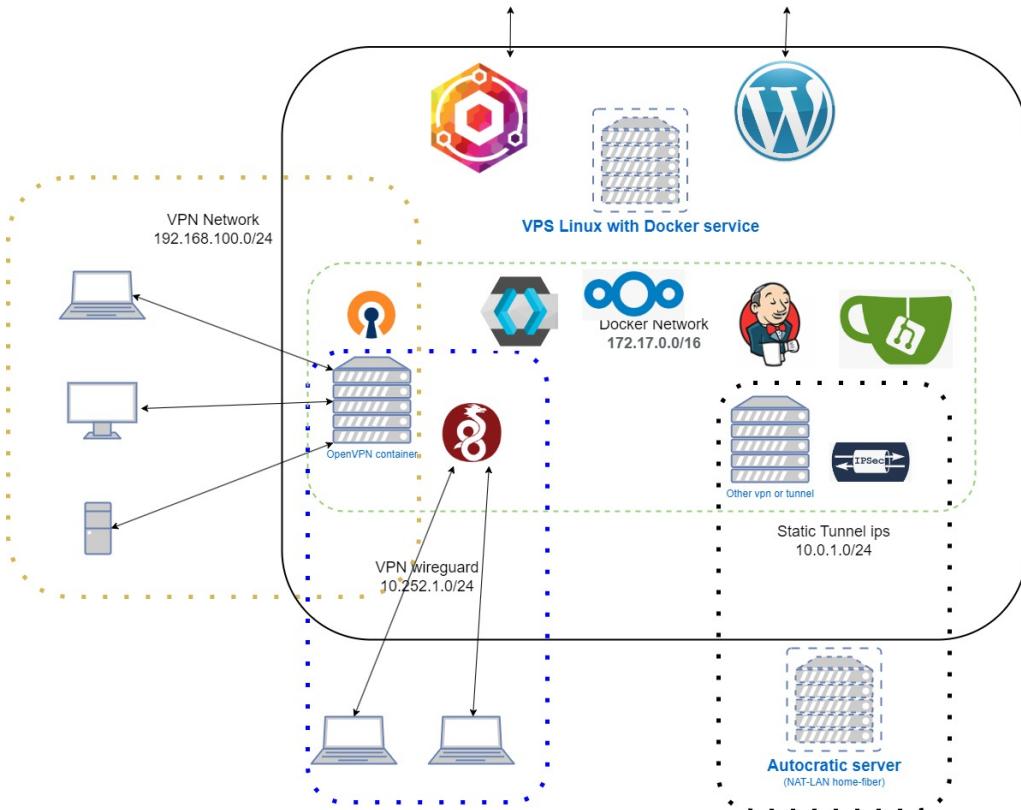


Figura 3.1: Casuística de interés de VPN - VPS.

- Túneles P2P, es decir, conexiones puntuales para interconectar redes o servidores. Usualmente estáticas, utilizadas como red de transporte troncal. Es una mono-red ya que es un único enlace entre dos nodos y se les denomina como túneles ya que una vez entra el tráfico en ellas sale en el destino.
- VPN como servicio-salida a internet, aunque puede ser entendido como un nodo central de enlaces p2p, formando una red al re-enrutar tráfico en el nodo central, su objetivo radica como puerta de enlace para conectar clientes externos. Su objetivo principal es acceder a intranets y geo-localizar el cliente dentro de la intranet, saliendo al exterior con IP y legalidad del servidor VPS.
- Conectar o exponer elementos de manera sencilla sin necesidad de IP pública, pre-configurado de nat-reverse y firewall. Se utiliza un cliente de una conexión reversa a una VPN como servicio-salida, evitando problemas intermedios tales como NAT, firewall, CG-NAT^[26] de la red proveedora de conectividad, el objetivo en este caso no es el enrutado sobre la vpn, sino lo contrario, la exposición de un servicio en la propia VPN a internet a través de IP pública del VPS que ofrece la VPN.
- Seguridad y compartimentación, la red así como las intranet son una capa más de seguridad por aislamiento al cifrar una comunicación y limitar el acceso a las comunicaciones e imponer una barrera de acceso a recursos únicamente ofrecidos dentro de la intranet.

3.2. Casuísticas de interés

Obviando la conexión directa de elementos públicos o privados (detrás de NAT), o la definición propia de VPN para usar dicha red en nuestros objetivos existen un conjunto de situaciones que se resuelven con una implementación por defecto de VPN.

3.2.1. Interconexión de redes privadas por VPN

Este caso es de interés muy habitual, múltiples LANs, especialmente si son redes de oficinas regionales, desean estar interconectadas de manera privada, es decir equivalente a interconectadas por cable, para permitir acceder a los diferentes servicios o clientes de cada una de ellas. Se define una VPN, la cual actúa como red de transporte en forma de

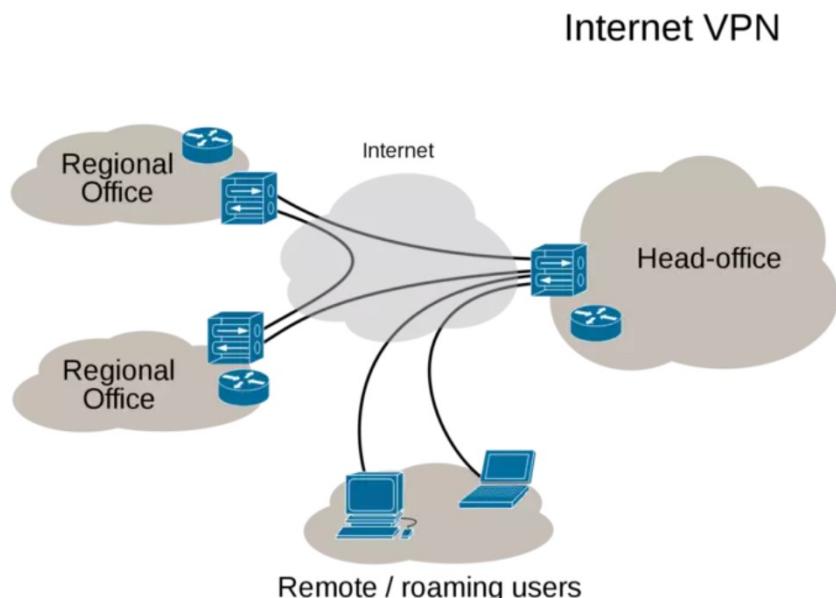


Figura 3.2: Diagrama interconexión de redes por VPN[147].

estrella, e incluye unas reglas de enrute para anunciar y permitir la comunicación entre Lan - VPN- Lan. Véase anexo D.1.3. con un caso detallado y explicado.

Los principales servicios más beneficiados son IT (soporte), servidores de almacenamiento en red y el uso de protocolos de comunicación (FTP, PING ..) entre elementos de ambas redes.

Un punto importante de la interconexión de LANs y VPS es el uso extensivo de captación de imágenes de seguridad y sensores tales como cámaras ip, sensores wifi o gateways de sensores (zigbee, z-wave o bluetooth) ya que son sistemas que suelen estar aislados a LANs no expuestas y no accesibles.

Se debe entender que no es necesario instalar un cliente VPN en cada elemento de la LAN, sino definir aquel elemento que “enruta” el tráfico por la vpn o utilizar un router con cliente VPN (véase anexo D.1.3.).

3.2.2. Múltiples capas de VPN

Desde la perspectiva de seguridad pero especialmente para la separación de conocimiento o grupos de trabajo existe la necesidad de una VPN principal (intranet) para acceder a los servicios esenciales y la conectividad una vez ya dentro de la intranet de otras VPN para redes concretas no expuestas por el VPS. Esto también aplica a la gestión UI de la propia VPN-layer2 evitando puntos de ataque por no estar expuesta a internet.

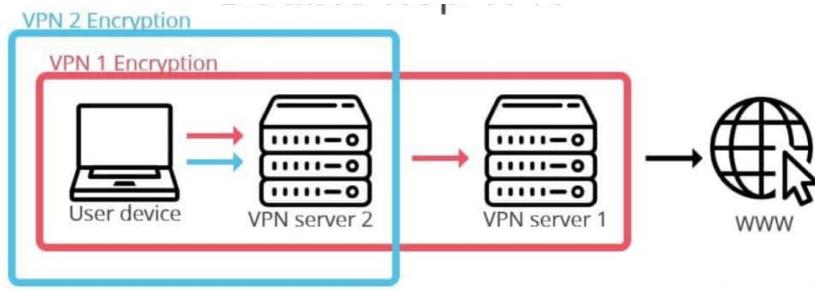


Figura 3.3: VPN multi-salto, multi-capa.

De especial interés también puede ser el uso de una o mas tecnologías vpn, combinando diferentes capas y tecnologías.

3.2.3. DNS filtro y espejos.

Un servicio DSN-proxy interno es de especial necesidad para ofrecer una interfaz intuitiva al acceso de los servicios internos, así como filtro de seguridad capaz de filtrar publicidad o accesos web inseguras o en black list similares a pihole[116]. Dentro de la VPN es

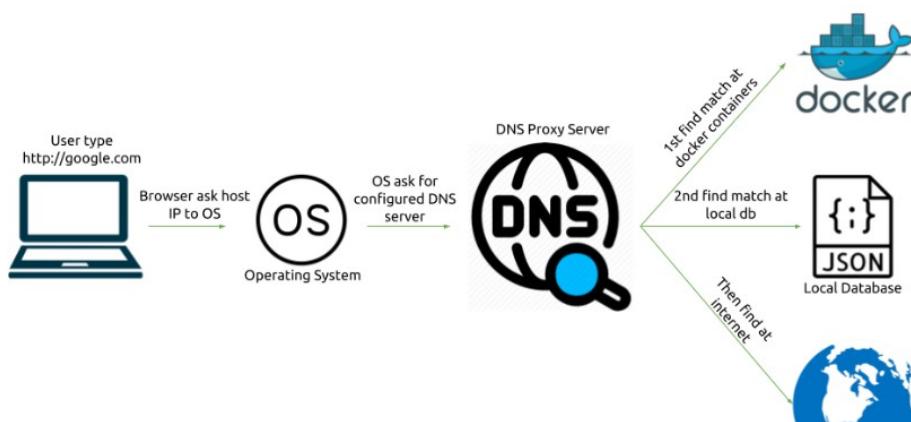


Figura 3.4: Diagrama DNS proxy[157].

de especial utilidad puesto que los servicios dockerizados no expuestos no tiene una IP estática, por consiguiente la asignación de un DNS interno que contenga los dominios internos dentro de la VPN es algo necesario (véase anexo D.1.5.).

Finalmente existe la necesidad de creación de repositorios espejos, es decir, se bloquea el acceso a repositorios de uso común como dockerhub, maven, node etc.. y se crea

un repositorio interno espejo que cachea o generan repositorios “aptos” (aprobados por seguridad o pendientes de auditar), para su uso interno como espejo de los públicos.

3.2.4. Exposición externa vía VPS

A veces no disponer de recursos, una ip pública en una sede o restricciones legales es un impedimento para ejecutar servicios en dicho lugar geográfico. En estos casos se utiliza un elemento público adecuado (IP y ubicación) para tunelar tráfico (vía VPN) a nuestros recursos privados.

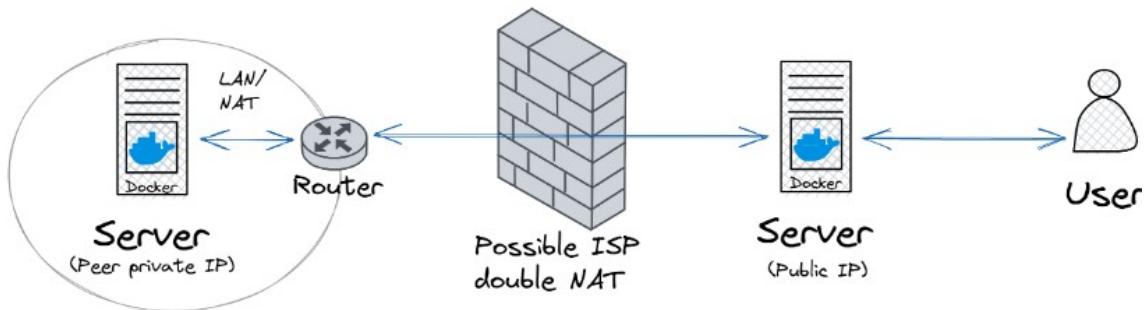


Figura 3.5: Conexión a recursos detrás de CG-NAT / Firewall o NAT.

Por otra parte la exposición de los recursos detrás del túnel pueden ser expuestos utilizando una técnica de "proxy reverse", que es también utilizada como automatización de multi-dominio en el propio servidor VPS. Otro interés puede ser la atenuación de ataques o recopilación de estadísticas si nuestra nube de gran potencia pasa por proxy en un VPS, fácilmente bloqueable o parapetado con un firewall.

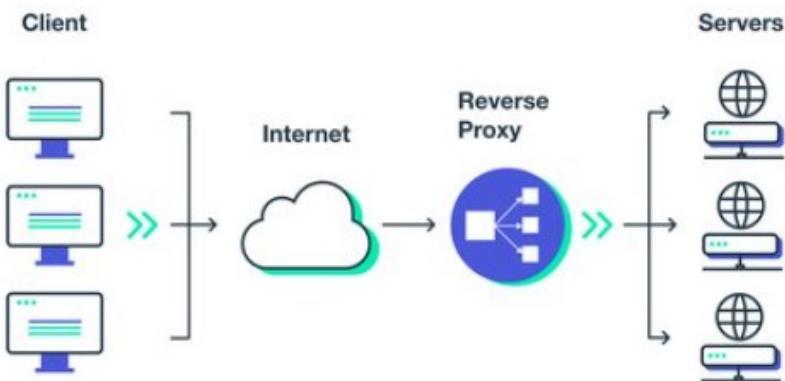


Figura 3.6: Reverse proxy diagram.

Un ejemplo simple es la reconversión de una vieja bodega en un pequeño cluster de servidores para un proveedor de pymes. Probablemente el aislamiento y temperatura de la bodega, es idóneo para la refrigeración y uso continuado de los servidores, así como la utilización de recursos verdes para generación eléctrica o refrigeración activa fácilmente amortizables. Sin embargo el coste de un enlace público o geo-localizar la ip del cluster

puede tener costes elevados(económicos o de seguridad), mientras que la contratación de dos proveedores de fibra, debido a las altas capacidades actuales (300-1000 Mbps) permite un uso profesional de enlaces de público general.

3.3. Docker y automatización de redes

Docker permite crear diferentes tipos de redes virtuales e interconectar los contenedores docker a una o más de ellas. Aquellos contenedores públicos, tiene los puertos directamente mapeado a la red hospedante del vps, sin embargo todos aquellos servicios no expuestos (db, dns, servicios privados) para evitar la interconexión de todos por una red bridge default y su vulnerabilidad de seguridad, se definen diferentes redes por temática (auxiliar, wordpress, vpn-layer1, vpn-layer 2 ...).

Los contenedores que pertenecen a más de una red, se interconectan a ambas donde la red default gateway es la primera conectada en orden alfabético.

Así mismo recordemos que docker contiene un dns interno que es capaz de redirigir nombre del servicio, id, hostname, alias de red. Esta redirección depende del contexto, es decir, si dos contenedores están en redes diferentes aisladas, la resolución no devolverá ip alguna, en caso de haber conectividad redirigirá a la ip interna de la interfaz de conexión.

3.3.1. Automatizaciones

Con el fin de evitar la creación de configuraciones estáticas o complejas, se ha optado por un enfoque automático y auto generado, es decir, cada vez que desplegamos un contenedores en un red de docker aquellos servicios como proxies, dns, portainer y otros elementos monitorizadores deben conocer de la existencia de los nuevos contenedores y auto configurarse para su correcto funcionamiento.

Esto se consigue mediante Docker out of Docker[33] véase anexo C.6. y dockergen[34], estos mecanismos permite acceder desde los contenedores a meta-datos de otros contenedores en ejecución, especialmente variables, labels y propiedades de redes.

3.3.2. Reverse Proxy y HTTPS

La primera automatización es el uso de un reverse proxy con certificados https automáticos, es decir, aquellos contenedores públicos se suscriben junto a un Traefik[123] o Ngix[124] proxy que gracias a un container de let's encrypt[122] y las utilidades de Dood[33] y dockergen[34], identifican los hostname, dominio o alias de los container, generan el certificado de let's encrypt y la configuración de reverse-proxy hacia los puertos indicados, véase flechas verdes figura 3.7 o prueba de concepto anexo E.1.5..

Por consiguiente únicamente con definir las variables y deployar en la red docker, el contenedor es accesible via el puerto 80/443 con https certificado públicamente hacia el dominio preseleccionado en las variables del contenedor (únicamente requiere tener registrado el dominio y la redirección pertinente hacia el VPS).

3.3.3. Dns automatizado

De una manera similar, puesto que docker internamente auto gestiona un dns propio, util, y actualizado es posible la utilización de dns proxy-relay, utilizando como master el dns interno de docker basado en Docker out of docker[33] o la extensión del contenedor con resolver basado en el dns docker.

Por lo tanto en base al dns master de docker, resolverá la petición en local (hosts files y resolv de VPS hospedante), después resolverá los valores entre los diferentes valores docker internos, se puede configurar un segundo nivel local en base de datos local del contenedor dns (json, db, dominios internos seteados por UI manualmente) y finalmente llamará a los dns externos que utiliza el VPS.

Es por lo tanto un sistema completo de dns en si puesto que permite todas las peticiones dentro de la red docker. Debido a que es posible situar el servicio de VPN en dicha red y utilizarla como default gateway, obtenemos un dns 100% funcional para peticiones internas como externas, al utilizar el VPS como default gateway a internet (véase figura 3.7).

3.3.4. Interconexión de Red docker, VPN y gateway

Existe un problema, para permitir la conectividad vía default gateway via VPS. No es posible salir a internet con IP internas de una VPN, es decir privadas. Es necesario la utilización de un mecanismo de NAT-Masquerade, para que sea la IP del VPS.

Este mecanismo esta proporcionado nativa mente en docker, por ello desde dentro de un container podemos acceder a internet, ya que entre las ip publica del VPS y las redes internas de Docker hay un NAT funcionando. Sin embargo desde la VPN únicamente esta habilitado el enrutado a otras redes, es decir, requiere de una NAT entre la red VPN y la red interna de docker que el servidor de VPN tiene como default gateway.

Por otra parte este NAT red VPN hacia las redes del contenedor-servidor VPN, puede tener especial utilidad, ya que las NAT se pueden definir por interfaces o dominios de redes. Esto nos permite hacer que toda petición proveniente de la VPN parezca estar originada en el propio VPN-server container, el cual si esta conectado a mas de una red (default gateway u otras), nos permita una comunicación de la VPN hacia los servicios dockerizados. Esta comunicación es unidireccional, es decir debe ser preestablecida en ese orden para ser bidireccional, por lo que inhabilita la posibilidad de acceder desde los contenedores dockerizados a IPs concretas de la VPN.

Otro elemento interesante es la declaración de dominios internos que apuntan a ip's internas de la VPN para servicios entre sedes. Aunque dichas ips no son accesibles desde los servicios dockerizados en el VPS, si lo son desde la VPN, por lo que el servicio dns configurado manualmente es funcional para todo elemento conectado a la VPN.

Como conclusión, aquellos elementos conectados a la VPN, pueden:

- Usar la VPN como default gateway, saliendo a internet por el VPS (flechas rojas, figura 3.7).
- Definir como servicio de dns, el contenedor dns-proxy interno, permitiendo resolver dominios internos y utilizando los dns externos apropiados para la ip del VPS.

- Conectarse y utilizar todos los servicios expuestos en las redes internas de docker a los que el contenedor servidor VPN esta conectado.
- Conectividad con las LANs de sedes, si estas son expuestas a través de la VPN si existe un elemento conectado a ambas (VPN y LAN) configurado como gateway LAN-VPN-LAN.
- Exposición remota de servicios, en aquellas VPN de mecanismos inverso (flecha negra figura 3.7).

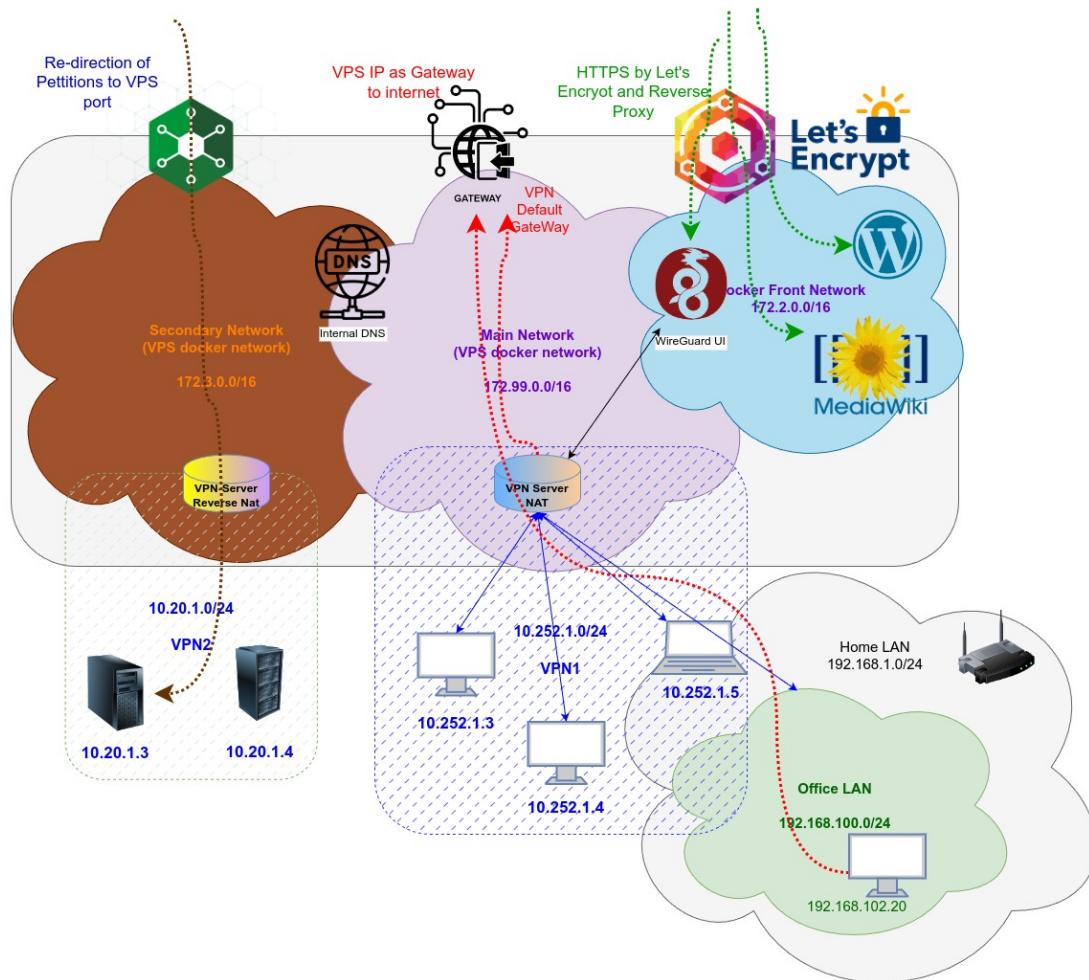


Figura 3.7: Diagrama de interconexión de redes docker y servicios.

Por otra parte, si deseamos que nuestra VPN permita un acceso reverso, es decir enviar peticiones o tráfico desde la red docker a nuestra IP en la VPN o nuestra LAN expuesta por dicha IP-vpn, no es posible la utilización de los servicios dockerizados, o el uso de la VPN como default gateway, ya que requiere implementar un Nat-reverso docker-network a VPN (véase flechas marrones figura 3.7), o una configuración específica (estática) de DMZ para permitir conexiones específicas a través del NAT.

Por último para la interconexión de sedes, es indiferente, ya que las sedes pueden comunicarse a través de la red VPN, sin necesidad de pasar por las redes docker y no se ven afectadas por ningún tipo de NAT, únicamente es necesario tener un elemento conectado a la VPN en dichas LAN y la configuración adecuada.

3.4. Caso desarrollado

El caso desarrollado se ha centrado en la implementación de 3 VPN, focalizados en tres objetivos diferenciados (véase figura 3.8).

Una primera VPN1 cuyo objetivo es la interconexión de elementos de la VPN(principal), la salida como puerta de enlace por el VPS y el acceso a servicios dockerizados internos (DNS, web, internos ...). Una segunda VPN2 cuyo principal objetivo es la interconexión de LANs o enrutado. Y por último una tercera VPN3 que se puede acceder únicamente conectado a la VPN1 / VPN2, es decir una segunda capa VPN como explica el punto 3.2.2. para aquellos servicios seguros y no disponibles desde la red principal del cloud.

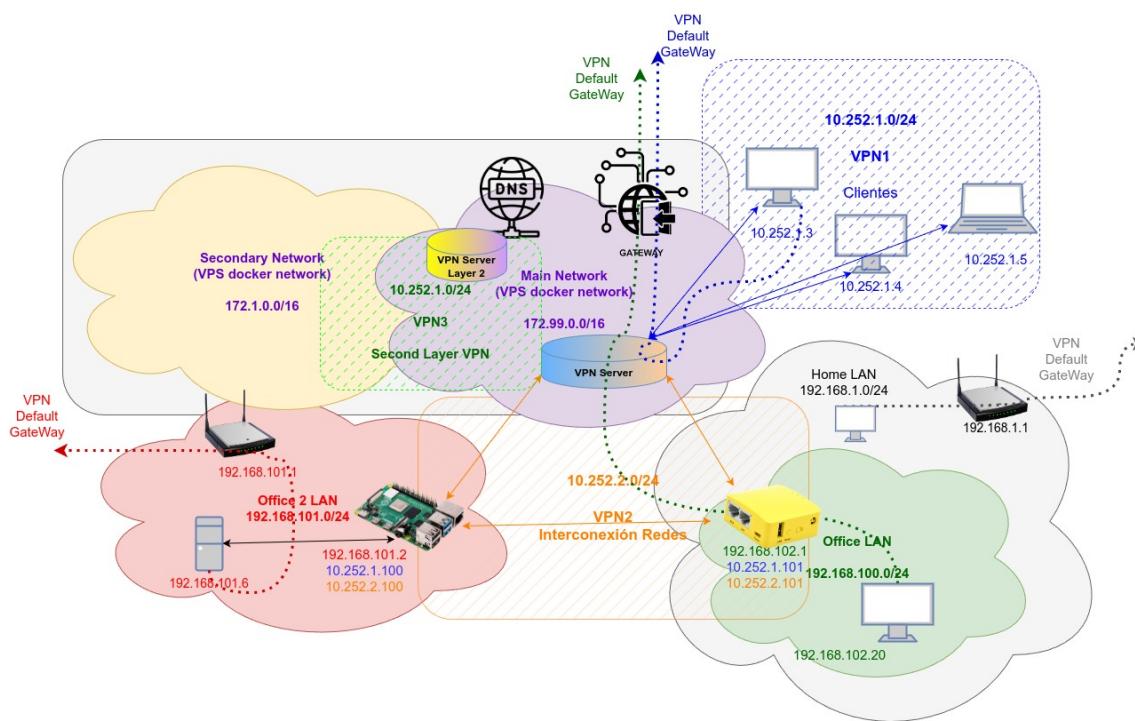


Figura 3.8: VPNs, relación entre ellas VPS y clientes.

En el diagrama de la figura 3.8 se muestra un ejemplo real donde tenemos 8 redes diferentes, 3 LAN, 3 VPN, y 2 redes docker:

- VPN1 10.252.1.0/24 (azul) , cuyos clientes entunelan tráfico en esta red virtual privada, gestionada por el VPN server. Implementa un NAT masquerade de azul-morado, provee de acceso a internet y conectividad con los servicios dockerizados de la red morada.
- VPN2 10.252.2.0/24 (naranja) , cuyos clientes entunelan tráfico en esta red virtual privada, gestionada por el VPN server. Provee conectividad de la red docker hacia las LAN de sedes y viceversa.
- Docker network 172.99.0.0/16 (morada), es una red privada de docker donde están la mayoría de servicios docker del VPS. Permite enrutar tráfico a través del VPS al exterior. Las sedes tiene conectividad con dichos servicios vía VPN2 así como el resto de clientes de VPN1. Es la red principal del cloud-VPS.

- Office 2 LAN, 192.168.101.0/24 (rojo) compuesta por una oficina con router ADS-L/fibra en la cual existe un dispositivo raspberry pi conectado a la VPN2. Permite acceder a la red morada (docker con servicios) y otras sedes (LAN verde oscuro). Ejemplo de casuística con default gateway por su propio router, no es por el VPS.
- Office LAN, 192.168.100.0/24 (verde oscuro) compuesta por una sede en casa con su propia LAN 192.168.1.0/24 adsl/fibra (gris) en la que se ha instalado un router wifi que habilita un NAT entre ambas e implementa un cliente vpn directo a la VPN2. Básicamente usa la red gris como infraestructura, se conecta vía VPN2 para enrutar todo el tráfico VPN-Red Docker hasta salir por el gateway del VPS.
- HomeLAN 192.168.1.0/24 (gris), LAN local de soporte a una red de oficina. Se puede observar como la red gris es independiente, no puede acceder a la red verde, ni ninguna otra red y su default gateway es su router, únicamente es usada como infraestructura.
- VPN Layer 2, VPN3 10.252.1.0/24 (verde claro), es una VPN únicamente accesible desde la red morada, es decir, se requiere de la VPN1 o estar en alguna LAN conectada a la VPN2 para poder acceder a ella. Similar a la VPN1-morada, realiza una conexión VPN3-amarilla, permitiendo acceder aquellos servicios dockerizados en la red docker interna amarilla y proveyendo de default gateway vía la red amarilla de docker.
- Second Docker net, 172.1.0.0/16 (amarilla) es otra red interna de docker la cual solo se puede acceder vía VPN3 que a su vez solo es accesible vía VPN1/VPN2. Es un caso de doble capa de VPN para acceder a los servicios.
- Elementos conectados a la VPN1, red azul, desde otros accesos, son terminales tunelados, es decir, su default gateway pasa a ser el VPS y usan el dns interno de la red morada docker. Tienen completa conectividad con los servicios de la red morada.

3.5. Generalización de Casos

Con el fin de explicar todas las posibilidades que brinda la nube generada en este documento, explicaremos los diferentes casos de uso general generar a través del diagrama de red de la figura 3.8 y de la figura 3.7 :

- Red verde oscura, es un ejemplo de despliegue por router portable, permite llegar a cualquier lugar y conectar por cable dicho router que genera un LAN autoconfigurada vía VPN2 a los servicios principales del VPS.
- Red roja, es un ejemplo de sede conectada al VPS. No modifica la funcionalidad de la red, pero añade rutas estáticas de enrutado a través de un elemento conectado a la VPN2 en este caso una raspberry pi.
- Red azul, es un ejemplo de VPN clásica, provee de acceso a internet vía VPS y permite la interconectividad a diferentes redes (intranet). Aunque los clientes pertenecientes a dicha red (VPN1) pueden verse entre si, pero no son accesibles directamente desde el VPS ni exponen sus propias LAN.

- Red naranja, en el diagrama [3.8](#) es usada como infraestructura de interconexión entre sedes. Sin embargo puede ser utilizada como una VPN parcial similar a VPN1 pero sin usar el VPS como default gateway o utilizando los dns internos como alternativos, únicamente se utiliza para acceder a unos servicios concretos, especialmente si están en otras sedes.
- Red morada, es una red virtual dentro del contexto del VPS. Es la **RED PRINCIPAL** de nuestro cloud, es decir la red troncal y en ella se sitúan los servicios dockerizados, está conectada a nivel bridge con el VPS por lo actúa como VPS-gateway y como proveedor de dns internos. Aunque la gran mayoría de servicios son internos, un mínimo de ellos es público (vpn server) o compartidos con otras redes docker.
- Red verde clara, es un claro ejemplo de VPN múltiple layer, genera una VPN3 que solo es accesible una vez conectado a la VPN1 o VPN2.
- Red amarilla, es un caso de red aislada de docker, no accesible desde el exterior, requiere de una o mas VPN para ser accesible.
- Red gris, ejemplo de red de infraestructura sin acceso o interacción con las redes de cloud-VPS.
- Red marrón, figura [3.7](#) ejemplo de red docker interna con proxy de redirección, es decir, puede contener ciertos servicios expuestos, pero su principal función es exponer clientes VPN a través de un Nat-reverse y un reverse-Proxy en la ip del VPS.
- Red azul clara, figura [3.7](#) ejemplo de red docker con servicios públicos con su principal característica es un reverse-Proxy y let's encrypt para generar HTTPS en los servicios expuestos. Es una red aislada de la red principal (morada), aunque varios servicios pueden estar en ambas redes como por ejemplo la front-ui de el servidor VPN.

CAPÍTULO 4. DESARROLLO DE SOFTWARE

Para el desarrollo de software de una manera adecuada, profesional e interdisciplinar (independiente de lenguaje, framework o grupo de trabajo) es necesario no solo unas herramientas en común, sino unas dinámicas de trabajo en equipo y especialmente una metodología de trabajo técnica, profesionalizada.

Usualmente se entiende que aunque un autónomo o freelance desarrollan un trabajo adecuado, en software el review o validación por terceros es un elemento crítico en la filosofía de verificación mejora y calidad del producto por ello asumimos que el entorno de trabajo cuenta siempre con un mínimo de 3-4 personas que forma un equipo técnico.

Por otra parte el equipo asume que tanto las tareas a realizar como la responsabilidad de las mismas no es personal sino grupal, por lo que todo el equipo debe ser involucrado en las tareas de gestión, desarrollo, testeo, documentación y puesta en marcha de la aplicación.

4.1. Dinámica de trabajo (equipo)

Actualmente por contexto laboral, las filosofía de trabajo AGILE[134], y en concreto SCRUM[135] son de amplia aceptación y exitosas. Por ello son las seleccionadas por este documento.

4.1.1. Agile

Agile es una metodología de trabajo cuya principal enfoque reside en la gestión de equipos y proyectos; define la manera de interaccionar personas, herramientas y procesos favoreciendo la colaboración con el cliente por encima de la negociación contractual. Su principal objetivo es el desarrollo de software funcional y la flexibilidad o respuesta al cambio a la hora de seguir una planificación.

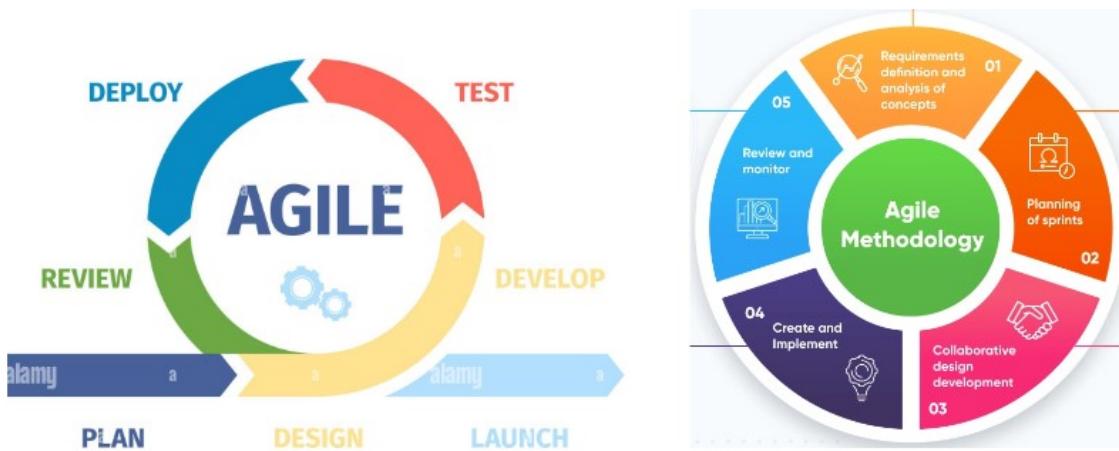


Figura 4.1: Diagrama metodología AGILE.

4.1.2. Scrum

Scrum es la implementación de una filosofía agile, destaca por su aproximación en fases y división del tiempo de trabajo en sprints (2-3 semanas de trabajo). El objetivo de scrum es la mejora continua, facilitando el valor añadido en un calendario de entregas, para ello define unas fases de planificación, evaluaciones diarias, re-definición, review y retrospectiva. Existen roles como el product owner que interacciona con los clientes para garantizar que se entienden los requisitos y se alcanzan los objetivos marcados para cada sprint. El equipo utiliza las “historias” del cliente-product owner, para definir tareas funcionales, estas deben ser especificadas y estimada por el equipo dentro de un backlog.

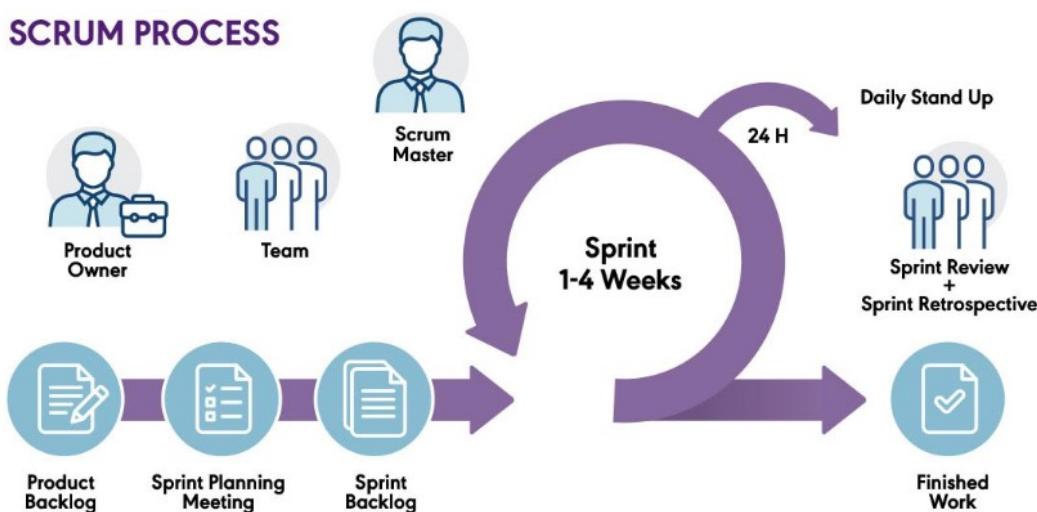


Figura 4.2: Diagrama funcionamiento SCRUM-AGILE[149].

El scrum master es el encargado de facilitar y supervisar el funcionamiento del scrum, dirigiendo las reuniones. El y el equipo planifican los tickets a realizar en un sprint, entre los estimados del backlog. Diariamente se comenta brevemente el trabajo realizado, problemas abordados del día anterior y la previsión diaria.

En caso de cambios no planificados se re-priorizan los tickets y se redefine el sprint restante, con el fin de obtener siempre una mayor satisfacción en la entrega. Una vez acabado el sprint, se realiza un proceso de review con el cliente con feedback real y de calidad sobre el resultado obtenido así como una retrospectiva del equipo sobre que se hizo bien, mal, causas, mejoras a implementar especialmente desde el punto de vista de gestión e información dentro del equipo.

Desde el puntos de vista técnico, cada cambio no solo está traceado por un ticket del sprint, sino que debe incluir las siguientes apartados:

- Documentación, descripción técnica de los cambios realizados por la tarea, que fundamenta la documentación general del software.
- Pull request, con el código de la tarea, auto-explicativo o apropiadamente documentado. Todo código debe respetar un código de buenas prácticas general o definido por el equipo. Test unitario o de integración que validan la funcionalidad añadida.

- Validación de un CI build con revisión aprobada, donde se compila ejecuta los test, se pasa validadores de calidad de código o vulnerabilidades como sonar[136] o blackduck[137]. Review cualitativo de otros programadores y review en profundidad por aquellos que conocen la naturaleza del servicio o los cambios (approved).

4.1.3. Generalización y profesionalidad

La importancia tanto de la metodología de equipo, como las buenas prácticas y estructuras de trabajo no solo aporta robustez, calidad y estandarización al trabajo entregado, sino que permite optimizar al equipo como cadena de trabajo.

Un caso destacado del trabajo estandarizado es la generalización de tareas, es decir, cualquier elemento del equipo puede asumir o traspasar tareas aunque no sean suyas, ya que el código debe ser fácilmente legible, testeado y documentado; fielmente protocolizado. Y este punto normalmente es inviable en equipos inferiores a 3-4 personas, donde no hay una dinámica de trabajo en equipo, sino división del trabajo y asignación de elementos.

Por ello la transferencia de tareas/código tienden a generar refactorizaciones o recreaciones de software ante la inviabilidad de mantenimiento o comprensión por ser un software no estandarizado o dicho de otra forma, creado por desarrolladores que no saben trabajar en equipo, o no profesionalizan su trabajo ya que no puede ser transferido.

4.2. Git, CI/CD y contenedor

Actualmente toda empresa de software debe de tener un control sobre los cambios del software y mecanismos automatizados para la evaluación del mismo. Pasar código por mail, usb o un almacenamiento en la nube no solo es un riesgo de seguridad sino que no es profesional y es el principio de un ciclo de malas praxis.

4.2.1. Git y repositorios de código

Git[138] es un programa de control de versión, significa que define versiones de código, permite generar nuevas versiones cuando se le generan cambios al código.

Permite calcular las diferencias entre versiones del código anterior y realizar “commits”, es decir, aglutinar un conjunto de cambios y almacenarlos como diferencias entre la versión anterior y nueva. Este commit, incluye un mensaje descriptivo de los cambios y en muchas ocasiones código o nombres del ticket asociado para trazabilizar el cambio.

El código se construye desde un commit inicial a la versión deseada, esta aplicación de las diferencias, permite la generación de “ramas” en base a aplicar unos cambios u otros.

El concepto de rama permite establecer versiones en paralelo del mismo software, así mismo permite unir ramas (merge) o crear nuevas a partir de las anteriores. El verdadero potencial de este software es un repositorio o servidor git en línea, el concepto es sincronizar nuestro git local con repositorios en remoto que también sincronizan con otros desarrolladores de software.

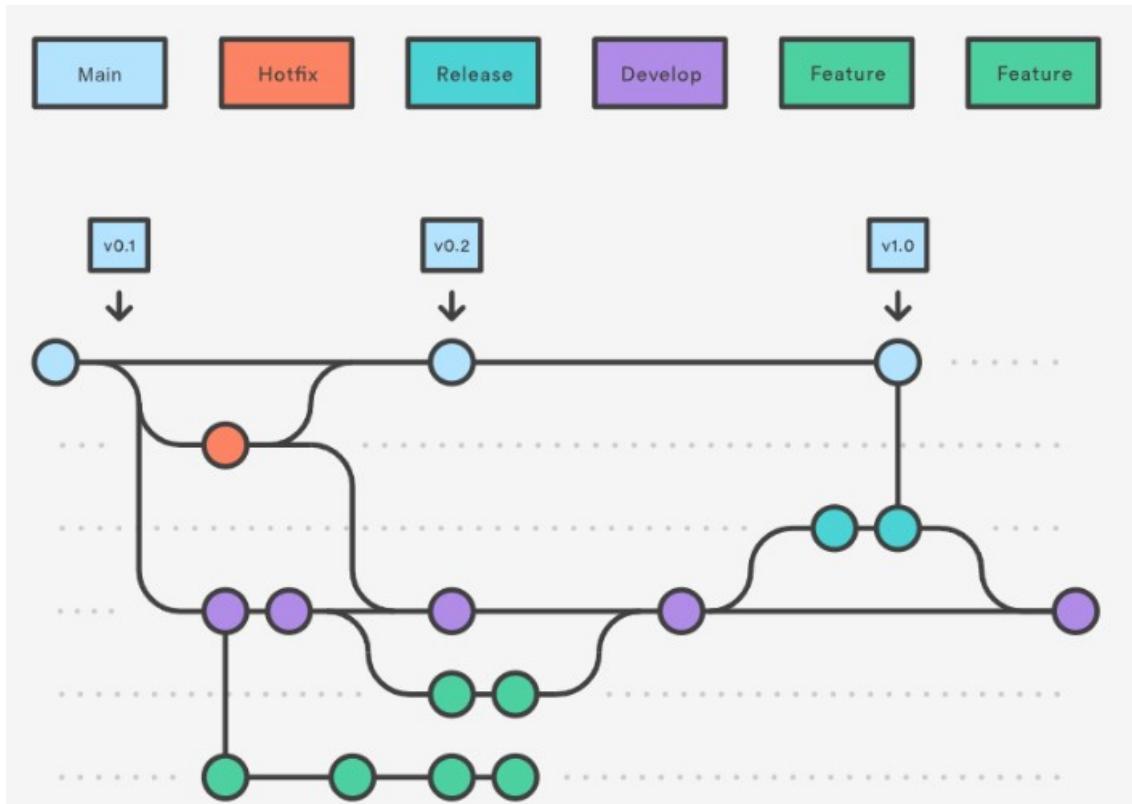


Figura 4.3: Diagrama flujo de trabajo en git[150].

Establecida una dinámica de trabajo (git flow figura 4.3), permite trabajar de manera paralela a programadores sobre el mismo código sin “pisarse” y sobre todo, tracear y poder volver a versiones anteriores del código, sean nuestras o de otros desarrolladores.

4.2.2. CI/CD

CI/CD se refiere a las siglas de Integración Continua / Entrega Continua en inglés, es un concepto que se basa en el uso del repositorio de git y un código apropiadamente testeado.

Cuando se termina una tarea, el desarrollador realiza un PR (Pull Request) que inicia una petición de revisión de los commits añadidos para la funcionalidad, esta petición acciona un programa agente, que compila y ejecuta los test. Si es satisfactorio almacena la nueva versión del código compilado en un repositorio no de código sino de binarios/paquetes (según sea el lenguaje utilizado) y espera el ‘ok’ de otros programadores en el PR. Esta acción de construcción y testeo continuada de versiones del software o integración (CI), puede ser acompañada de script automatizados que despliegan (CD) directamente el software construido en una plataforma de trabajo, que puede ser desarrollo, test o directamente producción, es decir, continuamente se entregan versiones nuevas del software.

Como último eslabón, tenemos la ‘contenerización’ que en nuestro caso de uso es dockerización, es decir, generar un contenedor docker con los binarios generados por el CI y guardar la imagen en un repositorio de contenedores, para que el CD únicamente tenga que actualizar la imagen docker vieja por una versión más nueva en nuestro orquestador

de despliegue.

4.2.3. Opinión de un usuario con experiencia

La combinación de un equipo agile con un entorno de git + CI/CD, más el uso de contenedores, permite el desarrollo en paralelo y revisión de tareas fácilmente, donde en cada cambio se valida el software y construye una nueva imagen, se despliega automáticamente sin riesgo humano. Todo ello con los beneficios intrínsecos mencionados que aporta docker a un software.

En mi opinión, este proceso completo únicamente se da en grandes empresas o startups tecnológicas con personal cualificado, pero en más del 50% de las pymes relacionadas con software o pequeños departamentos de software de grandes empresas, falla estrepitosamente, es decir, o no se aplica al completo o no se quieren aplicar tanto las metodologías de equipo o las automatizaciones y requisitos técnicos.

4.3. CI basado en docker

Históricamente se ha evolucionado desde un CI/CD de binarios, hacia un CI de binarios, que se construyen en un contenedor y el CD usa directamente el contenedor.

El caso más utilizado jenkins[69], es un agente multi-plugin que permite prácticamente la ejecución de cualquier lenguaje de programación, orquestador de dependencias y el uso de terceras aplicaciones. El problema reside en que el agente debe tener acceso a máquinas con el compilador, máquina virtual o dependencias necesarias para la compilación y ejecución de los test. Al igual que un servidor linux, requiere de un mantenimiento y complejidad a la hora de gestionar adecuadamente las versiones y compiladores; especialmente cuando una empresa cuenta con múltiples equipos donde cada uno con un producto que puede tener requisitos diferentes.

Una primera solución es el uso de docker como “aplicación”, es decir, utilizar los binarios de un container para ejecutar comandos específicos utilizando el directorio de trabajo como volumen. Así, una ejecución de maven/gcc/node/composer para generar los binarios/archivos necesarios y/o permite ejecutar los test. En dicho caso se reduce el mantenimiento, pero requiere de múltiples imágenes docker en el agente-ci.

En nuestro caso de interés, no disponemos de servidores-agentes sino que nuestra intención es que un container docker ejerza de agente-ci y por definición un container debe ser ligero. El punto novedoso, es el uso de container, con docker instalado, es decir, Docker in docker[32] y build ci[125] en dockerfile con múltiples[126] etapas. Véase anexo C.6.1. docker desde dentro del propio contenedor.

El segundo elemento es el uso de construcción de contenedores como build, es decir, compilar, ejecutar los test y construir la imagen en un único paso. Para ello se usa el concepto de múltiples etapas[126] que permite ejecutar pasos previos en imágenes base diferente que no son usadas para la imagen base final sino para la ejecución de comandos intermedios.

Finalmente obtenemos que con solo docker instalado y la apropiada configuración, aquellos requisitos técnicos serán descargados como imágenes auxiliares usadas, y generan

una imagen docker ligera únicamente con los ficheros necesario para la ejecución en producción.

4.4. CI/CD dentro de nuestra nube

Durante la realización de este documento se han realizado múltiples pruebas de concepto, entre ellas las prueba de un CI/CD basado en gitea[68]-drone[70]. Al evaluar lo junto a servicios públicos externalizados como github[65], bitbucket[66] o gitlab[64] entendemos que para un grupo reducido de desarrolladores es mas práctico y evita tareas de mantenimiento y segurización el uso de herramientas gratuitas externas. Sin embargo en aquellos casos donde el grupo de trabajo sea mas extenso o la naturaleza del código a generar tenga un carácter confidencial, se puede utilizar las pruebas conceptuales (véase anexo E.1.9.) como parte de nuestra nube privada.

4.5. Setup Software Local

En los entornos locales puede haber una mayor diversidad de SO, ya que no es un elemento crítico para el desarrollo. Sin embargo debe existir una automatización y especialmente una estandarización con el objetivo de minimizar problemas o la no reproducción de circunstancias a la hora de depurar o corregir un error detectado en producción, por lo tanto es de especial interés que las versiones y software esté alineado no solo en producción sino entre los propios programadores (herramientas auxiliares). En los siguientes apartados se detallan las estrategias principales utilizadas como software-local para cliente de nuestra nube o herramientas y configuraciones de desarrollo.

4.5.1. Automatización Local

Podemos usar Scripting, principalmente bash o batch, aunque se recomienda el uso de ansible apuntado a localhost. Podemos distinguir varias tareas:

- Instalación y configuraciones de elementos críticos de la compañía tales como cliente vpn, software de comunicaciones, correo o agentes de IT.
- Instalación de software necesario. Git, un gestor de repositorios git visual, compiladores, máquinas virtuales, IDE (Entornos de desarrollo), docker.
- Configuración de herramientas, gitignore, estandarización de IDE, puede ser definidas por la compañía, el grupo de trabajo o simplemente personalización individuales como los alias, guardadas en un repositorio de dot files.

4.5.2. Herramientas de desarrollo basadas en docker

Docker como compañía y empresa esta potenciando últimamente el segmento 'escritorio', especialmente en windows. Aunque docker CE (community edition) es open source, existe

una versión privada con tools, interfaz gráfica y extras, dicha versión tiene engine para servidores y engine utilizada habitualmente en escritorio windows.

De una manera muy similar a la "dockerización de servicios", su intención es la dockerización de todo tipo de aplicaciones, especialmente UI, permitiendo instalar y gestionar programas en escritorio. Este nuevo enfoque facilita especialmente la automatización de herramientas necesarias en nuestro Sistema Operativo de escritorio, ya que uno de los problemas mayores es olvidar o seguir fielmente pasos de una guía de herramientas, así como la gestión de herramientas con diferentes versiones dependiendo de la fecha de aplicación de la guía.

Por otra parte, si por ejemplo trabajamos con diferentes versiones o dependencias, facilita mucho la instalación en paralelo de las mismas ya que todo queda dentro de un container. Un ejemplo puede ser un Docker-compose que integra las diferentes versiones de Java / python/ php ..., el IDE de desarrollo, git y otras herramientas. Al igual que en el servidor únicamente con gestionar apropiadamente los volúmenes obtenemos un entorno funcional en segundos, con versiones estáticas y mas fáciles de actualizar.

4.5.3. Dot files y configuraciones portables

Recientemente se ha estandarizado una estrategia de archivos de configuración muy típica en el mundo linux/mac denominada “dot files”[139][140]. En los sistemas unix o derivados tiene archivos de configuración que normalmente usan el prefijo ‘.’ que a su vez indica un archivo oculto en el sistema.

La gran mayoría son configuraciones default o de respaldo, ya que muchos software de desarrollo no solo deben estar apropiadamente configurados sino que existe una tendencia a configurar customizaciones, especialmente para la realización de atajos, alias o la configuración de múltiples softwares interaccionando o selecciones de versión default en caso de varias instalaciones.

En muchos casos no solo se trata de un trabajo detallado y costoso sino tuneado a lo largo de meses de pruebas, ensayos y errores. Además dichos ficheros suelen residir en carpetas especiales del usuario, carpeta de instalación o en directorios del sistema, añadiendo no solo una compleja trazabilidad sino la dependencia del lugar de instalación personalizado.

La estrategia de “dot files” se basa en la premisa de centralizar todos estos ficheros junto a scripts, alias y claves de una manera centralizada. Para ello en vez de editar los archivos originales se crean enlaces simbólicos de la carpeta centralizada a la posición de los archivos. Aquellos ficheros como claves o configuraciones sensibles son apropiadamente cifrados y se genera un inventario de “software dot file”, con el fin de automatizar su instalación, creación de links o la conmutación de configuraciones basada en perfiles. Finalmente esta carpeta está bajo control de versiones apropiadamente sincronizado con un repositorio(véase ejemplo de codelitv[141]).

Como resultado se obtiene un control centralizado y sencillo de todas las configuraciones, un backup y restauración rápidos. Pero lo más importante un formateo-instalación o traspaso de configuración de SO rápido y eficiente, no estático, sino de manera continua ya que mediante git, se sincronizan diferente perfiles (ramas) de dot files en diferentes ordenadores, actualizando diariamente los cambios introducidos en otro pc.

CAPÍTULO 5. CONCLUSIONES

En la entrega de este documento (Octubre de 2023), se puede afirmar que la gran mayoría de los objetivos iniciales de este trabajo se han conseguido.

En primer lugar se ha planificado y ejecutado la creación de un lugar de teletrabajo adecuado para el autor, un trabajo continuo no exento de errores y mejoras realizadas durante el verano. Destacando especialmente en las comparativas con otros setup anteriores (anexo [B.4.4.](#)) y las soluciones aplicadas convergentes con otros autores en temáticas similares [\[9\]](#).

Segundo, se ha obtenido una nube virtual, económica, escalable fácilmente transferible que facilita y permite el teletrabajo. Especialmente aplicada a Elenkar S.L en el caso de mi pareja, pero de gran utilidad para proyectos personales o 'caseros'.

Por ultimo durante la investigación y recopilación se han comparado gran cantidad de servicios, tecnologías y proyectos basados en comunidades (véase tabla [C.4](#) así como pruebas de concepto [E.1.](#)), permitiendo seleccionar aquellos elementos mas útiles, ágiles y simples, que permiten ofrecer un producto customizable basado en un conglomerado de servicios gratuitos, LTS soportado por comunidades.

Desde el punto de vista técnico la conjunción de servicios gestionado por docker, docker-compose y ansible, ha permitido un ágil despliegue, backup y restauración, minimizando el mantenimiento (véase [C.5](#)) o el conocimiento necesario para utilizarlo, mientras requiere de un perfil bajo de recursos de gran utilidad en pequeñas y medianas empresas.

En definitiva un boceto de producto comercial al por menor, que ya es explotado minoritariamente como pack de servicios similares externalizados[\[118\]](#), productos especializados en casuísticas específicas como VPN[\[117\]](#), comunicación, web, almacenamientos(proveídos mismamente por VPS ovh[\[27\]](#)) o en autocracia (basada en raspberry pi) son por ejemplo Syncloud[\[119\]](#).

En conclusión, este trabajo no solo muestra el conocimiento o un resumen del estado actual entorno al teletrabajo y las herramientas necesarias para implementarlo. Sino que existe un verdadero mercado segmentado en la puesta en marcha de los servicios o la gestión directa de ellos. En ambos casos son pequeñas y mediana empresas, sin recursos donde normalmente uno de sus proveedores de red / material / hosting / vps / proveedor de software (web) han aceptado un rol de montaje, gestión y mantenimiento como una segunda fuente de ingresos pero especialmente como servicio **diferenciador** y complementario a sus clientes.

5.1. Conclusiones de la aplicación en Elenkar

Elenkar S.L es una pequeña empresa de 3-4 trabajadores enfocada en servicios inmobiliarios y servicios exclusivos relacionados, afincada en el baix penedes.

Sus principales necesidades son Web (captación de clientes), mail (método de comunicación vía internet), capacidad de almacenaje y compartición de documentos (dropbox / drive) con otras inmobiliarias/clientes.

Tras la aplicación de este documento se ha conseguido:

- Planificación real, y mejora de los recursos hardware, como software externalizados bajo el mismo presupuesto.
- Segurización y robusted real, tanto de red interna, interacción con clientes y vulnerabilidades de la gestión humana.
- Despliegue de recursos autócratas o confederales, a un coste low cost, que permiten reducir un presupuesto de 250€/anuales para un único servicio (dropbox), en un coste real de 60€/año por múltiples servicios incluyendo el almacenamiento en la nube.
- Acceso a servicios que han permitido la realización de trabajo en remoto parcial, VPN, acceso remoto, gestión centralizada de contraseñas seguras, control remoto de PC y compartición de recursos en remoto como impresoras, escáneres y servicios en red.
- El acceso a servicios no requeridos pero que mejoran el día a día como interacción de sensores y actuadores, gestión de alarmas o seguridad.

5.2. Conclusiones Personales

Desde el punto de vista personal puedo concluir que me ha permitido centralizar la casa de mis padres, casa de mi suegra, mi casa, otros inmuebles de una manera equivalente a las diversas sedes de una empresa. Y por similitud un soporte más directo y automatizado al no depender de terceras herramientas o de acciones humanas para configurar, arreglar elementos digitales.

Por otra parte como ayuda personal a proyecto de amigos/ex-compañeros, creo que les ha permitido reducir sus gastos iniciales así como reducir sus dependencias de softwares específicos, siendo una curva de aprendizaje más ligera para personas no técnicas, cuando se intenta realizar un proyecto personal de negocio.

5.3. Trabajo futuro

Aunque se ha obtenido el trabajo deseado y se han probado diversos conceptos de gran interés, se han detectado trabajos o conglomerados similares a este trabajo, especialmente focalizados en raspberry pi o en self hosting de servicios. Sin embargo todos ellos existen bajo un pretexto especializado o un conglomerado de servicios en pack, creo que lo que aun no existe en ningún producto privativo o opensource es el seleccionando de los casos de interés que autogenera la versión de docker-compose interesada o el despliegue directo de dicha versión sobre un VPS. Por lo tanto con posterioridad a este TFG, continuare con el proyecto en vías personales para la extensión y automatización del mismo incluyendo una UI que lo haga aun mas asequible sin conocimiento técnico.

BIBLIOGRAFÍA

- [1] López Cristina, Solana-González Pedro y Vanti Adolfo. "Industria 4.0: la transformación digital de las empresas". ([Link](#) ISBN 2199-8531. 05-2022): 15-35 . [iii](#), [v](#)
- [2] Jarosław Brodny y Magdalena Tutak. "Digitalization of Small and Medium-Sized Enterprises and Economic Growth: Evidence for the EU-27 Countries". *Journal of Open Innovation: Technology, Market, and Complexity, volume 8, number 2.* ([Link](#) ISBN 978-84-18167-15-7 2022). [iii](#), [v](#)
- [3] Andrew Fennell "Remote working statistics UK". *Standout CV.* ([Link](#) 2023). [iii](#), [v](#)
- [4] Equipos y talento Blog "Un 82% de las compañías ha implantado el teletrabajo". *Equipos y talento.* ([Link](#) 21-02-2023). [iii](#), [v](#)
- [5] Blszczyk Michal, Popovi, Milan, Zajdel, Karolina, y Zajdel Radoslaw "The impact of COVID-19 on remote work: A global natural experiment". *Sustainability Volume 14 Number 20.* ([Link](#) ISSN 2071-1050 2022). [iii](#), [v](#)
- [6] Katie Burke "Biggest Tech Companies Walk Back Remote-Work Policies". *CoStar News.* ([Link](#) 12-2022). [iii](#), [v](#)
- [7] Jefatura del Estado del Reino de España "Ley 10/2021, de 9 de julio, de trabajo a distancia.". *BOE* núm. 164, de 10 de julio de 2021. ([Link](#) BOE-A-2021-11472). [66](#)
- [8] Jon Messenger, Oscar Vargas Llave, Lutz Gschwind, Simon Boehmer, Greet Vermeylen y Mathijn Wilkens "Working anytime, anywhere: The effects on the world of work". *Eurofound* . ([Link](#) ISBN 978-92-897-1569-0 2017). [66](#)
- [9] Overemployed Blog community. "Overemployed, tools, setup, software and books.". ([Link](#) , 2000-23). [51](#), [70](#)
- [10] Santo Milasi and Ignacio González-Vázquez y Enrique Fernández-Macías "Telework before the COVID-19 pandemic". *OECD Productivity Working Papers.* ([Link](#) Number 21 2021). [67](#)
- [11] Jefatura del Estado del Reino de España "Real Decreto-ley 8/2020, de 17 de marzo de medidas urgentes extraordinarias para hacer frente al impacto económico y social del COVID-19". *BOE* núm. 73, de 18/03/2020.. ([Link](#) BOE-A-2020-3824). [67](#)
- [12] Manuel Fidalgo Vega, Instituto Nacional de Seguridad e higiene en el trabajo (INSST) "NTP 704: Síndrome de estar quemado por el trabajo o "burnout"(I): definición y proceso de generación". *CENTRO NACIONAL DE CONDICIONES DE TRABAJO..* ([Link](#) NTP 704, 2003). [69](#)
- [13] Angira Sharma, Edward Kosasih, Jie Zhang, Alexandra Brintrup y Anisoara Calinescu "Digital Twins: State of the art theory and practice, challenges, and open research questions". *Journal of Industrial Information Integration, Volume 30..* ([Link](#) ISSN 2452-414X, 2022). [2](#)

- [14] Rahman Chowdury, Uddin Syed, Karim, M. y Ahmed Mohiuddin “Evaluation of work postures - The associated risk analysis and the impact on labor productivity”. *Journal of Engineering and Applied Sciences, Volume 10 2542 - 2550..* ([Link](#) , 04-2015). **5**
- [15] Akulwar-Tajane Isha, Darvesh Musfira, Ghule Maithili, Deokul Spandita, Deora Bhavana y Mhatre Vedika “Effects of COVID -19 Pandemic Lock Down on Posture in Physiotherapy Students: A Cross Sectional Study”. ([Link](#) , 01-2021). **5**
- [16] Sampedro Casis, Rodrigo “Selección de hardware, guia y contexto histórico.”. ([Link](#) , 2023). **5, 6, 15, 76**
- [17] W3Schools “Browser Display Statistics.”. ([Link](#) , 2000-23). **76**
- [18] Iucera “PIA electricidad: Qué es y para qué sirve”. ([Link](#) , 2023). **79**
- [19] Wikipedia, La enciclopedia libre. “Worldwide Interoperability for Microwave Access.”. ([Link](#) , 2023). **82, 84**
- [20] Wikipedia, La enciclopedia libre. “Starlink.”. ([Link](#) , 2023). **82**
- [21] Wikipedia, La enciclopedia libre. “Digital subscriber line: DSL.”. ([Link](#), 2023). **82**
- [22] Wikipedia, La enciclopedia libre. “Power Line Communications.”. ([Link](#), 2023). **83, 86**
- [23] Alicia Zambrano Braun “Inhouse vs. Outsourcing – ¿Cuál elegir y cuándo?”. ([Link](#), 2023). **101**
- [24] Jefatura del Estado del Reino de España “Orden ECE/1166/2018, de 29 de octubre.”. *BOE* núm. 270, de 8 de noviembre de 2018. ([Link](#) BOE-A-2018-15341). **84**
- [25] Phil Odence “Five types of software licenses you need to understand”.*Synopsys blog.* ([Link](#), 2023). **xvi, 102**
- [26] Wikipedia, La enciclopedia libre. “Carrier Grade NAT”. ([Link](#), 2023). **32, 103, 134**
- [27] Web page OVH SAS. “VPS OVH cloud euro prices.”. ([Link](#), 2023). **17, 51, 104**
- [28] Web page Time4 VPS. “VPS Time4 VPS euro prices.”. ([Link](#), 2023). **17, 104**
- [29] Ansible project web page. “Ansible simplest automate aps and IT infraestructure software.”. ([Link](#), 2023). **18**
- [30] Docker Community. “Docker Network documentation.”. ([Link](#), 2023). **105**
- [31] Docker web page. “Docker entorno en tiempo de ejecucion para la creacion y gestion de contenedores.”. ([Link](#), 2023). **18, 20**
- [32] Docker community & Tianon Docker project. “Docker in Docker.”. ([Link](#), 2023). **47, 120**
- [33] Luc Juggery. “Docker Tips : about /var/run/docker.sock”. ([Link](#), 2023). **36, 37, 120, 121**
- [34] Ngix-proxy community. “Generate files from docker container meta-data”. ([Link](#), 2023). **36, 121**

- [35] Nestybox company and community. “An open-source, next-generation runc”that empowers rootless containers to run workloads such as Systemd, Docker, Kubernetes, just like VMs.”. ([Link](#), 2023). [xvi](#), [120](#), [122](#), [123](#)
- [36] Docker Compose documentación, web page. “Docker Compose herramienta de gestión multi-container.”. ([Linksource](#), 2023). [21](#)
- [37] Docker Swarm documentación web page. “Docker Swarm herramienta de gestión multi-container clusterizado.”. ([Link](#), 2023). [22](#)
- [38] Kubernetes web page. “kubernetes plataforma opensource de cluster de contenedores”. ([Link](#), 2023). [18](#), [22](#)
- [39] Wikipedia, La enciclopedia libre. “Single sign-on”. ([Link](#), 2023). [105](#)
- [40] Soluciones Corporativas IP, SL. “Don Dominio Servicios de correo.”. ([Link](#), 2023). [25](#), [108](#)
- [41] awesome-selfhosted “Mail selft hosted service collection tools.”. ([Link](#), 2023). [108](#)
- [42] Docker Mailserver Organization “Docker Mailserver Documentation.”. ([Link](#), 2023). [108](#), [117](#)
- [43] Slack Technologies, LLC, una empresa de Salesforce “Slack plataforma de comunicación.”. ([Link](#), 2023). [108](#), [109](#)
- [44] Microsoft “Teams Microsoft plataform.”. ([Link](#), 2023). [108](#)
- [45] Telegram company. “Telegram web page.”. ([Link](#), 2023). [109](#)
- [46] Franz company & community. “One service unlimited accounts”. ([Link](#), 2023). [109](#), [115](#), [119](#), [171](#)
- [47] Ferdi community. “Opensource fork of Franz.”. ([Link](#), 2023). [109](#), [119](#)
- [48] Station community. “One app to rule them all.”. ([Link](#), 2023). [109](#), [119](#)
- [49] Telefonica Ecuador. “Intranet ventajas”. ([Link](#), 2023). [xvi](#), [110](#)
- [50] Sangoma Technologies. “Asterisk powers IP PBX systems, VoIP gateways, conference servers and other custom solutions”. ([Link](#), 2023). [110](#)
- [51] Samba Team Members “Since 1992, Samba has provided secure, stable and fast file and print services for all clients using the SMB/CIFS protocol, such as all versions of DOS and Windows, OS/2, Linux and many others.”. ([Link](#), 2023). [110](#), [117](#)
- [52] Dropbox Company. “Dropbox mantén todo al alcance de tu mano.”. ([Link](#), 2023). [110](#), [115](#)
- [53] Google LLC “Google Drive es un servicio de alojamiento y sincronización de archivos desarrollado por Google.”. ([Link](#), 2023). [110](#)
- [54] MEga The Privacy Company. “Almacena archivos, chatea y reúnete, todo en un solo lugar.”. ([Link](#), 2023). [110](#)

- [55] Seafile community team. “Your Data Safety is our First Priority.”. ([Link](#), 2023). [110](#), [117](#)
- [56] ownCloud GmbH. “ownCloud, your file platform.”. ([Link](#), 2023). [110](#), [116](#), [117](#)
- [57] Nextcloud GmbH. “ Files, Talk, Groupware y Office en una única plataforma, optimizando el flujo de colaboración. ”. ([Link](#), 2023). [25](#), [110](#), [115](#), [117](#)
- [58] Collabora Ltd “Collabora Online es un paquete de oficina en línea de código abierto desarrollado por Collabora Productivity, una división de Collabora.”. ([Link](#), 2023). [25](#), [111](#)
- [59] MediaWiki community & Docker Community “MediaWiki is free and open-source wiki software.”. ([Link](#), 2023). [25](#), [111](#), [117](#), [149](#)
- [60] BookStack registered trade mark of Daniel Brown. “BookStack is a simple, self-hosted, easy-to-use platform for organising and storing information. ”. ([Link](#), 2023). [25](#), [111](#), [117](#), [149](#)
- [61] Alexander, xy2z (user). “A fast and lightweight site for viewing files.”. ([Link](#), 2023). [111](#), [117](#), [149](#)
- [62] Gitea Community docs. “Gitea General Features and Comparison”. ([Link](#), 2023). [112](#)
- [63] Git web page. “Git control de versiones distribuido.”. ([Link](#), 2023). [18](#), [45](#)
- [64] Gitlab Inc. “Un servicio web de forja, control de versiones y DevOps basado en Git.”. ([Link](#), 2023). [48](#), [111](#), [116](#), [118](#)
- [65] Microsoft. “GitHub is where over 100 million developers shape the future of software, together. ”. ([Link](#), 2023). [48](#), [116](#)
- [66] Atlassian “Bitbucket es un servicio de alojamiento basado en web, para los proyectos que utilizan el sistema de control de versiones Mercurial y Git”. ([Link](#), 2023). [48](#), [111](#), [116](#), [118](#)
- [67] The Gogs project “Gogs is a painless self-hosted Git service.’. ([Link](#), 2023). [111](#), [118](#)
- [68] Gitea Ltd. “Gitea es un paquete de software de código abierto para alojar el control de versiones de desarrollo de software utilizando Gi.’. ([Link](#), 2023). [25](#), [48](#), [111](#), [118](#), [160](#)
- [69] Jenkins Project. ‘Jenkins es un servidor de automatización open source escrito en Java.’. ([Link](#), 2023). [25](#), [47](#), [112](#), [118](#)
- [70] Harness Inc. ‘Drone is a self-service Continuous Integration platform for busy development teams.’. ([Link](#), 2023). [25](#), [48](#), [112](#), [118](#)
- [71] Wnpower New y blog. ‘Market share y estadísticas de WordPress actuales’. ([Link](#), Noviembre 2022). [112](#)
- [72] Wordpress company. ‘Build and grow your website with the best way to WordPress.’. ([Link](#), 2023). [25](#), [112](#), [116](#), [118](#)

- [73] PrestaShop company. 'Vende online en una plataforma e-commerce totalmente personalizable que se adapta al crecimiento de tu negocio.'. ([Link](#), 2023). [112](#), [116](#), [118](#)
- [74] BEES global company. 'BEES está transformando el modelo tradicional de ventas, ubicando a los clientes en el centro de todo, haciendo sus vidas más simples y sus negocios más rentables.'. ([Link](#), 2023). [112](#), [118](#)
- [75] Hugo open source project. 'Hugo is one of the most popular open-source static site generators.'. ([Link](#), 2023). [25](#), [112](#), [118](#)
- [76] KeePass project. 'KeeWeb is a webclient for KeePass software.'. ([Link](#), 2023). [113](#), [119](#)
- [77] Disbug, Aswin Kumar KP. 'Github vs Gitlab vs Bitbucket.'. ([Link](#), 2023). [116](#)
- [78] Rocket.Chat company. 'Deja que la conversación fluya.'. ([Link](#), 2023). [24](#), [109](#), [117](#), [147](#)
- [79] Tinode LLC. 'Instant messaging server.'. ([Link](#), 2023). [117](#)
- [80] Tinode LLC. 'Organized team chat, the calmer, more efficient way to work'. ([Link](#), 2023). [117](#)
- [81] New Vector Ltd. 'A secure communications platform built around you.'. ([Link](#), 2023). [117](#)
- [82] Wire Swiss GmbH 'Protect the information that you have been entrusted with.'. ([Link](#), 2023). [117](#)
- [83] Mattermost, Inc 'Accelerating mission critical work in complex operational environments.'. ([Link](#), 2023). [117](#), [118](#)
- [84] Mailu authors community. 'Mailu is a simple yet full-featured mail server as a set of Docker images.'. ([Link](#), 2023). [117](#)
- [85] IETF RFC 4510, RFC 4511. 'The Lightweight Directory Access Protocol.'. ([Link](#), [Link RFC](#), 2023). [24](#), [105](#), [114](#), [117](#), [156](#)
- [86] Red Hat projects community. 'Keycloak is an open source identity and access management solution.'. ([Link](#), 2023). [24](#), [114](#), [117](#), [156](#)
- [87] OpenVPN, Inc.. 'OpenVPN provides VPN server solutions for small to mid-size businesses.'. ([Link](#), 2023). [117](#), [131](#)
- [88] Jason A. Donenfeld. The zx2c4 project. 'WireGuard securely encapsulates IP packets over UDP.'. ([Link](#), 2023). [25](#), [118](#), [131](#)
- [89] IETF RFC 6071 'IPsec es un conjunto de protocolos cuya función es asegurar las comunicaciones sobre el Protocolo de Internet autenticando y/o cifrando cada paquete IP en un flujo de datos.'. ([Link](#), [Link RFC](#), 2023). [118](#), [131](#)
- [90] Pritunl, Inc 'Enterprise Distributed OpenVPN, IPsec and WireGuard Server.'. ([Link](#), 2023). [118](#)

- [91] NetMaker Inc ‘Unify your devices with an ultra-powerful overlay network.’. ([Link](#), 2023). [131](#)
- [92] Taiga Inc ‘A featured-rich software that offers a very simple start through its intuitive user interface..’ ([Link](#), 2023). [25](#), [118](#)
- [93] Planka, opensource team. ‘Free open source kanban board for workgroups.’. ([Link](#), 2023), [118](#), [155](#)
- [94] Passbolt S.A. ‘The open source password manager for teams.’. ([Link](#), 2023). [119](#)
- [95] Portainer company. ‘Portainer is the most versatile container management platform that simplifies your secure adoption of containers with remarkable speed.’. ([Link](#), 2023). [119](#)
- [96] Khanh Ngo, ngoduykhanh(user). ‘A web user interface to manage your WireGuard setup.’. ([Link](#), 2023). [119](#)
- [97] Wg-easy community. ‘The easiest way to run WireGuard VPN + Web-based Admin UI.’. ([Link](#), 2023). [119](#)
- [98] Portainer company. ‘AdGuard es la mejor manera de deshacerse de los anuncios molestos, rastreadores en línea y proteger tu computadora del malware.’. ([Link](#), 2023). [119](#)
- [99] Elvis Souza, mageddo (user). ‘Solve your DNS hosts from your docker containers, then from your local configuration, then from internet.’. ([Link](#), 2023). [118](#)
- [100] Carl Sverre, carlsverre (user). ‘Created to solve a single problem well. Run this container in a docker-compose v2 file and it will proxy dns requests to the docker daemon’s embedded dns server. Technically this will work for any docker user network, but its designed to be used with docker-compose.’. ([Link](#), 2023). [118](#)
- [101] Nicco, cupcakearmy (user). ‘Autorestic is a wrapper around the amazing restic.’. ([Link](#), 2023). [27](#)
- [102] Cuplicati community. ‘Free backup software to store encrypted backups online.’. ([Link](#), 2023). [27](#)
- [103] Ascensio System SIA . ‘Run your private office with the ONLYOFFICE’. ([Link](#), 2023). [111](#)
- [104] Linux command. ‘Rsync is a fast and extraordinarily versatile file copying tool.’. ([Link](#), 2023). [27](#)
- [105] Ivan Franchin. ‘Proof-of-Concept springboot-react-keycloak with openidap.’. ([Link](#), 2023). [xvi](#), [114](#)
- [106] Deon George. ‘Web based LDAP administration tool.’. ([Link](#), 2023). [114](#)
- [107] Freedesktop Organization. ‘systemd System and Service Manager’. ([Link](#), 2023). [27](#), [123](#), [125](#)

- [108] Command from systemD. ‘The journalctl command is part of the systemd suite of utilities and is used to query and display log messages from the systemd journal’. ([Link](#), 2023). [125](#)
- [109] Eduardo Zepeda. “Coffee bytes Blog: Artículos sobre desarrollo web y Linux.”. ([Link](#), 2023). [20](#)
- [110] Rubén Aguilera Díaz-Heredero *Adictos al trabajo, autentia.* “Ejecutar cualquier docker-compose como servicio.”. ([Link](#), 2018). [26](#)
- [111] Mehdi Hosseini,*Medium blog*. “How to Link Multiple Docker Compose Files.”. ([Link](#), 2022). [26](#)
- [112] Philipp Scheit,*Medium blog*. “docker-compose advanced configuration.”. ([Link](#), 2018). [26](#)
- [113] Jefatura del Estado del Reino de España “Ley 2/2023, de 20 de febrero, reguladora de la protección de las personas que informen sobre infracciones normativas y de lucha contra la corrupción.”. *BOE* núm. 44, de 21 de Febrero de 2023. ([Link](#) BOE-A-2023-4513). [28](#)
- [114] Comisión Europea. “El Reglamento general de protección de datos (RGPD), la Directiva sobre protección de datos en el ámbito penal y otras normas relativas a la protección de datos personales.”. *La protección de datos de la UE RGPD*. ([Link](#) BOE-A-2023-4513). [28, 104](#)
- [115] Agencia Española de protección de datos. “Derechos, Deberes y área de actuación.”. ([Link](#) 2023). [29](#)
- [116] Pi-hole, LLC. “Pi-hole es una aplicación para bloqueo de anuncios y rastreadores en Internet.”. ([Link](#) 2023). [31, 34, 135](#)
- [117] Arcem Tene INC, Seattle, USA “Pro Custodibus makes WireGuard networks easy to manage.”. ([Link](#) 2023). [51](#)
- [118] E-Internet “Productos VPS, hosting, y shelf host de terceros productos(wordpress, phrestashop, wiki, nextcloud, vpn ...).” . ([Link](#) 2023). [51](#)
- [119] E-Internet “Syncloud device runs your apps at your premises.”. ([Link](#) 2023). [51](#)
- [120] ISO/IEC 7498. ‘The Open Systems Interconnection model.’. ([Link](#), [Link](#) ISO/IEC 7498-1, 2023). [xvi, 31, 129](#)
- [121] WWW protocol ‘Hypertext Transfer Protocol Secure.’. ([Link](#) 2023). [135](#)
- [122] Internet Security Research Group ‘Let’s Encrypt es una Autoridad de Certificación gratuita, automatizada, y abierta.’. ([Link](#) 2023). [36, 118, 135](#)
- [123] Traefik Labs ‘The world’s most popular cloud-native application proxy that helps developers and operations teams build, deploy and run modern microservices applications quickly and easily.’. ([Link](#) 2023). [36, 118, 154](#)
- [124] F5, Inc ‘The open source web server that powers more than 400 million websites.’. ([Link](#) 2023). [36, 118, 154](#)

- [125] Docker Documentation ‘Continuous integration with Docker.’. ([Link](#) 2023). **47**
- [126] Docker Documentation ‘Multi-stage builds.’. ([Link](#) 2023). **47**
- [127] Docker Documentation ‘Compose file version 2 references.’. ([Link](#) 2023). **119**
- [128] Docker Documentation ‘Compose file version 2 references.’. ([Link](#) 2023). **119**
- [129] Docker Documentation ‘Compose include.’. ([Link](#) 2023). **128**
- [130] Docker Documentation ‘Compose Merge.’. ([Link](#) 2023). **128**
- [131] Borja Paz Rodríguez ‘Managed home server with Docker, Docker Compose, Make and Bash’. ([Link](#) 2023). **128**
- [132] WeTransfer ‘WeTransfer is the simplest way to send your files around the world’. ([Link](#) 2023). **171**
- [133] EmbeDD GmbH ‘DD-WRT is a Linux based alternative OpenSource firmware suitable for a great variety of WLAN routers and embedded systems.’. ([Link](#) 2023). **141**
- [134] Wikipedia, La enciclopedia libre. ‘Agile software development.’. ([Link](#) 2023). **43**
- [135] Wikipedia, La enciclopedia libre. ‘Scrum is an agile project management system commonly used in software development and other industries.’. ([Link](#) 2023). **43**
- [136] Wikipedia, La enciclopedia libre. ‘SonarQube es una plataforma para evaluar código fuente.’. ([Link](#) 2023). **45**
- [137] Synopsys ‘Black Duck software composition analysis (SCA) helps teams manage the security, quality, and license compliance risks that come from the use of open source and third-party code in applications and containers’. ([Link](#) 2023). **45**
- [138] Software Freedom Conservancy. ‘BGit is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency’. ([Link](#) 2023). **18, 45**
- [139] Joel Glovier. ‘Your unofficial guide to dotfiles on GitHub’. ([Link](#) 2023). **49**
- [140] Lars Kappert. ‘A curated list of dotfiles resources. Inspired by the awesome list thing..’ ([Link](#) 2023). **49**
- [141] CodelyTV. ‘Repository containing all the automation required to setup your MacOS in just a few seconds after a fresh install..’ ([Link](#) 2023). **49**
- [142] Ascensio System SIA. ‘Dirige tu oficina privada con ONLYOFFICE una suite ofimática en línea segura y compatible con los formatos de MS Office.’. ([Link](#) 2023). **111**
- [143] Steve Strutt, IBM Cloud Schematics Service. ‘Image Source Ansible diagram.’. ([Link](#) Septiembre 2023). **xv, 18**
- [144] Omar Aqlan. ‘Image Source Ansible diagram.’. ([Link](#) Septiembre 2023). **xv, 18**
- [145] Karthikeyan Shanmugam ‘Image Source Docker vs Virtual Machine.’. ([Link](#) Septiembre 2023). **xv, 20**

- [146] Atharv Yeole. ‘Image Source Docker-compose, swarm y kubernetes.’. ([Link](#) Septiembre 2023). [xv](#), [21](#)
- [147] Wikipedia, La enciclopedia libre. ‘Image Source Redes interconexion de sedes.’. ([Link](#) Septiembre 2023). [xv](#), [33](#)
- [148] defreitas/dns-proxy-server. ‘Image Source Redes interconexión de sedes.’. ([Link](#) Septiembre 2023). [xv](#), [xvi](#), [34](#), [134](#)
- [149] PM Partners ‘Image Source Scrum.’. ([Link](#) Septiembre 2023). [xv](#), [44](#)
- [150] Atlassian ‘Image Source Git flow.’. ([Link](#) Septiembre 2023). [xv](#), [46](#)
- [151] Redacción emprendedores ‘Image Source Coworking.’. ([Link](#) Septiembre 2023). [xv](#), [65](#)
- [152] Isabel Tolosa L. ‘Image Source Teletrabajo.’. ([Link](#) Septiembre 2023). [xv](#), [65](#)
- [153] Intel. ‘Image Source pantallas formas y dimensiones.’. ([Link](#) Septiembre 2023). [xv](#), [75](#)
- [154] M. Ali, hostnac. ‘Image Source servidores tipos.’. ([Link](#) Septiembre 2023). [xvi](#), [103](#)
- [155] Security Lit Limited. ‘Image Source SSO.’. ([Link](#) Septiembre 2023). [xvi](#), [113](#)
- [156] Dhirendra Patil. ‘Image Source Docker in Docker and Docker out of Docker.’. ([Link](#) Septiembre 2023). [xvi](#), [121](#), [122](#)
- [157] Abu Sayeed. ‘Image Source DNS.’. ([Link](#) Septiembre 2023). [xv](#), [xvi](#), [34](#), [134](#)
- [158] Steeve, dev.to . ‘Image Source Ngix proxy.’. ([Link](#) Septiembre 2023). [xvi](#), [136](#)
- [159] Made-in-china Image products. ‘Image Source UTP cables.’. ([Link](#) Septiembre 2023). [xvi](#), [139](#)
- [160] Rodrigo sampedro Casis github project “TFG documentation and code.”. ([Link](#), 2023). [24](#), [169](#)

APÉNDICES

APÉNDICE A. TELETRABAJO

Este anexo resume los conceptos básicos, estado del arte actual tanto desde el punto de vista legal, funcional o práctico relacionado con el “teletrabajo”. El principal objetivo es diferenciar nuestro caso de estudio del resto de casuísticas.

A.1. Contexto Semántico

¿Qué es el teletrabajo o home office? ¿Es lo mismo que el trabajo en remoto?

Comencemos definiendo el significado de las palabras, la interacción entre ellas y el uso técnico que entendemos al usarlas. Teletrabajo, significa trabajar a distancia (“tele-” prefijo griego lejos), “home office” es el término en inglés para indicar trabajar desde casa y “trabajar en remoto” indica que el trabajador realiza su tarea fuera de la oficina principal, es decir, de manera virtual.



Figura A.1: Coworking[151] vs home office[152]

Aunque estas palabras son similares su uso técnico tiene diferencias significativas e implicaciones legales diferentes. Hemos de separar conceptualmente 3 visiones diferentes y complementarias, la organizativa a la hora de realizar el trabajo (grupo de trabajo), la situación física del trabajador y el carácter legal del trabajador.

Existen empresas que no tienen una sede física con oficina o tienen múltiples, las hay que mandan a sus trabajadores a las oficinas de sus clientes. En cualquiera de estos casos, estos trabajadores realizan el 100% de su trabajo en remoto, puesto el grupo de trabajo no está reunido completamente en la misma oficina física, pero ninguno lo realiza desde su casa o consta como teletrabajo legalmente, puesto que se desplazan a la oficina que su compañía les indica.

Por otra parte, también existen muchas compañías que ofrecen flexibilidad horaria y la posibilidad de ‘teletrabajar’ dentro de su pack de conciliación. Que el trabajador pueda salir antes de la oficina y terminar un porcentaje de jornada desde casa (o cualquier otro lugar) con el fin facilitar su conciliación trabajo-vida familiar.

Por último tenemos el caso de muchos trabajadores, especialmente mandos intermedios o especialistas cuyos proyectos con picos de trabajo realizan horas extras “trabajando desde casa”.

Desde el punto de vista técnico, **trabajar en remoto**, se refiere única y exclusivamente a realizar el trabajo (función dentro del grupo de trabajo) siendo exclusivamente dependiente de **medios virtuales**, es decir, el grupo de trabajo no está físicamente en contacto directo, no considera el lugar desde donde se realiza la actividad. Trabajar desde casa, o '**home office**', indica que el trabajador dispone y realiza **un porcentaje significativo** (y mayoritario) de su trabajo en remoto, desde su vivienda habitual. Por último, **teletrabajar-legalmente** en España indica que el trabajador realiza más de un 30% de su jornada laboral fuera de la oficina física de la empresa (ejemplo más 1.5 días a la semana o más de 2 horas 20 min diarias), o hasta un 50% en contratos formativos.

Legalmente, el teletrabajo es voluntario, por lo tanto es un acuerdo fomentado por ambas partes. La empresa está obligada [7] a firmar un contrato individual con el trabajador, donde se detallan los pormenores de la cesión de equipamiento y la compensación para los gastos ocasionados por el teletrabajo, así como los plazos, renovaciones y la posibilidad del cambio de dichas condiciones. Habitualmente suele quedar negociado dentro del convenio sindical, indicando que equipos o coworking designados y la cuantía mensual extra asociada a los gastos generados o con un presupuesto máximo preestablecido para nuevas incorporaciones.

Esto en la gran mayoría de casos suele plasmarse con "un acuerdo individual" basado en el convenio colectivo, usualmente con un portátil más todo aquel material que permiten llevar de la oficina a tu casa o una lista de elementos a comprar que la empresa te enviará a casa.

A.2. Objetivo del teletrabajo

Múltiples empresas han acabado implementado el teletrabajo, cada una por motivos diferentes[8], entre los principales objetivos y beneficios son los siguientes:

- Productividad, principalmente relacionada con la mayor concentración de sus trabajadores en ambientes sin ruido ni interrupciones, junto a un uso más eficaz de su jornada al reducir el estrés de desplazarse a las oficinas y reducción de tiempos perdidos o incidentes (atascos, meteorología, aglomeraciones ...).
- Flexibilidad extendida, principalmente focalizada en la conciliación familiar. "Un trabajador feliz" no solo es más productivo, es retener talento.
- Trabajadores des localizados, permite adquirir una plantilla no geo-localizada en una región local. Permite captar más talento, ya que el lugar de residencia no es requisito indispensable. Permite equilibrar salarial mente, aquellos lugares con rentas especialmente altas asociadas a grandes metrópolis.
- Reducción de costes asociados a la infraestructura física de las oficinas o servicios auxiliares de las mismas. Normalmente la reducción de costes en oficina física es superior a los acuerdos compensatorios de teletrabajo.
- Causa mayor, en algunas ocasiones eventos especiales, circunstancias climáticas- locales, situaciones político-sociales o causas médicas, no permiten la realización de la actividad laboral, siendo el teletrabajo la única opción.

Por otra parte requiere de una gestión o problemática asociada a la no interacción física, así como el **no uso** del “lenguaje no verbal” o comportamientos de grupos, difíciles de crear **sin un trabajo previo no remoto**. Por lo tanto existe las siguientes problemáticas:

1. Mala gestión humana, recae en fuga de talento o bajada de productividad.
2. Bajada generalizada de la creatividad y cohesión en equipos sin encuentros físicos.
3. Requiere de una gestión jerárquica más directa y plana, así como una supervisión facilitada por ambos lados.

A.3. Histórico

El trabajo en remoto existe hace décadas, especialmente desde la llegada del teléfono y los tele-operadores, pero para nuestra actual comprensión teletrabajar, está íntimamente ligado a trabajar con elementos tecnológicos que desde los años 90 permiten conectarse entre ellos y trabajar directamente con herramientas virtuales en constante comunicación.

Sin embargo ha sido una excepción de una élite muy especializada hasta la llegada de la banda ancha generalizada. No solo a sectores concretos sino también delimitado a regiones densamente pobladas y desarrolladas donde la conexión cumplía con los requisitos adecuados.

A.3.1. Pre pandemia

Desde 2015 se observa un crecimiento significativo hasta el 5-15%[\[10\]](#) en países desarrollados y la aparición de teletrabajo-des localizado especialmente en externalizaciones y freelance en países más competitivos salarial mente. Ambos casos se focalizan en una necesidad empresarial de obtener empleados cualificados, en un contexto de falta de personal y burbuja salarial.

Las principales características de estos trabajos son la flexibilidad familiar, la formación y la gratuidad de ciertos complementos con el fin del acceso a una mayor disponibilidad de empleados cualificados que pueden realizar gran parte de su jornada laboral desde casa.

A.3.2. Pandemia

La pandemia covid-19 supuso en 2020-21 un verdadero experimento global que forzó por causas mayores[\[11\]](#) el uso de teletrabajo en una gran parte de la población debido a las restricciones médicas. Aunque en algunas empresas especialmente de oficinas el teletrabajo ascendió por encima del 80%, la realidad es que esta puesta acelerada donde destacaron la falta de medios, herramientas y organizaciones jerárquicas no preparadas para ello. Desde mi punto de vista, los análisis y datos de este experimento social están muy sesgados por las circunstancias y recursos asociados a los empleados teletrabajando durante esos dos años. Especialmente llama la atención como grupos principalmente personas del sector tecnológico, sin cargas familiares y con unos recursos tecnológicos

“semi-preparados” aumentaban significativamente la productividad y satisfacción. Mientras en otros con cargas familiares como niños o sin el adecuado espacio de trabajo lastraban su productividad y satisfacción. Por otra parte es complejo de evaluar, ya que muchas publicaciones son meramente estadísticas y no evalúan la formación y actuación de los mandos intermedios, ya que **gestionar empleados en remoto es algo totalmente distinto**.

A.3.3. Post Pandemia

Actualmente dependiendo de qué país, pero especialmente de qué tipo de empresa, se están decidiendo cambios en el teletrabajo, especialmente de origen político-cultural, es decir, no especialmente racionales o justificables. Se han detectado patrones de comportamientos completamente opuestos, mientras hay empresas que insisten en una bajada de la productividad global, otras muestran una subida, generando tres tendencias:

- Expansión y estandarización del trabajo en remoto, especialmente en empresas pequeñas, donde la cultura de objetivos y la digitalización favorece el debate ante una productividad mejorada o equilibrada con la reducción de costes asociados al teletrabajo.
- Trabajo híbrido, especialmente en aquellas empresas de gran tamaño que han obtenido resultados contradictorios en sus diferentes grupos de trabajo. Intenta solventar las demandas de teletrabajo de sus trabajadores y la fuga de talento con la baja productividad o mala gestión de los mismos, con cuotas mínimas de días en oficina.
- Vuelta a la oficina, gran cantidad de empresas especialmente de gran tamaño, han optado por la vuelta irremediable a la oficina cambiando el teletrabajo, por flexibilidad, es decir inferior al 30% de la jornada laboral.

Interesante es el colapso de servicios, principalmente en empresa o entes públicos, cuyas productividades ya eran bajas previamente a la pandemia, han colapsado ante el teletrabajo favorecido por la administración, ya sea por una productividad aún menor o una falta de medios, organización o la falta de atención pública adecuada, parece estar destinada como ejemplo de cómo no implementar el teletrabajo.

A.3.4. Opinión Personal

Primero de todo he de indicar que aunque soy partidario del teletrabajo 100% remoto, aceptó ampliamente la necesidad de 1 o más días en oficina y me interesa especialmente destacar ciertos puntos que no están en el actual debate de “teletrabajo vs oficina”.

El teletrabajo es una herramienta que debe facilitar la labor del trabajador, pero también los resultados para la empresa. Por ello debe cumplir un conjunto de requisitos si queremos que se utilice adecuadamente. La empresa debe proporcionar los medios tanto físicos como digitales para realizarlos, pero una parte importante recae en el trabajador, no solo a la hora de preparar su ambiente de trabajo sino de separar y auto-gestionar su manera de trabajar de su vida personal.

Los mandos intermedios y las dinámicas de equipo deben ser apropiadamente actualizadas para evitar la falta de cohesión, la baja producción o falta de tareas, y el abandono de las personas nuevas, quienes sin una presencialidad y formación no pueden rendir como un trabajador ya formado y cohesionado dentro del equipo.

En mi opinión las estadísticas del teletrabajo enmascaran comportamientos no adecuados dentro del grupo de trabajo. Especialmente aquellos trabajadores que no producen ni rinden cuentas de su trabajo, o nuevas incorporaciones que no saben, no conocen o no interactúan con el equipo, no aprendiendo ni produciendo adecuadamente.

En definitiva el teletrabajo no solo es una herramienta que ayuda a conciliar, hace evidente quien trabaja y quien no, donde existe una gestión de equipo planificada o no; y donde hay un equipo cohesionado o fraccionado.

Por ello muchas veces la presencialidad es necesaria para solucionar un problema; la dedicación de un día presencial cada 1-3 semanas para el equipo es necesario. Así como la dedicación de varias semanas con presencialidad para aquellos senior y junior involucrados en la formación de nuevas incorporaciones. Por último filosofías como SCRUM, e indicadores (chat, correos, commits, horas conectado) pueden ayudar generar índices de actividad que detecten cuando existe un problema, no solo de baja productividad sino de sobrecarga de trabajo o estrés, también excesivamente habitual en teletrabajo que tarde o temprano derivara en una baja productividad por “el síndrome del trabajador quemado”[12].

En mi opinión, hay que evaluar las tendencias a largo plazo, segmentando cada caso con su peculiaridad y evaluando de manera continua como mejorar o mantener no solo una productividad óptima sino una satisfacción continuada.

A.4. Ofertas laborales

Actualmente podemos observar la siguiente amalgama de ofertas laborales que integran la palabra “teletrabajo”:

- “Flex work”: trabajo flexible, condiciones flexibles en el horario de trabajo así como un máximo de 30% fuera de la oficina. No es teletrabajo legalmente en España.
- “Telecommuting”: se refiere principalmente a trabajadores que realizan su trabajo a distancia, principalmente focalizados en clientes. Aunque la traducción es “teletrabajo” se refiere principalmente a ventas o personal de enlace que viaja regularmente. En muchos casos aplican un 50-20-30, la mitad de la jornada trabajan desde casa, 20% en la oficina de la empresa y 30% en viajes o reuniones con clientes.
- “Partial remote”: Aplica a teletrabajo-legal, indica que es obligatorio asistir al menos 1/2/3 día a la semana y suele combinarse con “flex work”. Es el más utilizado sobre todo en metrópolis, donde el objetivo es minimizar los desplazamientos y mantener a una plantilla “metropolitana” o local. Permite reducir el tamaño de la oficina.
- “Work on Objectives”: ofrecen a sus trabajadores aquellas modalidades que ellos más deseen. El único objetivo de la empresa consiste en que las entregas y objetivos se cumplan dando libertad al equipo para auto-gestionar su forma de trabajar.

- “Full remote”: Esta modalidad indica que para realizar la actividad laboral no es necesario ir a la oficina. Sin embargo, normalmente se realizan reuniones periódicas cada 2/3/4 semanas con el fin de sincronizar, facilitar la comunicación y cohesionar el equipo. Los trabajadores pertenecen a un mismo país o región (4-6 horas-distancia). La oficina está pensada como lugar de encuentro no de trabajo intensivo.
- “International Full remote”: Aplica la modalidad “full remote” internacionalmente, por lo que las reuniones presenciales son escasas o trimestrales. Suelen generarse grupos regionales que quedan para cohesionar equipo y realizar reuniones físicamente-parciales, pero online entre los diferentes subgrupos.
- “Remote Freelance”: Personas que se dedican a colaborar o realizar pequeños trabajos. Trabajan en remoto mayoritariamente, desde su casa, pero ocasionalmente se desplazan al cliente para acordar entregar y establecer la dinámica de trabajo. Similar al “work on objectives” y al “full remote” su característica principal es la autogestión y la participación como trabajador externo dentro del grupo de trabajo. Legalmente no son teletrabajadores ya que actúan como trabajadores por cuenta ajena (coloquialmente llamados autónomos en remoto).

A.5. Teletrabajo overemployed

Desde 2021, se ha hablado de manera abierta del ‘overemployed’, focalizado en personas que teletrabajan en dos trabajos. No es el objetivo de este documento evaluar la ética o profesionalidad de dichas casuísticas, por lo que entendemos que el contexto es legal, no solapada y en muchos casos el 2º trabajo es dedicación personal a una empresa propia.

Es de interés^[9] que en muchos de estos casos, la infraestructura física, así como el aislamiento cibernético, red, dispositivos, vpn, en muchos casos se acerca al nivel teórico remarcado por este documento, especialmente entre la separación y uso del setup para usos personales y profesionales. Gran cantidad de periféricos, estrategias y softwares-herramientas se han solapado con los requisitos de este documento.

A.6. Caso de estudio

En este trabajo nos interesa aquellos casos donde existe una realidad de teletrabajo-legal, y dicho trabajador realiza mayoritariamente su actividad desde casa ya que son aquellos que mayor desafío y problemática generan con unos requisitos superiores tanto a nivel de setup, espacio como de herramientas de trabajo.

Por simplicidad así como afinidad a la profesión de este autor, se focaliza en la aplicación directa sobre Ingenieros realizando tareas de desarrollo, principalmente software pero fácilmente aplicable a diseños electrónicos, planos, prototipado y automatizaciones. De igual manera otras profesiones como diseñador gráfico, montaje audiovisual y soporte de incidencias tiene un solapamiento claro en la gran mayoría de requisitos.

APÉNDICE B. REQUISITOS Y OFICINA FÍSICA

Este anexo contiene las explicaciones, comparativas y conclusiones parciales relacionadas con la creación de una oficina física, hardware así como la documentación fotográfica y otros detalles de la implementación real de “mi oficina” como desarrollo práctico del capítulo 1.

B.1. Definición y tipos de requisitos

Definición de “**mínimo**”, entendemos un requisito de mínimos aquellos que permiten realizar el trabajo no exentos de problemática y repercutiendo en la performance o agilidad durante el trabajo. Ejemplos que permiten trabajar de manera “mínima” son aquellos donde tenemos un pc/laptop y una o varias de las siguientes casuísticas:

- Mesa y silla, no especializada y compartida para otros usos.
- Un pc/laptop que no cumple los requisitos mínimos para hacer el trabajo de manera holgada y afecta negativamente tanto en su uso como en el tiempo requerido.
- Un espacio de trabajo compartido, ruidoso, con constantes interrupciones ajenas al desarrollo profesional.
- Conexión de internet insuficiente, que limita las comunicaciones y retrasa significativamente el trabajo.

Definición de “**adecuado**”, este apartado puede variar en función de la especialización del trabajo a realizar. Mientras un soporte de incidencias necesita de unos requisitos de hardware básicos, buena conectividad; un renderizado de imagen/vídeo puede necesitar de un hardware potente y una conexión mediocre. Por lo tanto, se sobreentiende que la definición promedio debe estar sensiblemente acoplada a la actividad a realizar. Como se ha mencionado en nuestro contexto nos focalizamos en la creación o gestión de software.

Un requisito “**adecuado**” debe permitir una mejora sustancial del 30-40 % en las tareas a realizar sobre un requisito de “mínimo” y permitir evitar la gran mayoría de puntos negativos, que pueden degradar o repercutir no solo la realización y calidad de trabajo sino las comodidad y satisfacción del trabajador.

Un requisito “**óptimo**”, depende en gran medida de la especialización y los gustos del propio trabajador, incrementa normalmente la satisfacción del trabajador aunque no es una mejora sustancial en la productividad del trabajo intrínseco.

De una manera muy similar podemos hablar de 4 amalgamas presupuestarias, “**básico**” usualmente acoplada a la definición de “mínimo”, “**profesional**” focalizada en calidad precio cumpliendo los requisitos de “**adecuado**”, “**profesional pro**” excede los requisitos de adecuado y se acerca a “**óptimo**” a un precio justificado y finalmente “**business**”, excede no sólo los requisitos de óptimo y normalmente malgasta recursos o opta por acabados y marcas de mayor prestigio que no aportan mejoras medibles.

B.2. Requisitos físicos

Este apartado evalúa y compara diferentes niveles mínimos/adecuados/óptimos como requisito físico, es decir, elementos de hardware, espacio e instalaciones para la realización de un trabajo profesional.

B.2.1. Área de Trabajo

Un cubículo público como el que podemos encontrar en las bibliotecas de la UPC (fig. B.1) es el área de trabajo mínima. Un área de trabajo de menos de 1 metro cuadrado capaz de almacenar un portátil y documentación auxiliar, conectividad WIFI, alimentación eléctrica, una silla y como se aprecia en la imagen un área personal aprox. de otro metro cuadrado que en muchos casos delimitada por biombos, marquesinas o cristales translúcidos.



Figura B.1: Cubículo biblioteca UPC (campus nord)

El objetivo de esta área no solo es soportar los elementos de trabajo, sino aislar de distracciones o interacción social al trabajador. Usaremos este cubículo como definición mínima, duplicaremos los requisitos para adecuada y generalizamos en una habitación o pseudo habitación reservada como óptima.

Iluminación artificial (mínimo), compaginada con luz natural (adecuado), luz regulable, persianas, estores, tanto natural como artificial (óptima).

Ventilación diaria manual (mínimo), ventilación natural o automática (adecuado), espacio aclimatado con filtros, regulador de humedad y porcentaje de aire externo (óptimo).

B.2.2. Otros elementos

Es de importancia significativa la selección de mobiliario y elementos auxiliares tales como alfombrillas (mínimo), mesa o escritorio no especializadas (mínimo), escritorio de más

70 cm de altura o regulable (adecuado), mesa-escritorio especializada motorizada que permite trabajar de pie (óptimo).



Figura B.2: Silla y mesas, productos Amazon.

Silla ergonómica básica (mínimo), silla ergonómica ajustable con cojín lumbar (adecuado), silla gamer o business (óptimo).



Figura B.3: Porta monitor, ratón ergonómico, productos Amazon.

Elemento opcionales como, reposapiés (adecuado), ratón ergonómico / almohadilla con reposa muñecas (adecuado), eleva monitores (adecuado), porta monitor ajustable (óptimo). Debe entenderse que no solo es una cuestión de comodidad, sino que repercute seriamente en la salud de los trabajadores, y por consiguiente en el porcentaje de absentismo laboral generando un problema de salud crónico o reiterativo.

B.3. Setup informático

El setup informática es el corazón de la oficina, ya que es la principal herramienta en torno a la cual giran el resto de requisitos. Además es el elemento más customizable y en el que hay menor consenso, ya que puede ser evaluado económico, por mantenimiento, actualización e interoperabilidad.

B.3.1. ¿Cuál es tu prioridad?

Primero de todo existe una difícil decisión basada en la movilidad y el mantenimiento. Podemos disponer de un laptop que ofrezca rendimientos “adecuado” a precios “profesionales” con una movilidad e interoperabilidad a través de hubs o docking stations.

También podemos disponer de torre o semi torre comúnmente llamada PC, cuyo hardware-precio tiene mayor cantidad de recursos a menor precio fácilmente ampliable o mantenible pero requieren de un espacio dedicado y tiene nula movilidad.



Figura B.4: Portátil, mini pc y torre, google images.

Por último tenemos los mini-pc o torres-slim, contienen un hardware potente para uso ‘adecuado’, poco actualizable pero muy asequible, además de un volumen reducido y la posibilidad de ser reubicado fácilmente, sin tener la autonomía de un laptop.

Ante un presupuesto fijado a 6-8 años vista, se pueden emplear dos estrategias, ‘hardware potente mantenido’, o el uso de ‘hardware medio pero reemplazable’ a 3-4 años, siendo presupuestariamente equivalentes.

Como recomendación de este autor adoptó ampliamente del uso portátil como elemento más interdisciplinar, autónomo y de fácil uso. Sin embargo para aquellas startups o oficinas físicas el uso de mini-pc puede ser un nicho muy interesante. La estrategia de reemplazo dependen de las circunstancias tecnológicas del mercado en el momento de la compra que son explicadas en los apartados de CPU y RAM.

B.3.2. Pantalla, comodidad y opinión

Tenemos ‘el dilema del tamaño, forma y número de pantallas’; no existe un consenso claro, excepto de que es necesario 1 o más monitores para trabajar “adecuadamente”, es decir, la pantalla del laptop más un monitor extendido; una gran pantalla o dos pantallas en caso de pc o mini-pc.

Existe un gran debate sobre si la disposición de los monitores debe ser 16:9 en tamaño de 21'-24'/27' o de 21:9 en 30'-34' pulgadas. Mi punto de vista es: dos pantallas en caso de pc; depende del espacio, presupuesto y tamaño en el caso de laptop.

En mi opinión, si el portátil es “versátil” en movilidad, no supera las 13'-15' por lo que requiere dos pantallas como un pc, de 24' o 27' ambas con las mismas características. Si el portátil tiene una pantalla “adecuada” para trabajar 16'-17', puede elegir una única pantalla extendida, siendo recomendable 30'-34' disposición 21:9.



Figura B.5: Monitores, formas y geometrías [153].

La realidad es que si el presupuesto y el espacio lo permite, he llegado a ver el uso de 3 y 4 pantallas obviando la propia del portátil.

Requisitos indispensables son la resolución mínima FHD (1920x1080), adecuada QHD (2K) y óptima UHD (4k), tecnologías “eye care” y una frecuencia alta 60-75 hz (adecuado) para reducir el esfuerzo visual. Cualquier otro detalle fuera de estos queda catalogado como customización en función del precio-calidad, ya que mejoras sensibles en las pantallas con un gran presupuesto pueden quedar totalmente degradadas con una inadecuada iluminación ambiental.

90-2005	2000-2012	2010-2022	2016-2023
CTR 15'-24' 4:3	LCD / Plasma 15'-24' 4:3 / 16:9	LED (IPS/VA) 21'-27' 16:9	OLED 24'-34' 16:9 / 21:9 / 32:9
VGA 1024x768	VGA / DVI / HDMI HD 1280x800 1280x1024 1366x768	VGA / DVI / HDMI Full HD 1920x1080 QHD 2560x1440	HDMI / DisplayPort / usb-c QHD-UHD 2560x1440 3840x2160

Figura B.6: Evolución tecnológica monitores últimos 30 años.

Importante resaltar (véase fig. B.6) que el mundo de los monitores, evoluciona más lentamente en una perspectiva 5-10 años, donde la vida media de una pantalla excede los 12

años, por lo que suelen reemplazarse tecnológicamente cuando superan los 10 años por calidad en relación a sus prestaciones-precio.

Por lo tanto no es una cuestión de requisitos mínimos, sino de una tendencia económica, donde a un rango de precio “profesional” evoluciona con mejoras técnicas (resolución, tecnología, espacio-forma) a un precio equivalente. Como dato re-marcable[17] las pantallas más compradas en 2020-22 fueron 24', 27', 32' con resolución FullHD, así como la resolución más utilizada en webs de programadores fueron 40% superiores Full HD, 20% Full HD y 30% inferiores.

Como conclusión entenderemos que **no tiene sentido reemplazar los monitores con menos de 5 años**, así como **suele ser práctico-económico a partir de los 8 años**, pero se puede trabajar adecuadamente con ellos hasta el final de su vida útil 12 años, que es lo que reflejan las estadísticas de resolución utilizada en navegadores.

B.3.3. Hardware y recursos

El hardware y recursos empleados evolucionan en el tiempo (a una velocidad más rápida que los monitores), la vida media de un setup actual es de **4-6 años**, no superando los 8 años, durante los cuales es necesario un mínimo de actualizaciones, mantenimiento o reemplazo de piezas. Así mismo la usabilidad se resiente especialmente en baterías o equipos sin el apropiado mantenimiento.

Actualmente la gran mayoría de PC de 2010-2013 aún están en funcionamiento, 10 años después, algo verdaderamente improbable en los años 80 's, 90' s, o principios de 2000.

La realidad es que se ha ganado en rapidez y cantidad pero especialmente en multi-tasking, especialmente en programas de grandes volúmenes de datos, es decir, ‘más cantidad que calidad’. Por lo que los requisitos tanto del sistema operativo como de los programas más comunes son ampliamente movidos por hardware viejo. Por otro lado, el espacio en disco o la velocidad de acceso, así como la memoria RAM evolucionan continuamente aumentando velocidades y capacidades a un coste inferior, lo cual ha permitido mejorar los cuellos de botella en aquellos hardwares viejos con actualizaciones sencillas véase [16]. Este punto debe aclararnos que el mantenimiento es un elemento muy crítico, especialmente si esperamos extender la vida del hardware 8-12 años para otras tareas no profesionales.

B.3.4. Periféricos

Entendemos como periféricos aquellos elementos externos que usualmente se conectan por USB o bluetooth. Un hub de conectividad o hub de monitor externo es obligatorio, así como se sobreentiende un ratón y teclado. Existen varios elementos dignos de mencionar en la catalogación de los mismos:

- Auriculares y uso del micrófono-cámara del portátil (mínimo). Auriculares de alta calidad con micrófono (adecuado), auriculares con cancelación de ruido y micrófonos HD (óptimo), web-cam HD con micrófono con cancelación de ruidos (óptimo).
- Plataforma de elevación laptop (adecuado), plataforma hub con refrigeración activa laptop (óptimo).

- Sistemas de autenticación externos, lector de huellas, lector de tarjeta etc.... (óptimo).
- Panel táctil y bolígrafo asociado (óptimo), pantalla táctiles asociada a monitor o laptop (óptimo).
- Google home, Alexa, u otros elementos de domótica, automatización, fuentes musicales o notificaciones. (óptimo, pero requiere de análisis riesgos de seguridad).
- Requerimientos especiales, impresora-scaner, impresoras 3d, electrónica de monitorización o placas prototipado. (solo bajo necesidad práctica).

B.3.5. Gestión del setup

Existe una norma bastante compleja ya que la gestiona cada individuo, cuyo objetivo es separar el trabajo de tu vida personal. En muchos casos aunque un despacho o habitación dedicada a tu setup sirve como jaula de aislamientos y concentración. La realidad es que gran cantidad de profesionales tiene intereses alineados o coincidentes con sus labores profesionales y de igual manera tiene sus necesidades personales de acceso a la información, cuentas personales, almacenamientos, documentos y juegos.

¿Cómo separar el pc personal del profesional?

Un buen profesional no debe usar su setup profesional para usos personales, primero para mantener esta estricta separación que no afecte o predisponga a distracciones de índole personal durante la jornada laboral. Por otra parte están los riesgos de ciber seguridad de generar un agujero de seguridad desde sus cuentas personales a sus servicios profesionales.

B.3.5.1. Aislamiento de navegación

La gran mayoría de servicios personales a día de hoy son capaces de ser usados vía navegador, así como el acceso a noticias, comunicaciones en línea y servicios streaming como pueden ser música. La primera capa de seguridad es separar y la gestión de nuestras cuentas personales y profesionales en diferentes cuentas de sincronización. Permitiendo acceder a dos cuentas vía nuestro navegador, y evitando almacenar contraseñas, historial de navegación, cookies y otros elementos. Una separación más estable es el uso de diferentes navegadores para cada una de las cuentas, así evitamos una gestión compleja o la afectación de plugins inseguros en el navegador.

En mi opinión son una buena opción, aunque solo debe usarse en momentos de necesidad y debe evitarse lo máximo posible.

La opción más adecuada para dichos casos tales como revisar mails, esperar un mensaje vía whatsapp o simplemente poner tu playlist de relajación en los altavoces pasa por el uso de **un 2º elemento personal móvil/tablet/laptop** capaz de surtir dicha función sin necesidad de usar tu hardware profesional e **incluso utilizando diferentes redes** para conectarse.

B.3.5.2. Setup Commutable

Sin embargo qué hacemos cuando en la post jornada laboral deseo trabajar en mi hobby tecnológico, o simplemente jugar o ver series desde mi pc personal. ¿Debo acaso duplicar recursos teniendo un precioso despacho con un setup con múltiples pantallas?

La respuesta es simple; **No**. Si seleccionamos adecuadamente los monitores a elegir, así como el dock-station o hub de periféricos, pronto entenderemos que es fácil disponer de hasta 3 o 4 fuentes de imagen independientes, múltiples fuentes de sonido y un uso compartido de periféricos.



Figura B.7: Conectividad de monitores, amazon products.

Se puede optar por la estrategia centralista, todo se conecta a nuestro hub y con conmutar la conexión usb hub-portátil de laptop profesional a personal ya tenemos el setup completo para uso personal. O se puede seguir una estrategia de fuentes en paralelo, de tal manera que ambos hardware están conectados a puertos diferentes o a través de un multiplexor. Si solo uno está encendido, basta con cambiar de pc/laptop el usb que controla el teclado-ratón o el hub de usb y ya podemos utilizar el setup. Si ambos están encendido, requiere seleccionar en las pantallas la fuente a mostrar y conectar el teclado-ratón aquel que queramos usar.

En mi opinión la segunda opción es más interesante aunque compleja, requiere de una mayor planificación, pero facilita la simultaneidad de 3 o 4 pc, pudiendo estar conectado en sesiones en remoto desde aquel que gestione las pantallas.

B.4. Abastecimientos auxiliares

Abastecimiento auxiliares, son todas aquellas infraestructuras o herramientas necesarias para poder usar nuestro setup adecuadamente. Entre las principales necesidades tenemos los suministros de electricidad, climatización y acceso a internet. Otros interesantes son aquellos que usamos como alternativa funcional cuando un suministro básico falla, sistemas de alimentación ininterrumpida (SAI) o equivalentes, acceso alternativo a la red, setup mínimo alternativo o una planificación aceptable en caso de NO poder realizar home office.

B.4.1. Suministro Eléctrico

Este elemento parece simple y obviamente necesario, pero a parte de la disponibilidad de red eléctrica y de la partida compensatoria en el acuerdo de teletrabajo qué implicaciones tiene en nuestro oficina el suministro eléctrico.

Disponibilidad de enchufes o switch, verdaderamente es complejo alimentar 2 pantallas, un hub, un portátil, una lámpara y algún que otro periférico extra. Aun lo es más si el sistema eléctrico es viejo y el único enchufe disponible tiene una amalgama de extensores-ladrónes interconectados entre ellos. Recordemos que los enchufes tradicionales tienen 10-16A como límite, así como muchos 'alargos' no utilizan cable de 2.5 mm^2 sino 1.5 mm^2 de sección.



Figura B.8: Enchufes e instalación eléctrica.

Como conclusión hemos de recordar que si bien un setup no es una actividad industrial requerirá de un mínimo de 6 enchufes-switch y raramente supera un consumo límite de una toma en torno a los 2000-2500W pero si puede ser un problema en una oficina de una startup afincada en un antiguo piso, especialmente cuando se usan elementos como calefactores o múltiples estaciones de trabajo no planificadas previamente. En dicho caso recomendamos no solo la instalación de múltiples switches con el cableado adecuado, sino el aislamiento de las filas o islas de trabajo, en PIAs[18] independientes con el fin de aislar y detectar fallos eléctricos que no afecten a la oficina de manera generalizada.

B.4.2. Climatización

Este punto tiene controversia entre oficinas como en gastos de teletrabajo. En mi opinión, la temperatura de trabajo óptima es un tema muy personal, sin embargo se recomienda entre 23º-27º en verano y 17º-24º en invierno. El verdadero punto importante es la existencia de una ventilación continua y adecuada que reduzca los riesgos contaminantes, alérgenos y degradación de la calidad del aire.

Vivo en una zona privilegiada “Costa Daurada” de Tarragona cuyo invierno no baja de los 8º-10º y las temperaturas veraniegas raramente superan los 38º. Por lo que con apenas el aislamiento de la vivienda y una adecuada ventilación bien puede entrar dentro del intervalo de confianza de manera pasiva. Sin embargo nadie ha hablado de la humedad, la cual en la costa es siempre superior al 60% (invierno) y mayor al 90-95% (verano).

A su vez, un trabajo en remoto significa, poco movimiento físico y contacto continuado de extremidades con silla y escritorio, por lo que por experiencia propia puedo decir que no se puede trabajar adecuadamente con más de 28º en verano, ni menos de 18º en invierno, sea cual sea la ropa utilizada debido a la alta humedad ambiente y la naturaleza del trabajo.

Como conclusión los ventiladores o calefactores eléctricos son “mínimos”, un splitter de bomba calor frío “adecuado”, usar la aclimatación general de la casa “óptimo” pero costoso. En la instalación de la caseta se optó por un pingüino (bomba de calor/frió), sin embargo se ha desmantelado y sustituido vease mejoras de 2023 [B.5.2..](#)

B.4.3. Acceso a Internet

Los requisitos de acceso a la red son un elemento crítico para el home office, no solo por las características necesarias sino porque dicho medio es compartido en la vivienda por los usos particulares de la familia. Por lo tanto la “velocidad” de la conexión debe permitir

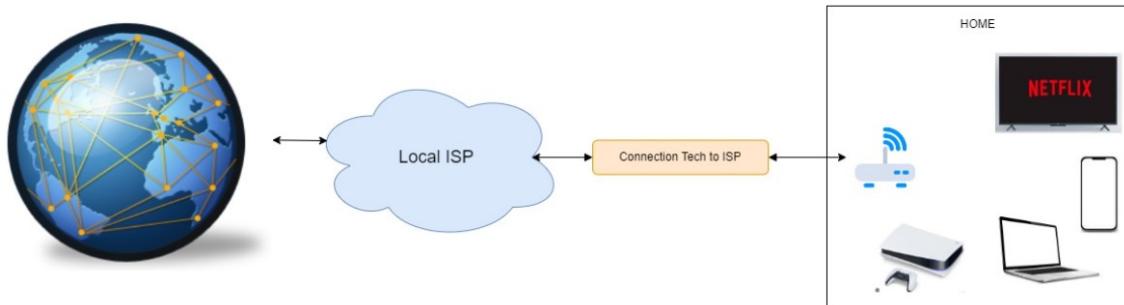


Figura B.9: Diagrama de conexión.

establecer videoconferencias junto al uso de streaming (netflix,spotify...) u otros usos personales de alta demanda. En la implementación física de mi oficina, usamos mi conexión familiar de 600 Mbps de fibra simétricos que excede ampliamente las necesidades.

B.4.3.1. Ancho de banda

Ancho de banda (BW), tasa o velocidad es la capacidad de transmisión de datos por segundo que permite la conexión de manera estable e ininterrumpidamente. Existen conexiones simétricas, velocidad de subida (uplink) y bajada (downlink) idénticas, y asimétrica donde la uplink suele ser inferior (entorno a un 20-30 % del downlink).

Una video conferencia exigente, con múltiples usuarios enviando imagen y sonido HD con posibilidad de pantalla compartida, requiere de un mínimo de 4-8 Mbps. Una transmisión HD en netflix oscila entre 3-5 Mbps, 15 Mbps si es 4K. Se sobre entiende que en un contexto de coexistencia familiar y de posible ejecución en paralelo de más actividades profesionales, un uso mínimo requiere una conexión de 10/3 Mbps (uplink/downlink), una conexión adecuada **15/5 Mbps** y una óptima son aquellas con tasas superiores a los 20/10 Mbps.

El ancho de banda de una conexión está intrínsecamente definido por la tecnología de acceso proveída por el ISP local, ejemplo conexión fibra movistar 100/100 Mbps. Pero influenciado por la congestión del ISP local o nodos intermedios, como ejemplo de las 100 megas simétricas, mi conectividad con diferentes servidores españoles puede ser de 60-80 Mbps (inferior a la conectividad directa con el ISP) y usualmente asimétrica, pero podemos obtener medidas de 15-30 Mbps con servidores asiáticos.

B.4.3.2. Latencias

La latencia, coloquialmente llamada lag o ping, es el tiempo medio de ida o vuelta en una conexión de red, es decir el tiempo mínimo de interacción entre dos elementos de la red.

Las personas detectamos que la comunicación no es “instantánea”, si la interacción de dos interlocutores es superior a los 300 ms (1/3 segundo), por lo que en telefonía se establece un umbral de 150 ms como latencia máxima a partir de los cuales se percibe una comunicación deficiente, las diferentes plataformas de videoconferencias marcan como deficiente una conexión con ping superiores a 100 ms. Por lo tanto latencias 100-150 ms (mínimo), **50-100 ms (adecuado)**, inferiores a 50 ms óptimo.

Se ha de resaltar que el tipo de tecnología usada entre la oficina y el ISP (red de acceso) puede suponer el 60% de la latencia a excepción de la fibra. Pero además aquellas conexiones con más nodos intermedios, es decir, normalmente más lejanas geográficamente acumularan una mayor latencia, debido a la red de transporte.

B.4.3.3. Jitter

Jitter o fluctuación de retardo es la variabilidad temporal del retardo. Conceptualmente podemos entender que si la latencia es el valor medio, la medición del jitter puede realizarse de diversas maneras como aquellos retardos máximos o usualmente estadísticos como la desviación típica de la latencia.

La principal consecuencia en comunicación online es que aunque la latencia media es aceptable, puede que un 20-30 % de los paquetes tengan retardos mayores o desiguales.

Aunque puede solucionarse con estrategias de buffer, en videoconferencia los buffer tienen tamaños de 50ms, implica la pérdida de información y por ende cortes o inteligibilidad

de la comunicación. Por lo tanto jitter max 40 ms requisito mínimo, **inferior a 30 ms adecuado**, inferior a 10 ms óptimo.

B.4.3.4. Tecnologías aptas

A la hora de analizar las diversas tecnologías de acceso a internet, las evaluaremos desde el punto de vista “adecuado” (verde), azul (óptimo), aquellas conexiones insuficientes (naranja) para adecuado y aquellas conexiones que no cumplen los requisitos mínimos(rojo).

Tabla B.1: Comparación Tecnologías de acceso

	Velocidad Downlink	Velocidad Uplink	Latencia	Jitter	Coste
Requisitos Adecuados	15 Mbps	5 Mbps	75 ms	30 ms	Profesional
ADSL/ADSL2 (zona alejada)[21]	6-8 Mbps	1-2 Mbps	50-90 ms	15-25 ms	Básico
ADSL/ADSL2+ (zona céntrica) [21]	16-22 Mbps	4-6 Mbps	30-60 ms	15-20 ms	Profesional
VDSL2 (zona céntrica) [21]	15-50 Mbps	5-18 Mbps	15-50 ms	10-20 ms	Profesional
Coaxial/cable	30-300 Mbps	3-30 Mbps	10-30 ms	10-15 ms	Profesional
3G/4G	10-80 Mbps	5-30 Mbps	20-80 ms	15-30 ms	Profesional Pro
WIMAX Otros Radio link[19]	6-30 Mbps	1-3 Mbps	30-80 ms	15-40 ms	Profesional Pro
Satellite Link (traditional GEO)	10-50 Mbps	1-6 Mbps	500-800 ms	20-50 ms	Profesional Pro (Spain*)
Satellite Link (LEO)	1-10 Mbps	0.5-2 Mbps	50-150 ms	20-40 ms	Profesional Pro
StarLink[20]	100-400 Mbps	5-50 Mbps	15-50 ms	10-40 ms	Business
Fibra Básica	50-100 Mbps	10-50 Mbps	5-15 ms	5-10 ms	Básico-Profesional
Fibra	300 Mbps - 1 Gbps	100 Mbps - 1 Gbps	1-15 ms	1-10 ms	Profesional Pro

Se ha de destacar que desde el punto de vista práctico calidad-precio, las mejores opciones son **la fibra básica, conexión coaxial o VDSL+ [21] en zona céntrica**.

Aquellas conexiones que dependen de enlaces radio (naranja), cumplen los requisitos, pero tienen valores inferiores o degradados en aquellas áreas geográficamente problemáticas, las cuales no tienen la posibilidad de otro tipo de conexión.

Una vez tenemos una conexión adecuada, en muchos casos el factor limitante no es el

tipo de conexión sino el medio de transmisión en la propia vivienda. Las tecnologías más utilizadas son WIFI, PLC[22] y cableado.

Obviamente cableado es la mejor opción permitiendo 1-2.5 Gbps, robusto a interferencias, y no afectado por el número de elementos en la red, así como físicamente ciberseguro. Los PLC, permiten velocidades de 10-500/5-50 Mbps no aumentando significativamente la latencia ni el jitter, por lo que son una alternativa mejor que los medios inalámbricos en muchos casos.

El uso de Wifi y repetidores depende tanto de la vivienda, materiales así como elementos conectados. Es de especial interés entender que cuanto mayor sea el número de elementos conectados, o repetidores, menor será la velocidad efectiva de acceso al medio puesto que este es compartido. Es un medio inalámbrico sensible a interferencias y comparte canales con vecinos-geográficos que interfieren la señal. Por otra parte incrementa sensiblemente tanto latencias pero especialmente Jitter, tiene pérdidas más significativas que los medios físicos no inalámbricos.

Así mismo se debe comprender que la velocidad media será afectada por el elemento más lento (más alejado y con estándar más antiguo). Por lo tanto se recomienda la coexistencia de viejos y nuevos estándares con la finalidad de que aquellos elementos de estándares b/g/n se conecten a un AP diferente de los ac/ax. Así como una separación del AP-profesional del AP-familiar. Por último es interesante evaluar si definir canales estáticos, minimizando la interferencia externa y usando canales diferentes en nuestros AP, o el uso de algoritmo de asignación automática.

Conclusión el wifi es aquel elemento que puede tener un rango tan ambiguo de aplicaciones y casuísticas que aun permitiendo en algunos casos mejores performance que cableado (wifi 6 802.11.ax) y siendo la instalación más ágil y sencilla, no es la recomendación de este trabajo. Se recomienda el cableado o uso de PLC como conexión preferente, con opción a la instalación de un punto wifi en la localización del Setup para uso exclusivo profesional (separación de redes).

B.4.3.5. *StarLink o el 5G*

La principal característica de las nuevas tecnologías de comunicación inalámbricas se basa no solo en un aumento significativo de la velocidad de transmisión sino de una reducción drástica de las latencias en la red troncal del operador de telecomunicaciones.

Por ello el 5G, su principal virtud no es la alta velocidad sino la baja latencia y jitter, así como la movilidad urbana que permite.

De una manera muy similar Starlink permite la comunicación directa de su red satélital de órbita baja mediante láseres. La reducción de las distancias respecto a satélites en órbitas más altas, así como su cobertura global y facilidad de comutado en su red esférica, permite obtener bajas latencias inferiores a las conexiones físicas por fibra entre puntos muy distantes en el mundo y una disponibilidad de cobertura completa con excepción de las regiones polares.

Estos puntos pueden ser de interés para actividades estratégicas, fast-trading mundial, seguridad e independencia política de la conexión, o un modo de vida nómada, como puede ser un freelance que vive-trabaja en una casa-barco, moviéndose por diferentes regiones, pero totalmente descartable para el 99 % de los casos de uso común en España,

especialmente debido a su **coste business**.

B.4.4. Circunstancias e histórico

Desde 2014 en mi etapa de becario me he dedicado a tareas concretas de ingeniería entremezcladas con la automatización y la programación desde algorítmica básica, programas de gestión. Así mismo desde 2016 me he dedicado a programación, especialmente backend de manera profesional en frameworks de php, python y java.

En 2018 me mudé a un barrio residencial baix penedes que es desde entonces mi actual residencia. El lugar puede describirse como el perfecto barrio residencial, a menos de 1 km de la playa, en una zona muy tranquila zona ampliamente turística y llena de segundas residencias dentro del límite del área metropolitana de Barcelona.

Un punto importante de esta localización, es la pésima infraestructura tecnológica e incluso infraestructura eléctrica de las costas del Baix Penedes. Históricamente la nula planificación, la rápida y exponencial urbanización durante el boom inmobiliarios del último lustro, junto a una infraestructura no dimensionada correctamente para las fluctuaciones de la temporada turística concluye en las siguientes casuísticas:

- Apagones, fluctuaciones de corriente o problemas variados durante las tempora da turística (junio-septiembre) y navidades, problemas eléctricos relacionados con inundaciones en otoño. Que pueden dejar la zona durante intervalos de 2-3 horas varios días ante una incidencia y su resolución.
- Nula conectividad por cobre o fibra, únicamente proveedores de WIMAX[19] o wifi por antenas point to point y conectividades inferiores a 5 Mbps en cableado antiguo de cobre. Completamente deficientes bajo condiciones meteorológicas adversas y afectadas por picos de demanda.

B.4.4.1. Pre pandemia

Se disponía de una habitación de estudio para mi etapa estudiantil de máster, hobbies o zona de estudio/trabajo compartido con mi pareja.

Curiosamente pocos meses antes de la pandemia, con el objetivo de cumplir la cobertura 90% establecida por ley para 2020[24], se desplegó satisfactoriamente una red de fibra en toda la zona residencial, permitiendo conectividades simétricas de hasta 300 Mbps.

Afortunadamente en mis planes familiares ya contaba con una reforma y mejora de la sala de estudio para adecuarla como dormitorio con escritorio, pudiendo usar como zona de trabajo o habitación de invitados, en perspectiva a un futuro familiar como habitación para un niño o niña.

Desde la llegada de la fibra también realice las incorporaciones de una pizarra y una impresora 3D para mis hobbies, así como un cableado de la habitación con ethernet para obtener la máxima velocidad y no ocupar la red Wifi.

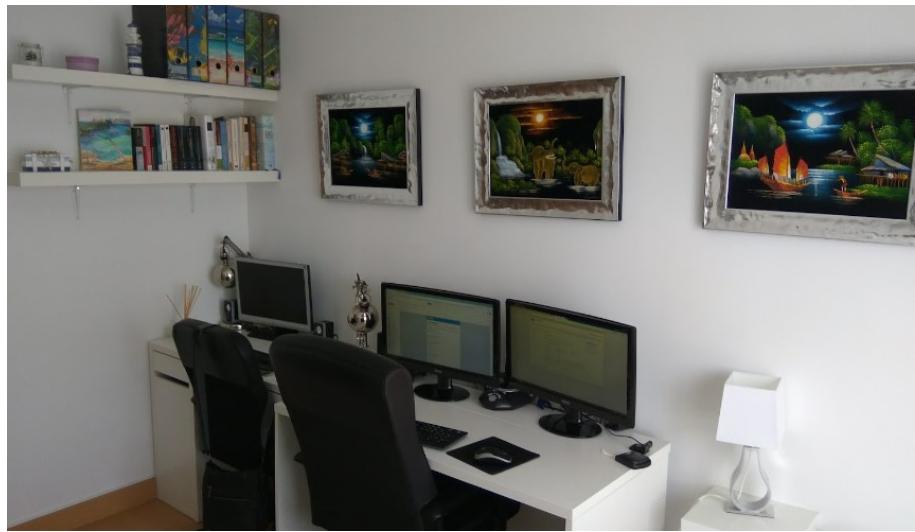


Figura B.10: Setup pre pandemia.

Tabla B.2: Setup pre pandemia tabla comparativa

	Zona de tra-bajo	Hardware	Red	Otros
Setup	Escritorio de-dicado Sillas de estudio	2 pantallas 21' HD PC old-2011 laptop 2013	Wimax 2/3Mbps Wifi g 54 Mbps	Iluminación natural y artificial ade-cuadas
Defectos o planes pen-dientes	Mesa baja y poco ancha	Viejo, uso hob-bies	Mejora a Fibra pendiente	Mayor capaci-dad de alma-cenaje y elimi-nación de 2º escritorio dimi-nuto

B.4.4.2. Pandemia

Dos semanas antes de los reales decretos que pusieron en marcha los mecanismos de aislamiento de la pandemia, la empresa en la que trabajo me proporcionó un portátil como nuevo hardware de trabajo, las credenciales y una vpn para conectarme, comenzando inmediatamente el 100% del trabajo en remoto como medida precautoria.

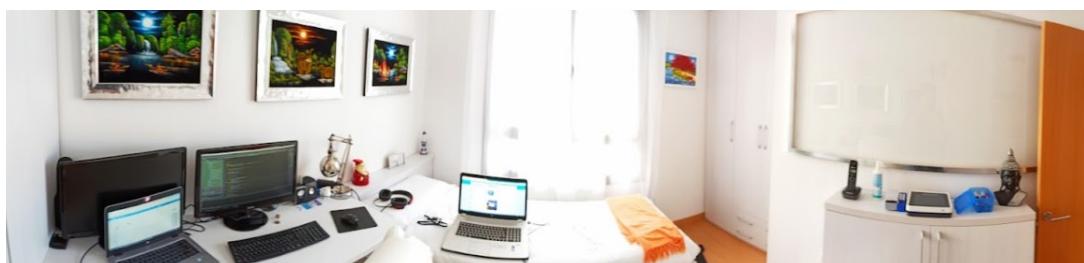


Figura B.11: Setup pandemia.

Los recientes cambios en la habitación de estudio me permitieron un teletrabajo satisfactorio únicamente degradado por pequeños problemas fácilmente solucionables como la adquisición de un split usb-vga para la utilización de ambos monitores con el portátil laboral, junto a la conexión de mi pc personal.



Figura B.12: Terraza en pandemia.

Una de mis tareas de bricolaje durante la pandemia fue la mejora con la instalación de cableado eléctrico para iluminación y enchufes en la terraza. Incluyendo red eléctrica en una caseta de almacenaje situada en la misma.

Tabla B.3: Setup Pandemia tabla comparativa

	Zona de trabajo	Hardware	Red	Otros
Setup incremental	+Zona de trabajo a medida adecuada +Mesa alta (+70cm) +Mueble a medida con pc, cableados ocultos +gran capacidad organizativa y de almacenamientos +impresora 3D oculta en armario	+incorporación de split usb en pantallas +cam HD +micro-auricular HD +switch ethernet	+Fibra 300Mbps +Wifi n 150 Mbps / 5G 300 Mbps + cableado ethernet	+Pizarra translúcida +cableado interior de la casa con ethernet + impresora 3D y otros elementos
Defectos o planes pendientes	Perfecta, pero habitación planificada invitados y futuros hijos.	Pantallas deficientes Hardware viejo	Ethernet a 100 Mbps mejorable a giga ethernet	La climatización e infraestructuras dependen de la casa.

Con el objetivo de poder disfrutar de horas de luz y la salida al exterior en los continuados confinamientos, instale un PLC[22] que permitía una conectividad de 7-12 Mbps emitiendo una señal wifi en la caseta de la terraza. Permitiendo trabajar algunas horas especialmente por la tarde cuando la intensidad lumínica no deslumbraba la pantalla.

Durante el año 2020, quedó patente la funcionalidad plena de la habitación de estudio como habitación dedicada a teletrabajo, reflejando aquellos punto mejorables, sin embargo también quedó patente la infraestructura del barrio, tiene 3-4 días anuales con incidencias eléctricas cuando el calor o el frío sobrecarga la infraestructura, requiriendo de un plan “b” para dichos días.

Este periodo lo damos por concluido tras finalizar los períodos de aislamiento excepcionales en 2021 y las campañas de vacunación. El teletrabajo 100% fue continuado como medida precautoria hasta 2023.

B.4.4.3. Post pandemia

Finalmente la mayoría de los puntos mejorables como las pantallas, uso de silla ergonómica-gamer y diversos periféricos o hub fueron subsanados.



Figura B.13: Setup post pandemia.

Sin embargo, desde el verano de 2021, quedó patente que el principal problema era la “planificación familiar”, básicamente esa habitación estaba destinada a habitación de niño/niña con escritorio propio. El final de la pandemia reinicio los planes familiares pa-



Figura B.14: Setup 2023.

ralizados. Como conclusión aun con la mejora sustancial del setup y sus deficiencias era necesario conseguir una alternativa real en un periodo de 12 a 18 meses incluyendo un nuevo lugar o habitáculo para el setup. Ya que a finales de 2022 el aspecto de la habitación-teletrabajo pasó a ser el siguiente:

B.4.5. Nueva habitación

Vivo en un ático de $90\ m^2$, desgraciadamente la disposición espacial no es eficiente ($75\ m^2$ útiles) y siempre hace falta espacio de almacenamiento. La predisposición del salón o habitaciones no facilitan la instalación de una zona de trabajo, por lo que no es posible, sin dedicar una habitación en exclusiva.

En primer lugar de interés al igual que muchos amantes del bricolaje es “el trastero”, ya que unos escasos $2-6\ m^2$ predispuestos son más que suficientes para nuestras necesidades, sin embargo, los pisos de mi bloque no cuentan con trasteros propios, por lo que coloque una caseta de pvc en la terraza que realiza dichas funciones de almacenaje.

El segundo lugar de interés ampliamente utilizado en ciudades como Barcelona, Castelldefels o Gava, son los balcones y galerías. Espacio estrechos pero alargados fácilmente convertibles mediante cerramiento de aluminio acristalado para conseguir un espacio extra. Aunque cuento con un amplio balcón-terraza de $10\ m^2$, la normativa de mi municipio no permite su cerramiento, ya que contiene una escalera de caracol que da acceso directo a la terraza superior.

Tercero, cerramiento de parking o estructura metálica sobre el parking. Dependiendo de las dimensiones acceso y altura del parking, es posible dedicar ciertos metros a una diminuta habitación mediante cerramientos simples o la instalación de estructuras metálicas con el objetivo de obtener la superficie de parking en vertical para la instalación de un trastero. La normativa comunitaria y especialmente la seguridad de la misma no permiten la instalación de dichas estrategias ni lugares cerrados de almacenaje, por riesgo de incendio.

Finalmente al ser ático, dispongo de una amplia terraza, algo mayor que el 50% de la superficie del piso ($50-60\ m^2$) que sin embargo, debido a su predisposición en “L”, un pasillo de acceso y una chimenea-respiradero en la parte principal más ancha, no tiene un uso práctico mayor a $35\ m^2$, donde el drenaje de la misma es muy deficiente con zonas inúndables marcadas en círculos en el plano (fig.B.15) que indica las pendientes de drenaje. Además es una terraza comunitaria de uso privativo, es decir, que de acuerdo a las normativa municipal y la ordenanza comunitaria, no se pueden instalar ningún tipo de elemento que:

- Se fije o taladre al suelo comunitario.
- Rompa la estética (color) del edificio (blanco) y grises (baldosas).
- Medianeras, pérgolas, mobiliario de terraza deben ser del tipo, color y dimensiones establecidas en las reglas comunitarias.
- Deben ser elementos de carácter no fijo (desmontables), barbacoas, barras, armarios o cerramientos únicamente anclados en paredes o alfizares.

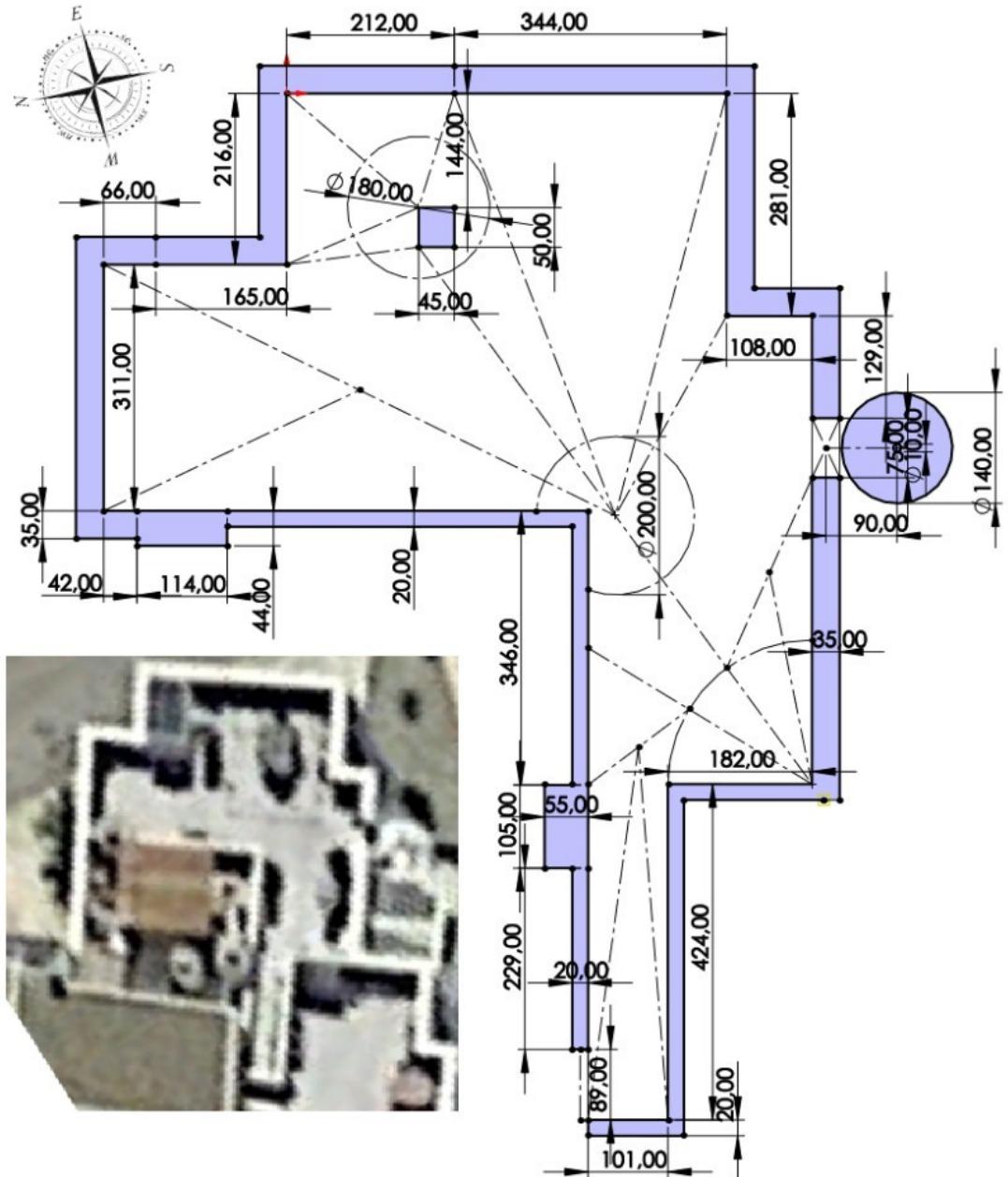


Figura B.15: Terraza planos, pendiente y zonas inúndales.

Como especial interés, se definieron las separaciones de medianil, por paneles de aluminio lacado blanco, fuertemente anclados en los muros. Permitiendo cubrir muros vecinales o espacios entre paredes y chimeneas, siempre y cuando hubiese consentimiento mutuo entre vecinos y no se anclaje nada en fachada o paredes comunitarias.

B.4.6. Caseta Oficina

A veces las soluciones son una combinación de casuística y defectos utilizados a tu favor. Durante la pandemia intentamos solucionar los problemas de la terraza, falta de privacidad, una gran cantidad de espacio infra usado y las continuas inundaciones de la caseta-

pvc usada como trastero. Así mismo la instalación del PLC y el “trabajo desde exterior” me dejó claro que con el cerramiento adecuado, era el lugar perfecto para trabajar.

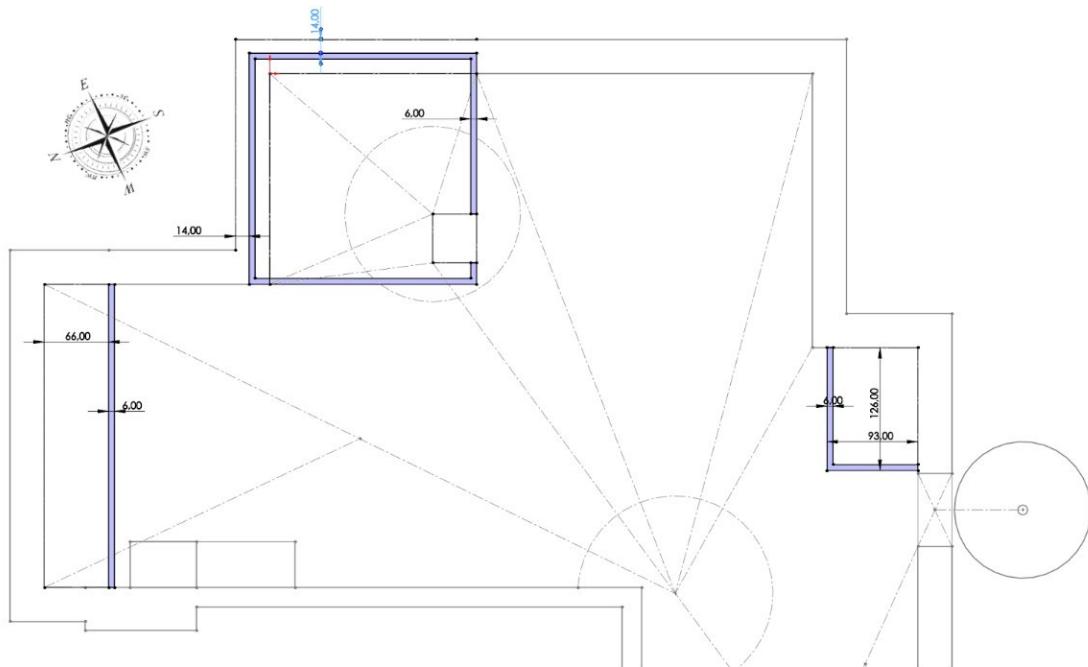


Figura B.16: Terraza cerramientos planificados con pendientes zonas inúndales marcadas.

El plan se basaba en cerrar el medianil con el separador de aluminio para ganar privacidad y que el mismo proveedor de aluminio nos hiciera un armario-almacenaje en torno a la zona no inundable de espacio muerto entre barbacoa y pared norte. Sin embargo no se llegó a entendimiento con el vecino.

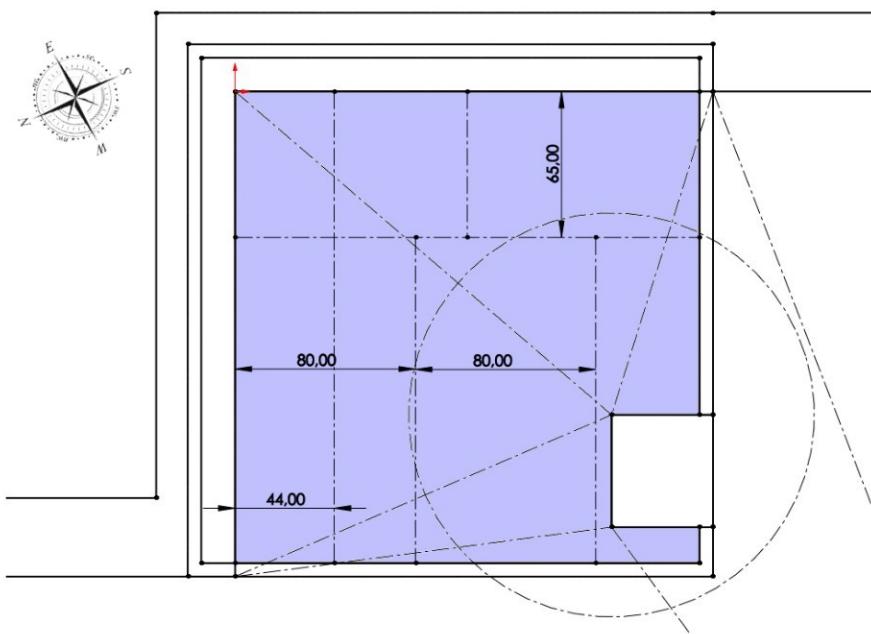


Figura B.17: Casetta división espacial.

Entonces comente la idea de similar al armario, rehacer la caseta-trastero de PVC, pero utilizando el material de separación entre vecinos. El objetivo inicial era conseguir una caseta hermética, evitando “el charco” y el moho, pero sobre todo ganando el alféizar como espacio útil y permitiendo colocar estantes en él, con el fin de obtener un pseudo trastero.

Al detallar, descubrimos que lo que normalmente es 1.80 m de altura, debido a la mala colocación de terraza y a un alféizar algo elevado, los puntos de anclaje de la futura caseta permitían unos valores de 1.86-1.89 m respecto al suelo real de la terraza, es decir, un lugar donde a mi altura 1.79, permite moverse como en una habitación de techos bajos.

Por otra parte la chimenea-respiradero, pasa de ser un elemento problemático, a un elemento estructural, que permite anclar y dar robustez al formar un cuadrado entre muro-alféizar y chimenea, permitiendo el anclaje lateral y en conclusión resistir cualquier tipo de vientos al cerramiento de aluminio. Por otra parte interiormente habilita un pared robusta sobre la que anclar elementos como pizarra o mini estanterías.

B.4.6.1. *Trabajos previos*

Durante los meses previos a la instalación de la caseta se realizó una limpieza completa de las zonas afectadas. La limpieza de alféizar, pared y suelo con agua a presión, posteriormente con bases ácidas vinagre blanco (alféizar) y salfuman (baldosas) para la eliminación y apertura de poros.



Figura B.18: Limpieza, pintura y trabajos previos.

Así como una adecuación de la instalación eléctrica con la instalación de canalizaciones, cableado y respectivas cajas con el fin de mejorar y separar completamente iluminación de terraza, enchufes húmedos, electricidad de la oficina y cable ethernet.



Figura B.19: Montaje de sistema eléctrico y cableado Ethernet.

B.4.6.2. Estructura, aislamiento y estanqueidad

La estructura y el aislamiento viene de la mano de paneles sándwich de aluminio, al igual que la separación entre vecinos se basan en anclajes verticales, que unen los paneles de aluminio cada 1.5-2 metros. La estrategia se basa en utilizar paneles más grueso de 4

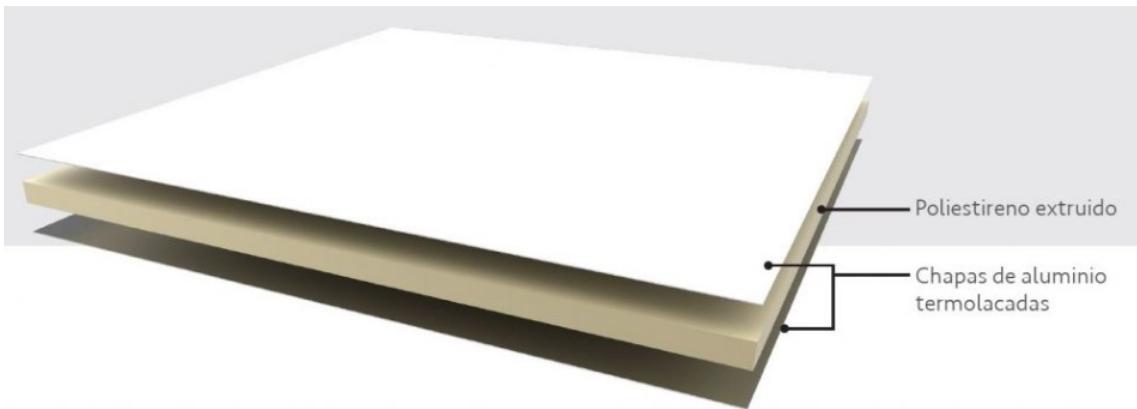


Figura B.20: Panel de aluminio lacado con aislante térmico.

cm para un mayor aislamiento, especialmente en tejado 7 cm, ya que la incidencia solar es directa y prolongada durante los meses de verano. Estos paneles, son sujetados por premarcos de aluminio (similares a ventanas) que a su vez se fijan directamente sobre alféizar, paredes o perfiles estructurales en las esquinas. Todos ellos entre sí, así como los elementos que se fijan quedan perfectamente sellados por siliconas especializadas, incluido la base del perfil en contacto con el suelo de la terraza.



Figura B.21: Montaje paneles aluminio y soportes.



Figura B.22: Vista de perfiles y techo con aislamiento.

B.4.6.3. Instalación interna y montaje mobiliario

Una vez terminados los trabajos de instalación de los cerramientos y el apropiado secado de los mismo, se inició un montaje tanto de cajas, enchufes e interruptores eléctricos de ambos cerramientos Junto con una migración del setup validando tanto conectividad, enchufes e iluminación, permitiendo comenzar el teletrabajo desde la oficina conforme se van añadiendo más elementos complementarios necesarios especialmente relacionados con temperatura e iluminación.



Figura B.23: Mobiliario y setup en caseta.

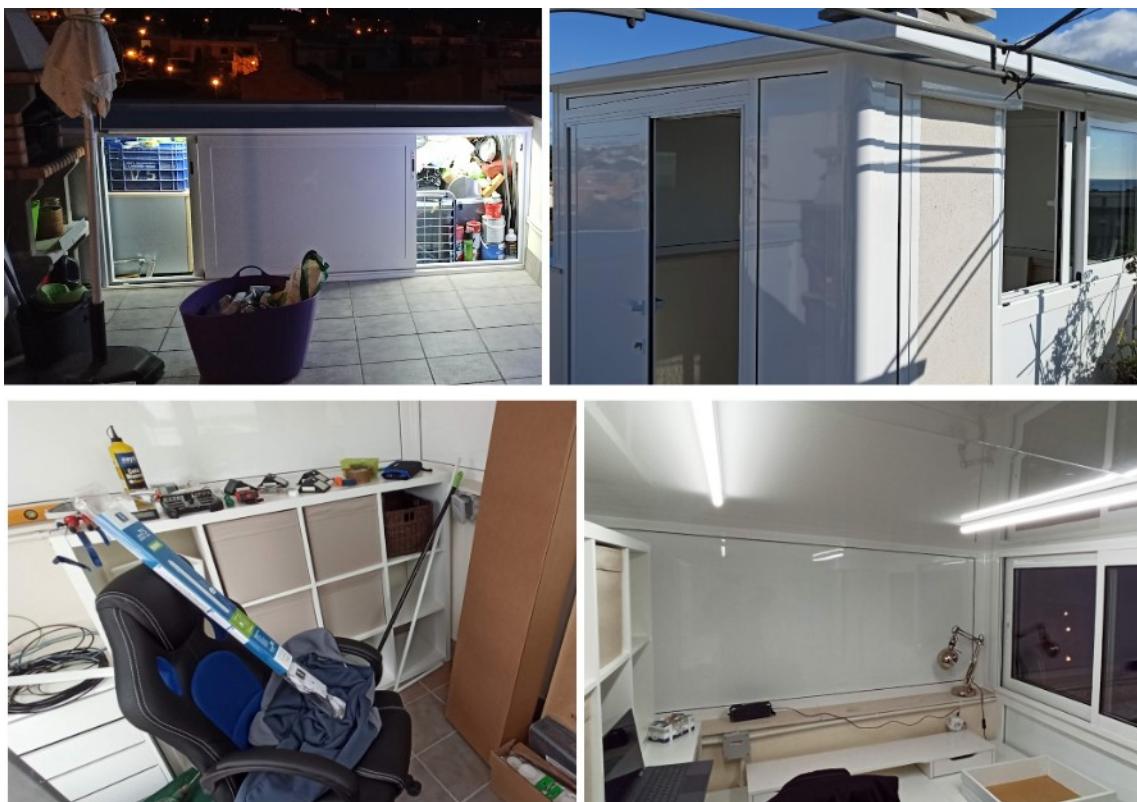


Figura B.24: Instalación eléctrica en caseta y cerramientos.

B.4.6.4. Aclimatación e iluminación

El primer elemento necesario de la “hermética caseta” es la instalación de varios respiraderos naturales para la circulación de aire, siendo estos fácilmente operados manualmente para evitar la entrada de frío/calor en circunstancias adversas.

Aunque el aislamiento bastante sobredimensionado para un cerramiento usual, no deja de ser 4-7 cm similar a una puerta de parking, es decir, bastante útil para evitar el calor directo pero insuficiente en invierno para evitar la bajada de temperaturas. Por otra parte tanto el muro como el suelo original son elementos fríos, que permanecen a la temperatura de la estructura del edificio, algo fresca en verano pero fría en invierno.

Por lo tanto se han instalado alfombras (duras y gruesas) junto a un calefactor eléctrico,

así como ventiladores pingüino de bomba de calor aire frío/calor.



Figura B.25: Regulación térmica y lumínica.

Finalmente la ventana se ha utilizado un cristal especial, que minimiza la entrada de calor por vía solar, especialmente intensa en verano, que a su vez debido también al fuerte reflejo directo continuado por la vista directa al mar, ha obligados a la colocación tanto de un estor difusor de luz, así como una alicantina externa con el objetivo de reducir sustancialmente la intensidad y la dirección lumínica de la luz natural.

B.4.7. Mejoras de Terraza

A continuación se han realizado varias mejoras a la terraza que aunque no tiene conexión directa con la “oficina-casetas” influyen positivamente en la misma.

Una de las grandes desventajas de la terraza es su nulo uso entre junio-septiembre debido a la alta intensidad lumínica y de calor, debido al número de horas de luz directa más el reflejo del mar. Con el objetivo de mejorar su uso, así como reducir la temperatura del suelo y reducir el calor que transmite al ático, se implementado dos toldos corredero de grandes dimensiones, que junto a una vela triangular y una sombrilla, permiten cubrir el 80% de la terraza en sombra las principales horas del día 11-17, así como provocan una zona en sombra tanto a la paredes de la caseta como parcialmente en el tejado de la misma.

Por otra parte se ha renovado el mobiliario exterior, la distribución del mismo, permitiendo salir al exterior de manera cómoda cuando las condiciones climáticas y lumínicas lo permiten.



Figura B.26: Mejoras externas de terraza que afectan a la caseta.



Figura B.27: Mejoras externas de terraza II que afectan a la caseta.

B.5. Evaluaciones, mejoras y correcciones

Por otra parte se han detectado mejoras realizadas sobre el diseño original así como estrategias fallidas durante 2022-23 que han requerido de arreglos o un cambio radical en la solución final.

B.5.1. Humedades y condensación

Aunque el hermetismo del cerramiento es perfecto existen dos fuentes de humedades resueltas parcialmente.

La principal son el alféizar y muros originales del propio cerramiento, aunque están debidamente pintados con pintura transparente anti-humedad, esta capa únicamente limita la salida o la evaporación de la humedad, permitiendo que aquellas semanas con 3-6 días

de lluvias continuadas la humedad progrese por la pared hasta llegar al zócalo por donde aparece en forma de superficie húmeda.

Las juntas entre las baldosas, como se ha explicado el cerramiento se sitúa sobre un punto más bajo que el resto de la terraza, esto junto a la porosidad tanto de baldosas como de juntas, promueve un lento avance de humedades los días de lluvia, y de igual manera cuando supera los 2-3 días comienza no solo a "humedecer" sino a acumular 1-3 mm de agua.

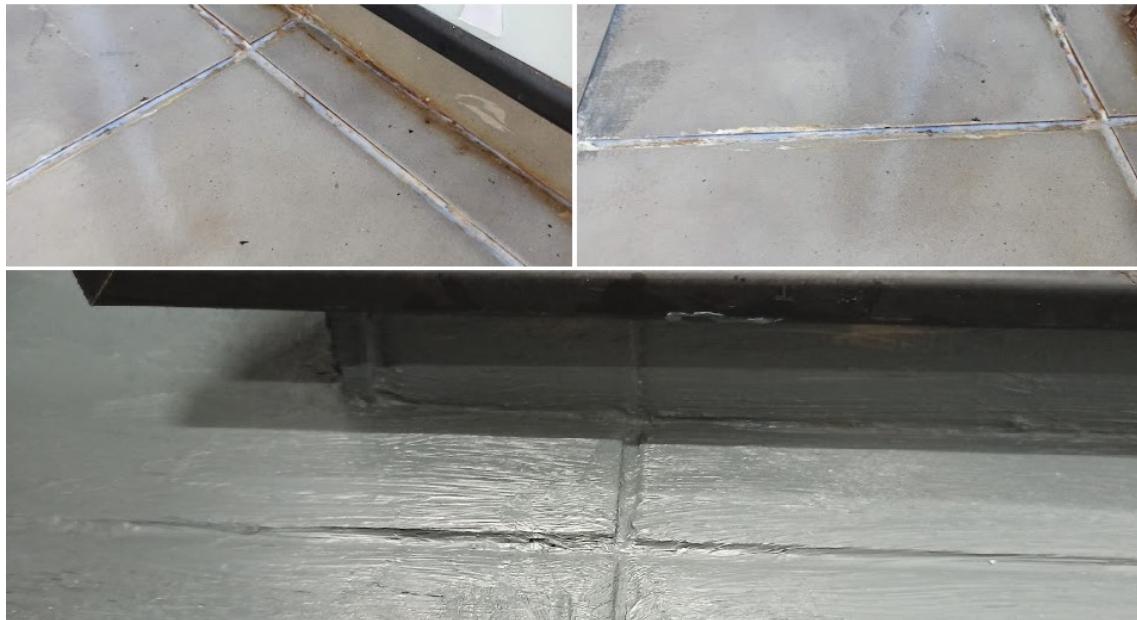


Figura B.28: Impermeabilización con pintura y silicona.

La solución de ambos problemas ha sido múltiples capas de pintura anti-humedad junto a el uso de silicona líquida para sellar completamente tanto suelos como zócalos o posibles entradas de agua por porosidad o capilaridad.

Otro problema recurrente es el propio hermetismo de la sala, si se cierran los orificios de ventilación en invierno con la finalidad de mantener la temperatura. Cualquier humedad interna, especialmente el propio vapor humano, termina condensando en los paneles de aluminio o ventanas, cuya única solución es ventilar intervalos de 3-5 horas, con especial interés previos a la noche.

B.5.2. Bomba de calor/frío

El uso del "pingüino" para calentar, enfriar o des humidificar ha sido un completo fracaso. Aunque la potencia del mismo es para habitaciones de 15 m^2 (ampliamente sobredimensionado), el caudal de aire entrante-saliente necesario para su funcionamiento hace ineficaz su uso debido a las pequeñas dimensiones de la habitación.

Enfría / calienta un aire que se renueva rápidamente 10-20%, es decir, que gran parte del movimiento térmico realizado, se utiliza para re-aclimar el aire nuevo, no permitiendo subir de los 21° en invierno ($12-17^\circ$ exterior), o bajar de 24° en verano ($28-33^\circ$ exterior), a su vez esta renovación de aire trae consigo humedad que el sistema debe retirar, es

decir está continuamente eliminando humedad, que debido a al mal drenaje de la terraza se convierte en un charco o un recipiente a vaciar manualmente.

La reducción u obstrucción de la ventilación para reducir el caudal de aire exterior soluciona parcialmente el problema pero convierte la pequeña y hermética caseta, en una sala de presión negativa, sobre esfuerza el motor que expulsa el aire, estresa la estructura de aluminio y genera “silbidos” por la presión negativa en la puerta corredera.

Por último su mayor problema es que su uso continuado genera excesivo ruido tanto para vídeo conferencias, dificultando la concentración e incluso generando dolor de cabeza. La humedad es un problema especialmente dañino en verano puesto que aunque la temperatura generada dura 90-120 minutos debido al aislamiento, la humedad se re-equilibra con el exterior rápidamente en cuestión de 15 minutos.

Finalmente la realidad ha sido que la desmantelación total de la bomba de calor y el uso de elementos más rudimentarios tales como un calefactor eléctrico tradicional o el uso de la brisa veraniega combinada con ventiladores, satisfacen de una manera más adecuada y sin problemas (ni el gasto eléctrico sobredimensionado).

B.5.3. Paneles solares y SAI

La desaparición de la bomba de calor portátil ha proporcionado un espacio debajo de la mesa útil para la generación de un SAI casero, véase figura B.8 donde se predispone la batería y generador de alterna contiguo a un conjunto de enchufes.

Con el objetivo de disponer de una fuente de energía alternativo para el setup, así como permitir la alimentación tanto de ventiladores, calefactores eléctricos o servidores autocráticos se ha instalado una batería solar de 128 AH a 12 V de ciclos profundo junto con la instalación de paneles solares que alimentan el sistema.



Figura B.29: Paneles solares, tejado y frontal chimenea.

Se ha predisposto de un panel solar de 160 W orientado a sur en el tejado, cuya eficiencia se ve lastrada por la inclinación del mismo (5°), pero permite una captación de luz permanente durante el 80 % de las horas diurnas, así como el reflejo y dispersión atmosférico. Permite captar 0.5-2 A durante las horas mínimas(6-24w), 6-7A al medio día en invierno y 10-11 A en verano, es decir, entorno 80-130W. Así mismo con el objetivo de compensar la horizontal y el cambio de posición solar invierno/verano se ha instalado un

panel de 100W de alta eficiencia en paralelo en orientación sur vertical, que aparte de la luz directa, recibe de manera constante y directa el reflejo marino durante todo el día.

La conjunción de ambos paneles generan un flujo constante y complementario de corriente, compensando tanto la inclinación solar durante las estaciones, como su transito durante el día. Buscando un suministro constante de 30-50W en las peores condiciones y picos de productividad 150-170W durante el medio día soleado.

La principal función de este sistema es un uso constante de los sistemas ambientales, ventiladores en verano (5w, 5w, 25-50w), el uso de calefactores en invierno 150 w, así como el almacenaje de la energía no utilizada para la funcionalidad de SAI, proporcionada por un generador continua-alterna de 1000W, fácilmente comutable para alimentar pantallas y laptop en periodo de 4-6 horas.

B.5.4. Domótica y alarma

Durante 2023 he implementado diversos mecanismo de domótica y seguridad en mi vivienda. Entre ellos la instalación de:

- Sensores de presencia, sensores magnéticos (puertas, ventanas), sensor de humo/gas, sensor temperatura / humedad, detectores de presencia y termostato inteligente.
- Alarma con batería y conexión 3G / wifi, usando aplicación genérica de domótica y API de eventos.
- Actuadores tales como bombillas inteligentes, enchufes inteligentes, regletas y interruptores inteligentes, PIA inteligente, diferencia rearmable.
- Interfaces humanas, smart tv, amazon alexa, google eco.
- Gateways de zigbee, z-wave, wifi, bluetooth y 144 Mhz.
- Cámaras Ip fijas, o con actuadores.
- Raspberry pi con home assitand y API configuradas (google, alexa y gautone).

Aunque son elementos asociados a la vivienda y automatizados en mis cuentas personales, afectan a la oficina automatizado el encendido y apagado de elementos tales como climatización, monitorizando terraza o interior de caseta (cámaras y sensores de temperatura y humedad) así como proveen una seguridad y notificación efectiva del área de trabajo.

APÉNDICE C. SOFTWARE Y NUBE

Este anexo muestra el razonamiento detenido, la comparativa de pruebas de concepto realizadas en el anexo E y configuración dentro del capítulo 2.

C.1. Software

El primer punto que debemos decidir[23] es la modalidad del tipo de servicios que queremos utilizar. Desde el punto de vista técnico, podemos optar entre la externalización o la autogestión, y llegado a un extremo la nube autócrata. Por otra parte, la licencia del software a utilizar así como la flexibilidad, customización y costes asociados difieren entre dos tipos de soluciones “la solución completa” dentro de un marco de trabajo (usualmente privado y de pago) o “soluciones concretas” que se integran con diferentes gestores y se basan en comunidades auto documentadas (generalmente más económica y libre).

C.1.1. Proveedor, mantenimiento y gestión

Externalizar significa utilizar un servicio público proveído por empresas ya sea en modalidad gratuita o de pago. Al ser un producto estándar conlleva ventajas y desventajas, no es customizable, pero facilita su uso y mantenimiento, siendo normalmente transparente al usuario.

Auto gestionar sin embargo indica que aunque sea un producto estándar, la instalación, mantenimiento y solución de incidencias es propia. Requiere de un mayor conocimiento técnico y una dedicación semanal o mensual para tareas de mantenimiento o prevención, sin embargo además de una mayor customización y agilidad con las incidencias, usualmente es mucho más económico en términos generales.

La externalización se puede producir a diferentes niveles, la autogestión usualmente externaliza el hardware/servidor o la gestión de la máquina virtual pero mantiene y gestiona el software instalado. Un usuario puede decidir usar su propio hardware y su conexión a internet para generar una nube autócrata, pero cuya complejidad y mantenimiento a veces no la hace funcional, simplemente autócrata y debido al coste eléctrico asociado lo hace inviable económicamente hablando.

Finalmente se debe entender que a veces un software funciona como un servicio. Gestión de un servicio muchas veces no tiene que ver con el propio software sino con la escalabilidad, backups y transferencia del servicio a otro proveedor, por lo que la autogestión y la externalización parcial deben ser fácilmente intercambiables.

C.1.2. Licencias y tipo de software

Por otra parte, hay que decidir qué aplicativo usamos entre las múltiples posibilidades que ofrecen un resultado similar, pero cuyas características legales y técnicas cambian.

Existen 5 propiedades intrínsecas a todo software:

- Propiedad intelectual.
- Tipo de licencia de uso, distribución y/o modificación.
- Código abierto (publicado) o cerrado.
- Compatibilidad, requisitos o ecosistema integrado.
- Vida útil, refiere al mantenimiento, LTS, actualización del software.

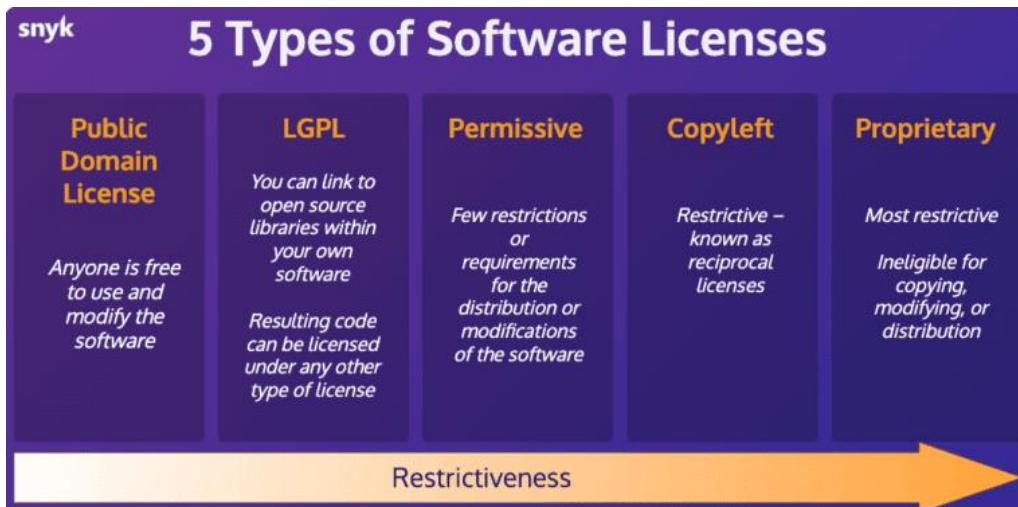


Figura C.1: Tipos de licencias mas comunes[25].

Obviando la casuística concreta de cada software, en una visión global se pueden distinguir dos bloques claros de tipos de software:

“Software privativo”, son aquellos cuyo propietario es una persona o empresa y el uso de dicho software requiere de licencia gratuita o de pago. Así como los requisitos del mismo depende de la compatibilidad realizada por el propietario. Desde el punto de vista técnico, estos software pueden ser abiertos o cerrados indicando si el código que los construye es público/visible o no. Usualmente estos softwares son de pago, o licencias gratuitas bajo restricciones y suelen funcionar bajo entornos de trabajo privativos que unifican un conjunto de aplicaciones como solución interdisciplinaria. Ejemplo entorno de trabajo Microsoft (azure, windows, office, github, atlassian solutions), entornos de trabajo Google(cloud, chromium OS, gmail, maps ...) .

“Software Libre”, es un término que hace referencia a la libertad para usarlo, estudiarlo, distribuirlo, adaptarlo o mejorar el programa. Estos software pueden ser tanto empresariales como realizados por comunidades o programadores independientes, en todos los casos los creadores del software siguen siendo sus propietarios, pero no existe restricciones a su uso o mejora, así como tienen licencias de coste cero. La principal ventaja de estos software son las comunidades y la facilidad de mejora del código al ser abierto. Por otra parte las empresas que colaboran o sustentan estas aplicaciones, ofrecen dichos software a precios competitivos, donde el coste está asociado a la infraestructura, gestión y mantenimiento del software no a la licencia en sí y normalmente permiten la existencia de un mercado de plugins, módulos o configuraciones extras proveídas tanto por empresas como programadores profesionales. Ejemplos: linux, libreoffice y aplicaciones web

o frameworks (ubuntu, wordpress, prestashop, frameworks como laravel, angular, spring boot).

Actualmente existe una mayor popularidad de software libre en servidores, aunque por contra posición el mercado es mayoritariamente privativo en el segmento escritorio. Esto se debe especialmente al SO (Sistema Operativo) empleado en las diferentes plataformas que condicionan los ecosistemas utilizados.

C.2. Elección de Servidor

El primer elemento necesario de nuestra cloud, es el servidor principal. Debido a que nuestro objetivo primario es la instalación de múltiples softwares y la gestión de los mismos, lo más intuitivo es el uso de un servidor o VPS (virtual private server) sin embargo otras opciones como la dockerización de servicios directamente por un proveedor también es una opción valida.

C.2.1. Servidor Físico vs Servidor virtual vs Cloud Externo

Existen dos opciones basadas en servidores físicos, la principal es la contratación a un proveedor de una máquina con características específicas; usualmente este tipo de servidores son de un coste alto y unos requisitos de hardware excesivos para nuestras necesidades. La secundaria el uso de VPS o servidores virtuales dinámicos no sólo permite reducir significativamente su coste, sino que también está asociados a un escalado dinámico (vertical) más flexible y en función de la demanda.

			
Shared Hosting	VPS Hosting	Dedicated Hosting	Cloud Hosting
<p>Pro of Shared Hosting:</p> <ul style="list-style-type: none">✓ Lowest Cost <hr/> <p>Cons of Shared Hosting:</p> <ul style="list-style-type: none">✗ Low Performance✗ Low Security✗ Restricted Configurability✗ Limited Scalability	<p>Pros of VPS Hosting:</p> <ul style="list-style-type: none">✓ High Scalability✓ High Security✓ Some Configurability✓ Mid-Tier Performance <hr/> <p>Con of VPS Hosting:</p> <ul style="list-style-type: none">✗ Mid-Level Cost	<p>Pros of Dedicated Hosting:</p> <ul style="list-style-type: none">✓ Highest Security✓ High Performance✓ Highest Configurability <hr/> <p>Cons of Dedicated Hosting:</p> <ul style="list-style-type: none">✗ Mid-Level Scalability✗ High Cost	<p>Pros of Cloud Hosting:</p> <ul style="list-style-type: none">✓ High Scalability✓ High Performance✓ High Security✓ Some Configurability <hr/> <p>Con of Cloud Hosting:</p> <ul style="list-style-type: none">✗ High Cost

Figura C.2: Tipos de servicios en servidores[154].

La alternativa es el uso de un servidor propio instalado, en la “oficina real”. Obviando el hecho de la compra del hardware o el reuso de un pc para dichos fines, requiere de una conexión a internet alta velocidad y simétrica (fibra) y un consumo eléctrico continuado. Dependiendo del ISP(Internet Service Provider), existirá un problema de IP dinámica, puerto dinámico o servicio de pago para la obtención de IP estática, incluso la no existencia de una ip pública por CG-Nat[26].

Tabla C.1: Tabla comparativa opciones servidor.

Tipo de solución	Ventajas	Desventajas	Coste
Virtual Private server (bajos recursos)	Externalizado, alta disponibilidad, escalaabilidad y migración	Servicios de backup costoso. Recursos limitados	3-15 € / mes
Dedicated Server	Zero problemas técnicos	Coste alto, menos flexible a escalar que VPS	25-300 € / mes
Office Server (pc antiguo, ip dinámica)	Autócrata, uso de ddns para la IP gratuitos. Reciclado Hardware	Disponibilidad no asegurada.	40-70 € / mes (coste eléctrico) 10-20€ / mes (ddns profesional)
Office Server, de bajo consumo (Raspberry pi, nuc) IP dinámica	Autócrata Uso de ddns para la IP gratuitos	Disponibilidad no asegurada. Recursos limitados Especialmente CPU	5-20 € / mes (coste eléctrico) 10-20€ / mes (ddns profesional), compra hardware
Office Server puerto dinámico	Autócrata	Requieren de conexión inversa a través de un enlace exterior Disponibilidad no asegurada.	5-30 € / mes (coste eléctrico)
Office Server IP estática	Autócrata	Ip estática excesivamente cara. Disponibilidad no asegurada.	5-30 € / mes (coste eléctrico) 30-60 € / mes (IP estática)

Analizando los costes de cada una de las posibles opciones tabla C.2.1. se concluye en todos los casos en soluciones **no profesionales** o el **coste no es proporcional al servicio obtenido**. Por lo tanto las opciones más económica, aceptable y elegida es **VPS** con unos recursos asegurados o la contratación de ejecución de contenedores en un cloud. Esta elección facilita un despliegue 'low cost' pero condicionara diferentes elecciones durante todo este documento buscando un uso bajo de recursos.

La contratación de containers en una nube, usualmente es proveída por Amazon, Google permiten un uso de recursos basado en escalado, es decir, un pago por uso. Aunque puede ser muy interesante como plataforma base para desplegar un producto que requiere escalar, **no son los proveedores más económicos** para un uso constante y estático de recursos. Por ello el uso de '**VPS low cost**' permite obtener un servidor base en precios de **3-5€ / mes** que pueden compaginarse con escalados, migraciones o el uso en paralelo de las nubes de contenedores para aquellos servicios o productos que sí lo requieran.

En este caso **se ha seleccionado OVH[27] o time4vps[28] como proveedores**, debido a su gran competitividad en recursos / precio, por los descuentos especiales que se pueden conseguir y ser un proveedor europeo con varias posibles sedes que **cumplen nuestros requisitos legales como proveedor bajo la legislación europea[114]**.

C.2.2. Securización

La seguridad preventiva externa se obtiene exponiendo únicamente el servicio SSH interno del servidor (apropiadamente securizado) y los servicios dockerizados públicos, negando o reaccionando con baneación de IP (black list) a la gran mayoría de ataques reincidentes denegados.

La seguridad preventiva interna se basa en limitar el acceso y traceado. Durante la instalación de docker se suscribe el servicio a un grupo 'docker' guid 997, con el fin de centralizar los permisos del servicio como el sistema de archivos de los volúmenes a utilizar. El acceso, cambio y ejecuciones tanto de 'sudo' como 'docker' y 'docker-compose' pueden ser auditados con logs y notificaciones de las acciones. Y especialmente gestionando y evitando el colapso de recursos como disco, ram o cpu con las estrategias pertinentes.

Finalmente solo el administrador de sistema debe tener acceso directo puesto que el resto de acciones referente a software pueden realizarse mediante ansible ejecutado previamente por in CI/CD o scripts automatizados lanzados por eventos de cron.

La utilización de **redes internas docker**[30] donde suscribir aquellos servicios dockerizados no expuestos directamente, y el uso de vpn-intranets, conectado a estas redes internas permite un acceso directo a los servicios, o una red puente por la que acceder a otros servidores o granjas interconectados.

La implementación de un servicio SSO [39](single sign on) o LDAP[85](Lightweight Directory Access Protocol) como autenticación centralizada de usuarios debe evaluarse si la empresa supera los 10 trabajadores(véase anexo E.1.7. ejemplo), o las rotaciones de personal son altas, pero no son prácticos o conllevan un coste excesivo para nube personales.

C.2.3. Setup y abastecimiento

Con el objetivo de poder migrar fácilmente nuestro vps, así como el mantenimiento del mismo existe una serie de preceptos que debemos seguir:

1. Usar distribuciones **linux LTS de gran soporte comunitario**, en su versión mínima original, es decir, Debian, Redhat y derivados de ambos.
2. **Utilizar scripts de abastecimiento** (instalado actualización y configuración de los servicios bases) que automatizan y aseguran una configuración estandarizada.
3. Seguir una guía de securización del servidor, principalmente basada en una configuración estricta no default que deniega toda petición que no sea estrictamente dedicada y minimizar los servicios expuestos al mínimo necesario.
4. Seguir una guía de buenas prácticas en Linux, creación y configuración de reglas o script que aseguren el nivel de seguridad interna y automatización deseado. Especialmente según sean nuestros requisitos de trazabilidad y auditoría.
5. Realización de backups, que permitan el replicado de nuestro VPS en cuestión de minutos.

Como decisión personal se usará Debian o equivalentes a RedHat (Centos, Rocky Linux o Alma Linux) por ser aquellas distribuciones con menor uso de recursos y más usadas profesionalmente y afines a las dos distribuciones con soporte de pago Ubuntu y Redhat.

La principal tarea del “setup y abastecimiento” es el uso de un usuario, autorizado con privilegiados capaz de ejecutar un script como root para:

1. Actualización y limpieza del sistema.
2. Configuración principal del SO y auto-update, script de mantenimiento y cron asociados.
3. Creación de usuarios y grupos de trabajo.
4. Instalación de herramientas básicas, alias.
5. Configuración de servicio SSH y setup firewall.
6. Instalación de Docker y docker-compose.

Las principales apartados del script de ansible para Setup y abastecimiento son:

Apartado	Acciones
Tareas en Local	<ul style="list-style-type: none">• Bash Script y configuración de ansible (knownhost, actualización de ansible proyecto).• Crea claves asimétricas del VPS RSA, DSA, EC.
Configuración básica de servidor	<ul style="list-style-type: none">• Recolección de datos del server (so, distro, interfaces, ipv4).• Configurar hostname, timezone, file limits, FQDN.• Mensajes personalizados (logging, sudo etc..).
Actualización e instalación esencial	<ul style="list-style-type: none">• Actualización de caché y paquetes SO.• Purgado de paquetes inseguros o no usados.• Instalación de paquetes esenciales.

Creación de usuario 'deployer'	<ul style="list-style-type: none"> • Creación grupo de usuarios de Administración con permisos de sudo. • Añadir usuario “deployer”. • Autorizar claves simétricas creadas en localhost para conexión de usuario en VPS vía SSH server. • Añadir alias, scripts y archivos de usuario.
Servicio Docker	<ul style="list-style-type: none"> • Añadir repositorio de docker para la distribución. • Purga e instalación de paquetes necesarios. • Creación de grupo de usuarios ‘docker’ (997) • Habilitación y arranque de servicio docker.
Securización Básica Exterior	<ul style="list-style-type: none"> • Purgado y des habilitación firewall ufw. • Utilizar el puerto ssh diferente al 22 por 22015(ejemplo). • Deshabilita SSH-password. • Seteo de firewall-d sobre interfaz pública. • Añadir excepción SSH server puerto (22015). • Configurar fail2ban y f2bst para bloqueo de ataques.
Securización Básica Interna	<ul style="list-style-type: none"> • Configuración de ‘sudo’, password timeout, notificaciones grupos y registros auditables. • Scripts de filtrados de logs y reportes. • Limitación de usuarios (disco, permisos etc..)

C.3. Servicios MVP

¿Cuáles son los servicios mínimos que toda empresa debe proveer a sus empleados en remoto? Minimum Viable Products o MVP.

Primero un servicio de comunicación, ya sea mail y/o herramienta de mensajes. Segundo conectividad o web a la que acceder a un mínimo de servicios. Tercero servicio VPN (Virtual Private Network) para acceder a intranets, especialmente si la nube no es pública y queremos tener un acceso rápido, seguro y monitorizable.

Cuarto y quinto una “nube” es decir, servicio de almacenamiento, y servicio de documentación (wiki). Pero hay un elemento administrativo importante, dependiendo del número de empleados, la gestión de los usuarios puede realizarse manualmente o centralizada mente, en aquellos casos con más de 10 trabajadores o rotaciones altas, sexto autenticación centralizada.

Séptimo no hay que olvidar servicios externos de cara a los clientes donde en forma de comunicación tenemos, servicio web, clientes externos (whatsapp, teléfono). Octavo gestión de contraseñas, links, autenticación en clientes de empresa debe almacenarse y gestionarse como un servicio de información accesible por los trabajadores.

Por último, en empresas de software un servicio de repositorios (code, binarios, container) y servicio de CI/CD implementado, así como un servicio de gestión agile-scrum (tickets, versiones, stories etc).

C.3.1. Servicio de Comunicación

Desde mi punto de vista este es un elemento crítico, es decir, no solo de suma importancia sino de mantenimiento y backup. Existen dos vertientes de comunicación, la interna entre los equipos de trabajo y la externa de cara a los clientes. Asimismo tenemos comunicación en tiempo real (chat, videoconferencia) o estática (email).



Figura C.3: Comunicaciones mas utilizadas.

El email, es de suma importancia y de un reducido coste de externalización ya sea en cuenta gratuita (gmail/hotmail) o profesionalizado con tu dominio por ello externalizar el servicio con Don Dominio[40] o equivalentes, es la opción mejor. Para sub-dominios o automatizaciones no expuestos al público puede ser interesante dockerizar internamente véase un servidor de mail[42] así como otras herramientas de mail [41].

La comunicación interna depende de la cantidad de miembros y del tipo de comunicación que puede utilizarse, audio, llamadas, documentos etc.. Usualmente existen múltiples herramientas especializadas en este nicho de mercado especialmente teams[44], slack[43] o ecosistema google.

A su vez existen varias alternativas open source que permiten un uso equivalente o clónico a las mencionadas mediante shelf hosting. Sin embargo en estos casos el montaje, mantenimiento y especialmente volumen de datos y backup de estas herramientas puede llegar ser complejo para más de 10-20 personas.

Por lo que este autor, después de una prueba de concepto basada en Rocket Chat[78], y el uso gratuito de Slack[43] y Telegram[45], **recomienda el uso de herramientas externalizadas en versión gratuitas y la implementación de clones opensource de slack, cuando el número de personas del equipo supera las 15 personas** o la confidencialidad de las tareas así lo requiera.



Figura C.4: Clientes centralizadores de comunicaciones.

Debido a que los clientes son un elemento variable e independiente, muchas veces se necesita tener cuentas en diversos servicios, whatsapp, telegram, teams, zoom, con el fin de tener una comunicación directa y fluida con los clientes a parte del típico formato email o llamada directa. Existen clientes hubs o centralizadores de comunicación, son aplicaciones de terceros que permiten autenticar y añadir multitud de servicios en una única interfaz funcional y simple. Permiten por ejemplo aglutinar en una app, whatsapp, telegram, slack, drive, discord. El ejemplo más conocido es Franz[46] software propietario con su clon opensource Ferdi[47], o Station[48] otro software similar. Este tipo de software es el objetivo de nuestras necesidades por lo que es un elemento indispensable del software local, por lo que seleccionaremos comunicaciones externalizadas con elementos centralizadores como clientes.

C.3.2. Servicio de interconectividad

Después del servicio de comunicación, poderse conectarse adecuadamente entre empleados, herramientas sin ser un blanco publico de ciber-ataques, es usar una intranet, es decir, usar una VPN(virtual private network).

Aparte de las mejoras de seguridad, geo-localización IP, filtrado y monitorización de tráfico, poder interactuar con elementos de intranet (servicios) o otros clientes conectados en formato de LAN, permite el uso de múltiples herramientas no aptas vía internet y ofrece unas mejoras sustanciales para los empleados y minimiza los problemas de configuración o acceso.

El uso de acceso remoto a pc, impresoras, escáneres, ip cams, samba servers, permite en muchas ocasiones flexibilizar el trabajo físico de oficina convirtiéndolo en un remoto parcial, que puede reducir significativamente el número de horas y días físicamente en la oficina, especialmente cuando la pyme tiene una sede física de cara al público.

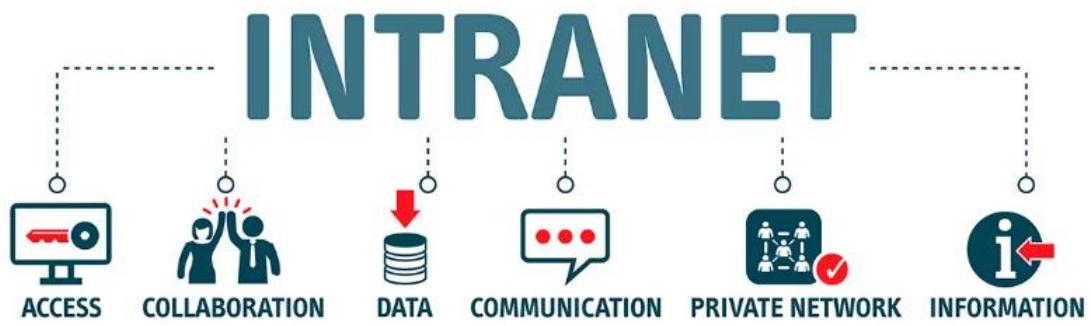


Figura C.5: Intranet ventajas.[\[49\]](#)

Por otra parte en algunos casos puede facilitar la interconexión internacional de diferentes sedes o reducir el coste de servicios internacionalizados a servicios locales por ser conducidos vía la sede internacional más local. Un caso conocido es el uso de centralitas Asterisk[\[50\]](#) para el enrutado telefónico a llamadas locales/nacionales entre países, otras veces servicios de pago pueden ser accesibles desde la intranet como por ejemplo las publicaciones científicas dentro de la red UPC.

C.3.3. Servicio de almacenamiento

Existen múltiples categorías de servicios de almacenamiento. Podemos tener discos o carpetas en red, como un servidor NFS (Network File System) o servidor samba[\[51\]](#) que nos permite acceder, editar o crear ficheros. Normalmente aunque son sencillos, dependen de una instalación previa y no están enfocados a la compartición o acceso remoto, sino al acceso a través de LAN o intranet.

Un servicio de almacenamiento-sincronizado por cliente y con página web como dropbox[\[52\]](#), drive[\[53\]](#), mega[\[54\]](#), etc... que no solo sincroniza múltiples clientes sino que permite compartir ya sea vía cuenta o link carpetas; tener un traceado, historial, restaurar ficheros etc...

Existen multitud de ellos pero la gran mayoría de ellos tiene espacios reducidos o número máximo de clientes que limitan su funcionalidad en pequeñas empresas. Las suscripciones de pago escalan rápidamente o no están ajustadas a las demandas de una pequeña empresa o negocio por lo que no son viables económicamente 10-20 € mes con restricciones de usuarios y espacios de disco sobredimensionados.

Por último existe ecosistema que no solo almacena documentos sino que permiten interactuar con ellos, ya sea modo red social empresarial, otros servicios que acceden a los documentos (office online, drive documents).

Se entiende que un servicio equivalente a dropbox o con ecosistema es lo más apropiado para una pyme, aunque debido a la naturaleza de los recursos del VPS (poco espacio de disco), se debe usar un servidor dedicado a ello, o utilizar un proxy-redireccionador de las peticiones a servidores no limitados en espacio como puede ser una raspberry pi en la oficina o una nube autocrática. En cualquier caso, la combinación de cuentas públicas gratuitas con el servicio interno no es excluyente.

Se ha evaluado seafile[\[55\]](#), owncloud[\[56\]](#) y nextcloud[\[57\]](#), obteniendo un mejor resulta-



Figura C.6: Opensource alternativas a almacenamiento en la nube.

do y satisfacción en nextcloud con una amplia integración de terceras aplicaciones y una amplia aceptación para el uso de pymes por su semejanza a plataformas gratuitas conocidas. Así mismo permite el uso de Collabora^[58] como onlyoffice^[142] y su propia tool; todos herramientas de documentos en linea similar a open365 o drive documents.

Aunque requiere de un uso dedicado de recursos, es factible y económicamente viable el uso de un VPS o servidor autocrático dedicado, ya que ofrece un servicio mucho más completo, no limitado, sobre plataformas de un coste asociado reducido 4-8 € / mes, cuya finalidad no es una carga de trabajo alta, sino un servicio a un grupo reducido de usuarios.

C.3.4. Servicio de documentación

Aunque en muchos casos este servicio puede estar “camuflado” bajo servicio de almacenamiento de fichero o con terceras aplicaciones office-online, entendemos que un verdadero servicio de documentación no solo debe permitir la creación, edición y lectura de documentos sino la búsqueda, versionado e historial de los mismos.

Verdaderamente un servicio fork de wikipedia no son especialmente “densos” ni en configuración ni en recursos, por lo que son una buena práctica a utilizar. Se han evaluado 3 casos, similar a wikipedia^[59] servicio más complejo Bookstackap^[60] similar a confluences o wiki de desarrollo y servicio más ligero y funcional especializados en documentación de código PineDocs^[61] que no permiten busquedas. El resultado es que cualquiera de los 3 son aptos para su uso, y debe ser una cuestión de dinámica de equipo la decisión del uso de uno de ellos o equivalente en función de si queremos una wiki interactiva publica, solo una documentación o una wiki interna.

C.3.5. Servicio de Repositorios y CI

Cuando nos referimos a entornos CI y repositorios, prácticamente la totalidad de softwares existentes pueden ser útiles, sin embargo por similitud a las plataformas privativas con cuentas gratuitas y a su interconectividad con terceras aplicaciones y plugins se han comparado una dockerización de gitlab^[64], bitbucket^[66], gogs^[67] y gitea^[68]. El resultado ha sido que tanto gitlab como gitea son las opciones interdisciplinares y de mayor uso colectivo que son open sources y no tiene limitaciones.



GITEA + DRONE CI

Figura C.7: Gitea junto a drone opción mas común y actual.

Una buena comparativa encontramos en la propia documentación de gitea [62]

Respecto a CI/CD existe un gran ganador durante los últimos 15 años que es Jenkins[69], proyecto open source que tiene integración y documentación para ‘casi todo’ por lo que es el CI de referencia. Con el objetivo de probar otras plataforma se ha usado drone[70], al ser también de uso extensivo, más moderno, pero enfocado a un CI de tamaño más reducido y configurado en los propios proyectos, es decir, que el propio desarrollador interacciona directamente con el CI sin necesidad de tener un administrador dedicado.

C.3.6. Servicio de web externa

La web externa puede ser de dos tipo estática o dinámica, aunque una web estática puede ser de utilidad para únicamente publicitar la empresa, normalmente aunque no tengan un funcionalidad concreta se tiene a usar framework dinámicos que no solo permiten ese dinamismo sino implementan estandarizaciones de temas, plugins y facilitan el mantenimiento y actualización.

En el mundo el 43%[71] de todas las web usan wordpress[72], que es el ganador indiscutible por lo que la recomendación es clara. Sin embargo en aquellos casos de web enfocada a ventas se ha de destacar que prestashop[73] o forks opensource como thirty bees[74] son las maneras más automatizadas, sencillas y económicas de montar una tienda web.

Por último existe una amalgama de frameworks que generan webs estáticas via CI/CD, su principal ventaja es la ligereza, bajos recursos. Su manera de funcionar es similar a un framework dinámico, con la diferencia de que tras la realización de cambios, se compila y genera una nueva versión de la web que el ci/cd actualiza en tiempo real, un ejemplo es Hugo[75].

C.3.7. Servicio de MetaDatos y contraseñas

Existe la necesidad de dentro de un conjunto de personas que trabajan en equipo o personalmente a la hora de gestionar múltiples contraseñas, claves link y otro tipo de meta data, utilizar un gestor. Existen gestores de nota en linea así como gestores de contraseña. Este documento aboga por la conjunción de ‘profiles’ en el navegador asociada a un cuenta

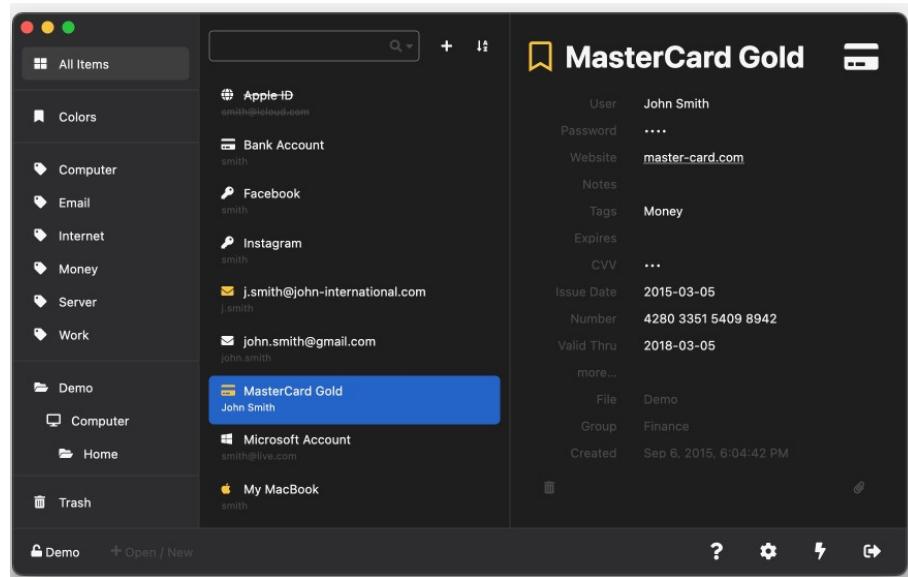


Figura C.8: Gestor de metadatos y contraseñas.

(drive,nextcloud...) que almacena un fichero de meta-data (principalmente contraseñas), y son abiertos por KeeWeb[76] un cliente web, con aplicación de escritorio e integrado en la gran mayoría de navegadores, para acceder a documentos en la nube asociada al profile, y permitir un gestor de contraseñas, meta-data o servicios apropiadamente cifrados por una 'master-password'.

C.3.8. Servicio de Autenticación Centralizada

La autenticación centralizada es un mecanismo no solo de agilidad y facilidad para el cliente sino de gestión interna de usuarios y cambios centralizados de contraseña. Un

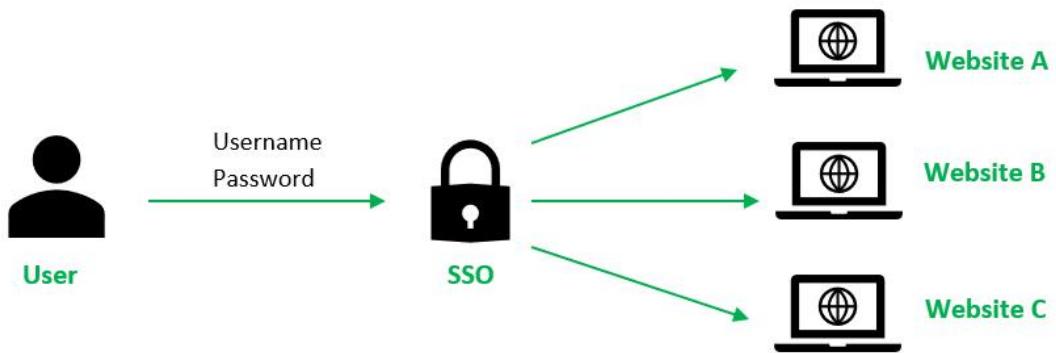


Figura C.9: Single Sign on, diagrama[155].

SSO o Single Sign on (único inicio de sesión) es un mecanismo centralizado de múltiples servicios a través de un elemento centralizador. Usualmente se basa en servicios con autenticación por token a través de una API centralizadora, que genera un token al

logearnos que no solo nos auténtica sino que puede incluir información extra como perfil, permisos o accesos.

Por otra parte abre la puerta a conceptos más extensos como pueden ser el uso de identidades federadas (acceso a sistemas de terceras empresas), Open Id, es decir proveer identidad a través de una URL, OAuth de token con acceso a recursos.

No es el objetivo de este trabajo evaluar la implementación de un SSO, pero sí realizar una prueba de concepto basada en servidor LDAP[85] + Keycloak[86], que permite realizar un ágil SSO para una pyme de más de 10 usuarios. Se ha probado y corroborado la

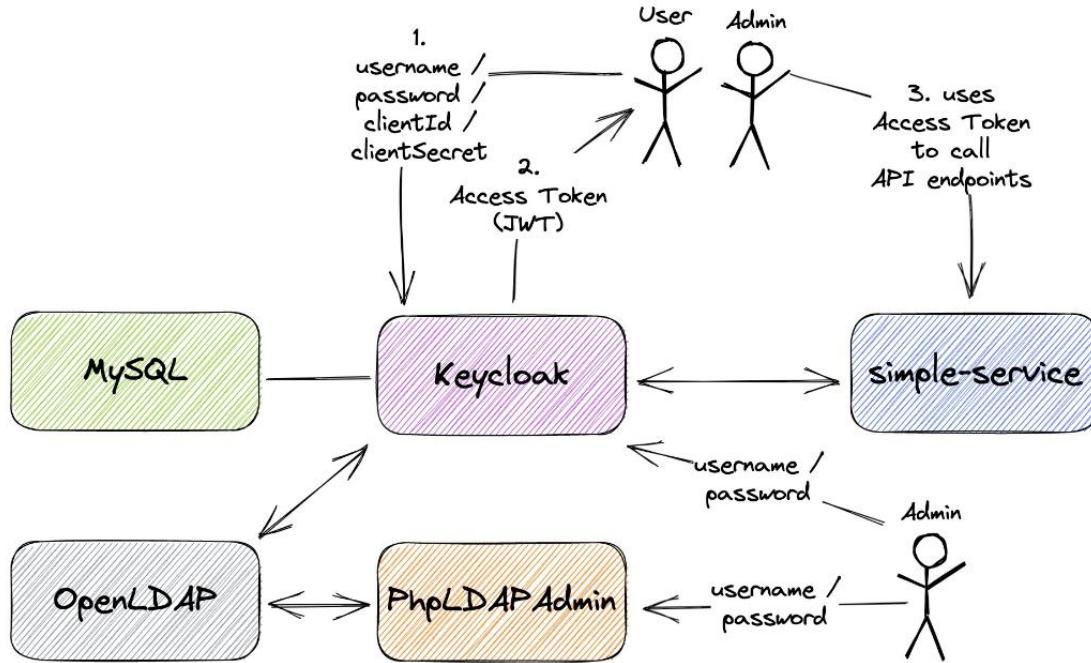


Figura C.10: Prueba de concepto[105] SSO Keycloak con openldap.

prueba de concepto en anexo E.1.7. basado en [105], simple rápida y efectiva de Keycloak + openldap que además provee de un fácil precargado de dominio ldap por fichero y comando así como un acceso UI a través de phpLdapAdmin[106].

Actualmente muchas de las nuevas aplicaciones soportan keycloak de manera nativa a través de uno de su protocolos o vía instalación plugin, por otra parte la fuente openldap usa el protocolo ldap que es antiguo pero también ampliamente usado mecanismo de autenticación. Como conclusión la práctica totalidad de las aplicaciones permiten uno u otro método.

C.4. Tabla de servicios públicos - externos

Aunque se ha predefinido el uso de servicios dockerizados como principal elección, es necesario la evaluación y coste de aquellas herramientas gratuitas o externas dentro de los requisitos más inmediatos de una oficina en la nube. Se han definido una tabla de servicios y proveedores actuales para permitir evaluar su uso y coste.

Tabla C.3: Tabla servicios públicos para externalización.

Tipo de solución	Ventajas	Desventajas	Coste
Correo electrónico	Google / Hotmail	dominio gmail/hot-mail/msn	Gratis
	gmail con dominio	no hay gratuita	5 € cuenta/año
	externalizado	no hay gratuita	1-10 € / año
Comunicación(real time)	Microsoft teams	Limitaciones históricas, plugins. No administración	~5 € / mes por usuario
	Zoom	Max videoconferencia 40 min. 100 Asistentes limitación mensajes.	19€ / mes usuario
	Discord	Limitaciones de ficheros, histórico, enfocado a streaming	9€ / mes usuario
	Slack	Historial (90 días). 10 integraciones, no Admin en canales	6.75€ / mes usuario
	WhatsApp (mobile)	Requiere número móvil, usa Drive o similares como almacenamiento (15 gb)	Gratis
	Telegram (mobile)	Requiere número móvil, no tiene gran mercado en España (como acceso a clientes)	Gratis
Almacenamiento y documentos en la nube	Integradores de mensajería	Franz[46], Rambox, Unipile software que integran desde escritorio múltiples plataformas de mensajería	Gratis
	Google - platform	Gmail o integración de tu mail con gmail	5.75€ / mes usuario
	Microsoft - platform	Outlook (hotmail) o integración con tu mail	5.63 € / mes usuario
Otras nubes	NextCloud[57]	Tecnología open-source.	3€ / mes usuario
	Dropbox[52]	Espacio limitados.	100€ / año

	Owncloud[56]	Tecnología open source community y versión enterprise licenciada	Solo grandes empresas
VPN o intranets	Múltiples servicios locales no hay proveedores globales estandarizados	Se centran en intranets y redes sociales internas. No facilitan la integración en redes sino el uso online de una intranet a través de sus herramientas.	5-10 € / mes user
	Solo Redes	No focalizados en empresas	3-6 € / mes
Repositories y entornos CI/CD. Comparativa [77]	Github[65]	Limitación de proyectos privados y acciones ci	5€ / mes (equipo 2GB)
	Gitlab[64]	Limitación de proyectos privados y acciones ci	5€ / mes full supported con 2000 ci/cd
	Bitbucket[66]	Limitación de proyectos privados y acciones ci	4€ / mes hasta 5 usuarios
Web Service	Prestashop[73]	hospedados o autogestionados	10-20 € / mes
	Other custom	solo páginas webs creadas por cms	10-15 € / mes
	Wordpress[72]	plugins y gestión limitada	4€ / mes

Todos los proveedores de herramientas **fidelizan** tanto a sus ecosistemas como evitan la interoperabilidad con otros, así mismo las mejores plataformas (microsoft y google) proveen de un servicio gratuito algo inferior al requerido por una pyme y a su vez no es económicamente viable usar planes business para grupos de trabajo inferiores a las 50 personas.

La realidad es que si el ecosistema completo puede ser “barato”, una pyme no usa el 100% de lo ofertado y el precio del pack es desproporcionado a las necesidades. Por lo tanto la implementación de soluciones opensource es la única solución que cumple nuestros requisitos de coste reducido, especialmente aquellas referidas con almacenamiento de espacio, code o intranets (vpn y definición de redes) a la vez que no excluye el uso gratuito de plataformas propietarias, como la comunicación.

C.5. Servicios Dockerizados Shelf Host

Para cada una de las necesidades virtuales existe una amalgama de servicios que suple dicha necesidad, especialmente en entornos open source. En este apartado se proponen diferentes servicios dockerizables como soluciones factibles a las necesidades más prioritarias o habituales.

En muchos de los casos existen múltiples alternativas, por lo que aquí se representan únicamente los más soportados y utilizados. En la siguiente tabla se muestran las soluciones factibles, en las cuales únicamente se han realizado pruebas de concepto en las señaladas con un '*'. Se ha realizado la prueba de concepto aisladamente (no integrados), que permite evaluar el software y su adecuación a nuestro MVP. Verificando su funcionamientos, complejidad al instalarlo y configurarlo y recursos y agilidad de uso en el VPS.

Tabla C.4: Tabla servicios dockerizados.

Tipo de Servicio	Herramienta	Detalles
Chat / Vídeo conferencia	Rocket chat[78] *	Alternativa completa a slack
	Tinode [79]	Alternativa opensource
	Zulip [80]	Open source similar slack
	Element [81]	Open source alternativa a discord
	Wire [82]	Mix de whatsapp, Telegram y signal, open source extremadamente seguro.
	Mattermost [83]	Alternativa a slack y teams open source con integraciones de otros softwares
Correo electrónico	Docker mail server[42]	Complex but completed
	Mailu[84]*	Fácil y completo server
Autenticación Centralizada	LDAP[85]*	Implementación de openldap, automatizadas por variables
	Keycloak[86]*	Federativo y contable a un ldap.
Almacenamiento y documentos en la nube	Samba[51]*	Servidor de ficheros (carpetas en red)
	Seafile[55]*	Sincronización y cifrado de ficheros, simple y seguro.
	owncloud[56]*	Clon de dropbox con add-ons, apps e integración en otros softwares
	nextcloud[57]*	Clon open source owncloud con mejoras y más apps. Pseudo red social y confederado.
Servicio de Documentación	Bookstackapp[60]*	Similar a confluence, más pesada.
	Mediawiki[59]*	Equivalente a wikipedia
	PineDocs[61]*	Simple ligera, perfecta para documentaciones pero no es una wiki no hay búsqueda.
VPN o intranets	openvpn[87]*	Estándar open source actual. Complejo en cierta medida

	wireguard[88]*	Novedoso, más fácil de usar, mejor performance
	ipsec[89]*	estático y tiene problema con firewalls.
	pritunl[90]*	Servidor y cliente vpn que combina múltiples protocolos incluyendo openvpn y wireguard.
Repositories	gogs[67]*	Servicio Git auto hospedado simple, estable y extensible, open source.
	gitea[68]*	Basado en Gogs, incluye mejoras e integraciones con terceras aplicaciones que lo equivalen a github o gitlab.
	gitlab[64]*/bitbucket[66]*	tienen versiones self hosted limitadas o licenciadas.
CI/CD	jenkins[69]*	Ampliamente extendido, pero no focalizado a dockerización
	drone[70]*	especialmente enfocado a dockerización
Web Service	wordpress[72]*	Servicio mas extendido de web (psudo dinámica)
	prestashop[73]* / bee[74]*	Servicio de gran calidad para tiendas online
	framework MVC a medida*	Permite la creación de web dinámicas customizadas.
	estática o framework compilado[75]	Fuente eficiente de web estáticas.
Ticketing	taiga[92]*	Web service para Scrum-agile.
	mattermost[83]*	Gestión de comunicación y ticketing en uno. consume bastante recursos.
	planka[93]*	Simple Web service para Scrum-agile.
DNS proxy	mageddo[99]*	Simple dns relay and proxy, configurable por json y interfaz gráfica.
	simple go dns proxy[100]*	Ejemplo de prueba de servidor dns proxy escrito en go con relay basado en docker dns.

También existe una amalgama de servicios no incluidos en el MVP conceptual pero de mayor necesidad como herramientas auxiliares para gestionar nuestra nube véase [C.5](#)

Por último cabe destacar que cualquier elemento de una red empresarial tales como firewall, dhcp server, etc... todos son susceptibles de ser dockerizados e integrados en intranet, especialmente se ha hecho un uso intensivo de pruebas con [nginx\[124\]](#) como proxy y [traefik\[123\]](#), así como generadores de autoridades de certificación o [Let'sencrypt\[122\]](#).

Tabla C.5: Tabla servicios dockerizados Extras.

Tipo de Servicio	Herramienta	Detalles
Gestor de contraseñas	KeeWeb[76]*/ Passbolt[94]	Un uso muy practico de gestión de contraseñas.
Docker UI	Portainer[95]*	Un web service de despliegue de contenedores mediante UI.
VPN UI	wireguard ui[96]*	Practico y eficiente UI para gestionar wireguard.
	wg-easy[97]*	Sencillo UI para gestionar wi-reguard.
Monitorización y bloqueo	adguard[98] home	Servicio de bloqueo de anuncios y spam que protege tu privacidad.
Hub de Chat o comunicaciones	Ferdi[47]* / Franz[46] *	Permite la aglutinación de los servicios más populares.
	Station[48]	Permite la aglutinación de los servicios más populares.

C.6. Contenedores y relaciones extendidas

Docker compose es una herramienta con una gran cantidad de detalles que permiten no solo agrupar y escribir en un lenguaje mas humano comando docker, sino permite setear relaciones y limitaciones de interés, tales como:

- Comandos docker y sus opciones, puertos, volúmenes, variables de environment, creación e inclusión de redes etc..
- Dependencias de los containers especialmente a la hora de arranque, política de reinicio de containers, generación de estados o helthycheck points.
- Definición y restricción de recursos hardware.
- Imágenes, construcción de imágenes, argumentos (docker build).
- Labels o propiedades auto descriptivas o usadas como variables finales.
- Kernel parameters, para ejecutar en el container.

Por otra parte existe varias versiones de docker-compose format, existe V2[127] antigua y centrada en únicamente un host, y V3[128] una versión mas reciente con sintaxis compatible con docker-swarm que permite escalas de un único host a un cluster swarm con el mismo documento y sintaxis.

Entre otras estrategias, se puedes sobre escribir archivos o parte de los ficheros o incluir sub-ficheros para generar un docker-compose completo.

C.6.1. Docker multi-capa

¿Se pueden tener un container docker con servicio docker? ¿qué es este concepto? Por definición docker ejecuta los contenedores directamente en el sistema hospedante, pero aislando los mismos para que desde dentro sea completamente indiferente. Sin embargo si intentamos instalar docker dentro de un container docker, dicho aislamiento inhabilita la ejecución de docker ya que no tiene un control total del kernel ni de los recursos reales del servidor hostpedante.

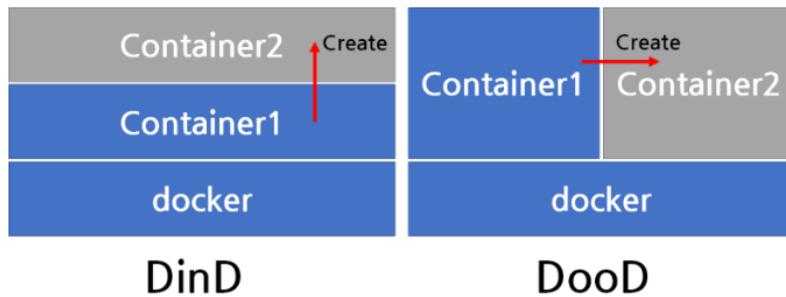


Figura C.11: Docker in docker vs Docker out of docker.

Existe tres soluciones, la denominada 'docker in docker' [32], la denominada 'docker out of docker' [33] y la utilización de frameworks especializados como sysbox[35]. En ambos casos existe una escalación o acceso a privilegios no ordinarios en un container, por lo que siempre son un posible fuente de problemas de seguridad.

Las ventajas de esta complejidad son dos, permitir una realimentación negativa de los containers y el hosts (empleada en este documento) o la creación de diferentes niveles de docker ejecutándose en paralelo en el host (de gran utilidad en CI/CD).

C.6.1.1. Docker in docker (dind)

Docker in docker o contenedores dind[32], sirven para poder ejecutar docker dentro de un container. Por definición el hilo principal debe ser 'root' real para poder salir del aislamiento, lo que implica un escalado de privilegios y un gran agujero de seguridad.

Con el objetivo de no comprometer el sistema hospedante vía un container con dind :

- Se genera un nuevo demonio-servicio de docker, que se ejecuta exclusivamente dentro del container, por lo que no es accesible desde el sistema hospedante.
- Se define un nuevo usuario 'privileged' muy similar a root, que re-mapea al usuario root dentro del container. Este usuario tiene aparente root-access dentro del container, pero es un usuario con menos permisos que root del sistema hospedante, para permitir la compatibilidad se habilita un flag que permite en casos señalados escapar de su entorno namespace, para poder acceder a ciertos elementos del sistema hospedante especialmente en términos de hardware como redes, devices y particiones.

En ambos casos, los contenedores siguen ejecutándose desde el sistema hospedante, pero aquellos creados por dind, su administración se realizada por el daemon incluido en

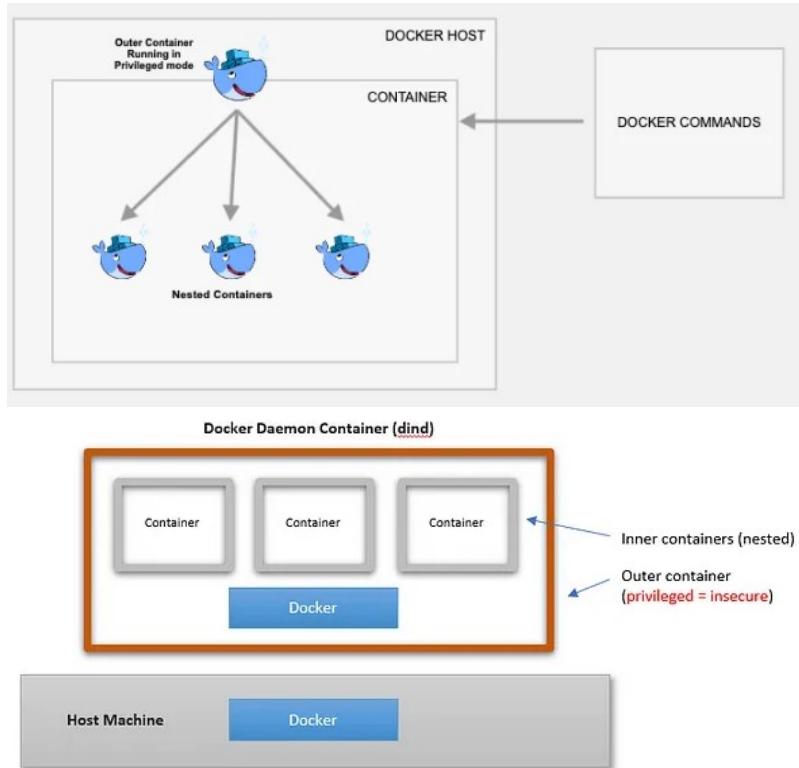


Figura C.12: Docker in docker diagrama de ejecución de comandos[156].

el dind container. Y siguiendo el árbol de jerarquía de procesos parando o eliminando dicho container también se acaba con todo sus subprocessos incluyendo estos contenedores adicionales no gestionados por el sistema hospedante.

La principal utilidad es la creación de contenedores temporales, especialmente en fines de CI/CD o para la ejecución de comandos vía container directamente desde dentro de un contenedor. La principal desventaja es la duplicación de espacio para la gestión de imágenes docker, al desacoplar la gestión del demonio y los posibles de ataques de seguridad directos al hardware.

C.6.1.2. *Docker out of Docker (dood)*

Una alternativa similar es el acceso del propio daemon docker del sistema hospedante dentro de un contenedor[33] y por consiguiente un acceso a la monitorización, creación o parada de contenedores, desde un propio contenedor en ejecución. De una manera similar esta realimentación negativa, debe ser controlada para evitar inestabilidades o fallos de seguridad, utilizando softwares como docker-gen[34], que se focalizan en las notificaciones y monitorización. Se monta en el volumen el socket del daemon de docker hospedante, sobre el mismo daemon docker en el container. Como resultado el contenedor puede acceder vía socket-daemon a información ejecutando comandos, sin incidir en riesgos de seguridad como dind.

Sus principales aplicaciones son notificaciones o monitorizaciones que automatizan muchos procesos, como son en este trabajo los proxy revers, dns-proxy automatizados y los certificados 'let's encrypt', en todos los casos un contenedor 'escanea' a sus contenedo-

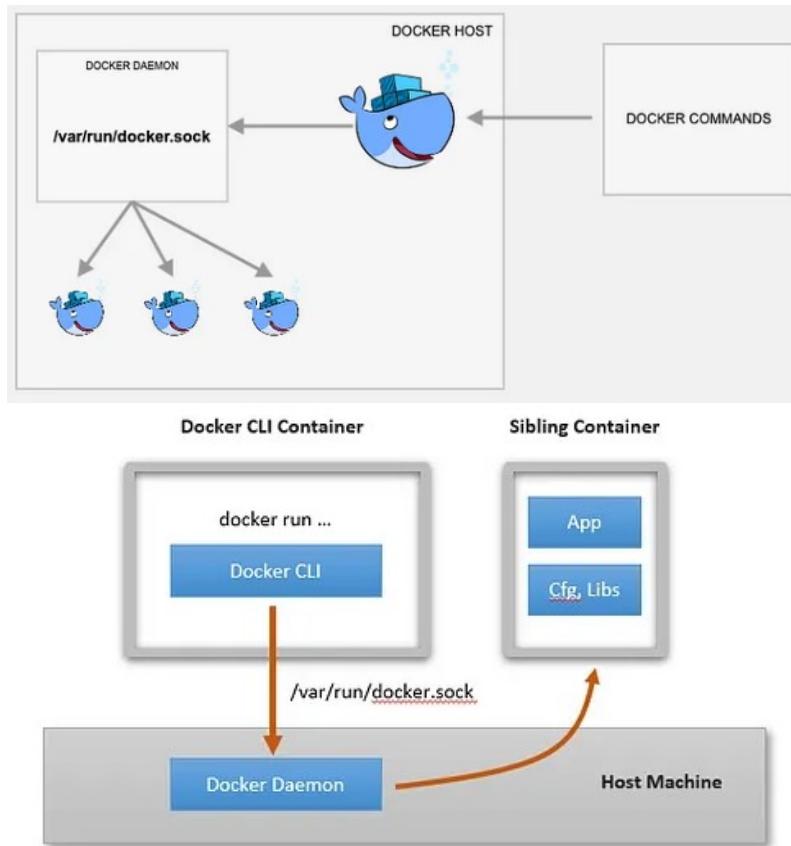


Figura C.13: Docker out of docker diagrama de ejecución de comandos[156].

res hermanos en ejecución en paralelo y permite el acceso a datos tales como network, hostname, labels, aliases, estatus or helthycheck. En base a la información recopilada, ejecuta o genera configuraciones 'automaticas', que permiten realizar los certificados para los hostnames declarados, la redirección proxy a los mismos o el indexado dentro de nuestras rutas locales dns.

C.6.1.3. *Sysbox o equivalentes*

Existen frameworks de trabajo como Sysbox[35] que proporcionan un entorno de ejecución equivalente a dind, pero sin la necesidad de utilizar los flag de privilegios.

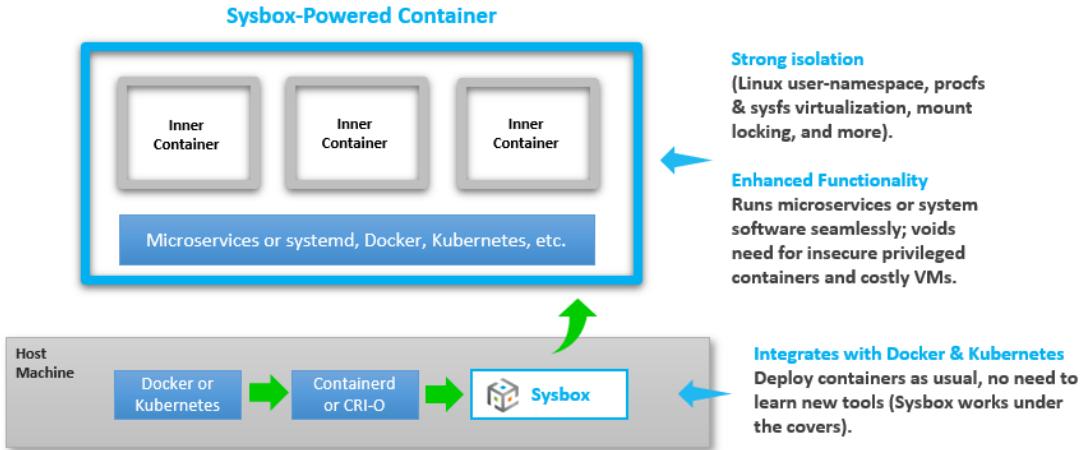


Figura C.14: Sysbox [35] diagrama de funcionamiento.

Esta es la solución más innovadora y relevante, ya que la misma compañía de docker ha comprado a sysbox el pasado año. Sin embargo muchos de los imágenes docker utilizadas en este documento requieren de un soporte comunitario para su mantenimiento, que aun no ha sido migrado a ejecuciones bajo sysbox.

C.6.2. SystemD service con docker-compose

Una vez que tenemos un grupo de servicios definidos, automatizados dentro de un docker-compose con ciertos scripts, podemos decir que tenemos unos 'servicios dockerizados' en funcionamiento. Sin embargo para el sistema hospedante no dejan de ser unos procesos arbitrarios como comandos que podemos ejecutar dentro de una terminal. Cuando reiniciamos el servidor o se enciende después de un apagado dichos procesos han muerto y sido eliminados.

Definir dichos servicio/servicios como un servicio o modulo de systemD[107] no solo aporta valor de cara a estandarizar dentro del sistema linux dichos servicios sino que permite automatizar, reinicio, modos de mantenimiento y arranque del servicio de una manera mas automática y estándar.

Se puede optar por un script genérico que ejecuta los servicios de docker desde un modulo de systemD (customizado), basado en argumento '%i' pasado al modulo para ejecutarse sobre diferentes sub-carpetas cada una con el docker-compose del servicio (véase C.1).

```
[Unit]
Description=%i service with docker compose
PartOf=docker.service
After=docker.service

[Service]
Type=oneshot
RemainAfterExit=true
WorkingDirectory=/yourFolderwithService/group_services/%i
ExecStartPre=/usr/local/bin/docker-compose pull --quiet --parallel
```

```

ExecStart=/usr/local/bin/docker-compose up -d
ExecStop=/usr/local/bin/docker-compose down
ExecReload=/usr/local/bin/docker-compose pull --quiet --parallel
ExecReload=/usr/local/bin/docker-compose up -d

[Install]
WantedBy=multi-user.target

```

Code C.1: SystemD /etc/systemd/system/docker-compose@.service

Por otra parte también se pueden automatizar tanto la actualización de imágenes de docker (al basarse en LTS continuamente mantenidos por su comunidad), llamando a un 'Reload' del servicio. Como alternativa definir un servicio customizado de SystemD por cada servicio o agrupación de servicios apuntando a un docker-compose o un script customizado (véase C.2). Ambas opciones pueden ser puestas en marcha a la vez, sin embargo hay que tener en cuenta la interacción entre las mismas.

```

[Unit]
Description=docekrized services manage by scripting and docker-compose
PartOf=docker.service
After=docker.service

[Service]
Type=oneshot
RemainAfterExit=true
WorkingDirectory=/yourFolderwithYourScripts
ExecStart=./start.sh
ExecStop=./stop.sh

[Install]
WantedBy=multi-user.target

```

Code C.2: SystemD /etc/systemd/system/custom-service

Por ultimo remarcar que la ejecución de ciertas tareas puede automatizarse mediante 'cron' C.3 o directamente usando systemD C.4 C.5

```
0 4 * * * root /bin/systemctl reload docker-compose@*.service
```

Code C.3: Crone line /etc/crontab, actualizacion imagenes

```

[Unit]
Description=Refresh images and update containers
Requires=docker-compose.service
After=docker-compose.service

[Timer]
OnCalendar=*:0/15

[Install]
WantedBy=timers.target

```

Code C.4: SystemD /etc/systemd/system/docker-compose-reload.timer

```

[Unit]
Description=Refresh images and update containers

[Service]
Type=oneshot

```

```
ExecStart=/bin/systemctl reload docker-compose@*.service
```

Code C.5: SystemD /etc/systemd/system/docker-compose.reload.service

Utilizando una aproximación similar podemos centralizar los mecanismos de mantenimiento o backup de los diferentes servicios dockerizados via 'cron-script' o 'systemD service-timer'.

Por ultimo si se desea poder acceder a los logs de los nuevos servicios systemD[107] desde la plataforma de 'journald'[108] se debe indicar en la configuración de docker el driver de log apropiado:

```
{  
    "log-driver": "journald"  
}
```

Code C.6: Fichero de configuracion daemon.json

C.6.3. Backups y restauración de servicios

Con el fin de realizar los backups se asumido una gestión por scripting o systemD, el principal objetivo es mediante un timer o cron que ejecuta un proceso de apagado de servicios dockerizados y realizar el backup a una horas poco usuales, como puede ser 2-4 de la noche. Como existe una limitación de espacio y de CPU en el VPS, pero no tenemos

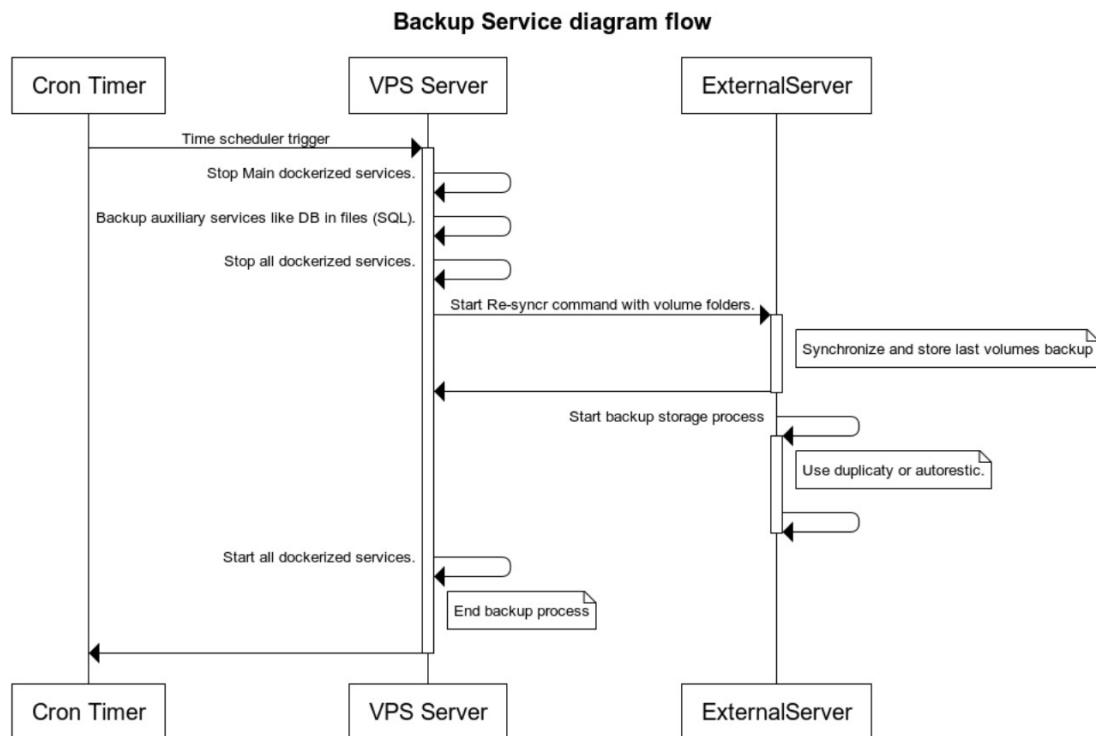


Figura C.15: Backup diagrama de secuencia.

trafico intenso en las VPS o el VPS durante el periodo nocturno la solución ha sido externalizar el proceso de backup, entendido como almacenamiento de versiones con historial

a un servidor externo, que puede perfectamente pertenecer a la VPN (ser autócrata). Por lo tanto el proceso de backup únicamente realiza una sincronización de carpetas y ficheros con el servidor externo, es decir, en muchos casos servicios que no realizan cambios de manera continuada como el VPN server, pueden ser sincronizados sin necesidad de apagado. Por otra parte es en el servidor externo donde se configura apropiadamente la gestión de backups, donde no tenemos limitaciones de CPU o espacio en disco y se utilizarán herramientas que permiten el almacenamiento de un histórico basados en tags o identificadores y en diferencias entre versiones de una manera similar a git. El objetivo se basa en como en git, permitir un almacenaje reducido, poder acceder a versiones específicas por fecha o identificador, añadir una capa de redundancia, checksum o incluso cifrado al almacenamiento de los backups. Por otra parte la lógica de restauración funcio-

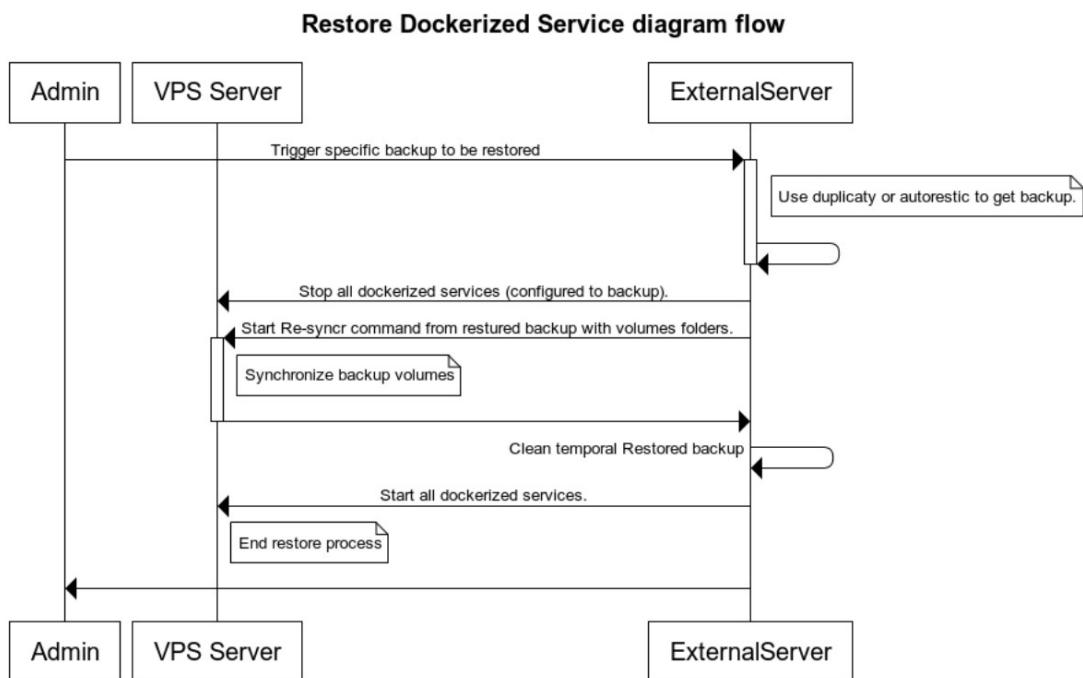


Figura C.16: Restore diagrama de secuencia.

na de manera inversa, puesto que una restauración no es automática, sino que es iniciada por el administrador, inicializa el servidor externo para preparar el backup específico. Una vez generado el backup requerido se procede a la parada de los servicios docker en el VPS y una posterior sincronización inversa servidor externo - VPS para finalmente volver a arrancar los servicios véase figura

C.6.4. Estructura de ficheros

La estructura de ficheros y docker-compose debe ser coherente y ágil para realizar backups, declarar servicios systemD o permitir el arranque, parada o actualización local de grupos de containers. Se han probado diferentes soluciones durante la realización de este trabajo, aunque no se ha definido 'la mejor manera de trabajar' y simplemente queda mostrada varios casos para que el administrador use la más adecuada a sus necesidades.

Por una parte es importante definir una carpeta principal sobre la que trabajar, en esta car-

pta contiene un docker-compose general con todos los servicios, carpetas de volúmenes compartidos por uno o mas servicios, o contraseñas token segurizados correctamente mediante secrets. Debido al gran numero de servicios, redes y otras propiedades, es probable que se separen diferentes grupos de contenedores, en base a servicios principales y auxiliares de estos servicios principales (figura C.17).

```

. ServiceA/
    ├── docker-compose.yml -> allow to manage directly the service alone, just with auxiliary containers, import other files.
    ├── serviceA-compose.yml -> define main service A, it is included by docker-compose files (service or global)
    ├── auxiliaryA-compose.yml -> define auxiliary service to A
    ├── backup.sh -> task to perfrom backup in A service
    ├── restore.sh -> task to perfrom restore in A service
    └── volumes/ -> volumes related with A/
        ├── volumeA1
        ├── volumeA2
        └── volumeA3

```

Figura C.17: Estructura de un Servicio A.

```

. DockerizedFolder/
    ├── docker-compose.yml
    ├── network-compose.yml
    ├── backupScript.sh
    ├── restoreScript.sh
    ├── secrestFolder/
    │   └── secretFile
    ├── shared-volumes/
    │   ├── volumS1
    │   └── volumS2
    ├── ServiceA/
    │   ├── docker-compose.yml
    │   ├── serviceA-compose.yml
    │   ├── auxiliaryA-compose.yml
    │   ├── backup.sh
    │   ├── restore.sh
    │   └── volumes/
    │       ├── volumeA1
    │       ├── volumeA2
    │       └── volumeA3
    ├── ServiceB/
    │   ├── docker-compose.yml
    │   ├── serviceB-compose.yml
    │   ├── auxiliaryB-compose.yml
    │   ├── backup.sh
    │   ├── restore.sh
    │   └── volumes/
    │       ├── volumeB1
    │       ├── volumeB2
    │       └── volumeB3
    └── ServiceC/
        ├── docker-compose.yml
        ├── serviceB-compose.yml
        ├── auxiliaryB-compose.yml
        ├── backup.sh
        └── restore.sh

```

```

include:
  - ../commons/compose.yaml
  - ../another_domain/compose.yaml

services:
  webapp:
    depends_on:
      - included-service # defined by another_domain

services:
  web:
    extends:
      file: common-services.yml
      service: webapp
    environment:
      - DEBUG=1
      cpu_shares: 5
    depends_on:
      - db
  db:
    image: postgres

```

```

$ docker compose -f compose.yml -f compose.prod.yml
1  ROOT_DIR := $(dir $(abspath $(lastword $(MAKEFILE_LIST)))
2  SERVICE := diun
3  include ${ROOT_DIR}/../../core/common.mk
4  include .env
5
6  .ONESHELL:
7
8  .PHONY: install
9  install: ## Start all containers in background
10  @$(DOCKER_COMPOSE) up -d
11
12  .PHONY: up

```

Figura C.18: Estructura completa y ejemplos de include, merge o makeFile.

Para la gestión de los servicios se puede optar por un script 'services.sh' o como se ha indicado en [C.6.2.](#) un servicio SystemD, en ambos casos la estructura de subcarpetas por servicio principal, nos permite auto-gestionar por argumento '%i' el servicio en cuestión, o aplicar comandos al conjunto de ellos ('i = *').

Utilizar las funcionalidades de 'include'[[129](#)] y 'merge'[[130](#)] puede permitirnos definir en diferentes fichero aquellos elementos globales (redes, volumen compartido), como los servicios principales y sus auxiliares, permitiendo definir las funcionalidades en un único fichero que es posteriormente incluido en un docker-compose.

El concepto es permitir gestionar autonómicamente el servicio principal, ya sea desde su docker-compose desde la sub-carpetas ServicioA, o directamente desde el conglomerado de servicios de la carpeta principal.

Por ultimo, existen los script de 'backup.sh' y 'restore.sh' realizar dichas tarea de manera reiterativa sobre los subfolders y scripts de cada servicio. O por una temática similar a bash script, archivos makeFile, con comandos de instalación, arranque, parada, backup y restauración, equivalentes muy similar.

Se tiene que comprender, que dependiendo de las necesidades y especialmente del numero de servicios, una nube mas compleja, requerirá de una segmentación mas clara, y una estructura jerárquica mas definida. Por ejemplo podemos definir nuestros servicios dockerizados tal y como muestra el ejemplo, e incluirlo en un proyecto de git, que excluya aquellos volúmenes variables (con datos), y folders sensibles como secrets. De esta manera podemos guardar nuestra 'backup de configuración' sin datos asociados a un proyecto fácilmente desplegable vía ansible.

Otro ejemplo puede ser similar a mars-server[[131](#)], pre-configurar el caso mas complejo, y únicamente habilitar o 'instalar' aquellos servicios que realmente deseemos utilizar dentro del servidor.

En todo caso la opción mas apropiada seria la generación de un aplicativo capaz de agrupar las diferentes pruebas de concepto de este trabajo, y generar dinámicamente vía ui, los docker-compose o ficheros a utilizar, así como una gestión mas intuitiva. Desgraciadamente debido a limitaciones de recursos y tiempo no ha sido objetivo dentro de este documento, aunque probablemente se materialice durante 2024.

APÉNDICE D. REDES

Este anexo muestra el razonamiento y pruebas de concepto realizadas durante el tema 3 de redes.

D.1. Conceptos básicos

Para entender la naturaleza de las redes informáticas así como el servicio que brinda una VPN, inicialmente hay que tener un contexto y conceptos básicos. Una red es un conjunto de dispositivos informáticos interconectados por medio físico o inalámbrico. Para la apropiada conexión y gestión de las comunicaciones entre los dispositivos existe una infraestructura, cables, routers, switch, antenas así como identificadores y software interrelacionados.

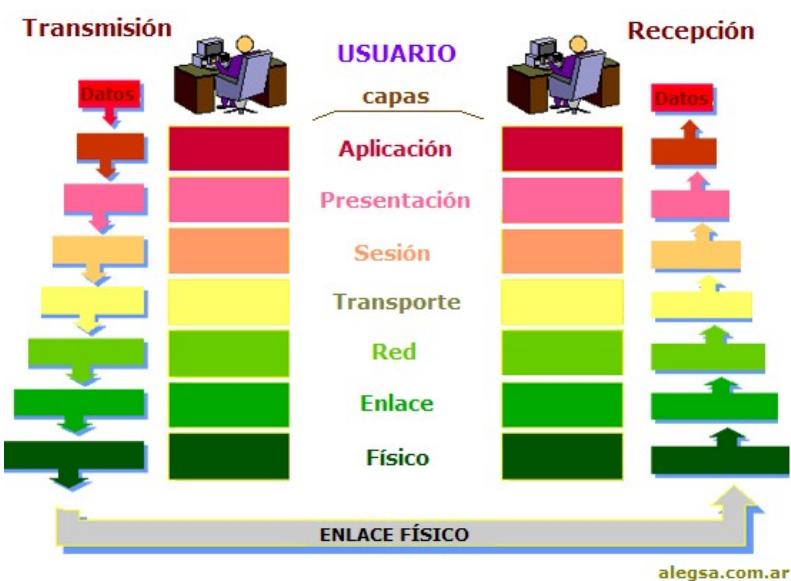


Figura D.1: Diagrama capa OSI [120].

Para entender la comunicaciones dentro de una red, se necesita el modelo de la capa OSI[120] así como nociones básicas de la capa física (cables, antenas, hubs, ethernet) y de la capa de enlace (L2) como protocolos arp, icmp e identificadores MAC.

Podemos resumir brevemente que la capa L1 físico, es propiamente el hardware y la manera 'física' de comunicarse a través de señales electromagnéticas, procesar las señales y software-drivers de dicho hardware.

La capa L2 o Enlace, se refiere al conjunto de protocolos, identificadores que permiten la comunicación y repetición de comunicaciones de elementos conectados físicamente.

La capa L3 o de red, son los protocolos, identificadores y funcionalidades (enrutar, nat, dns ...) que auxilian, permiten y realizan el envío de paquetes a través de una red.

La capa L4 o transporte son principalmente protocolos de transporte, permiten verificar y optimizar la transmisión de paquetes entre elementos de la red.

El resto de capas superiores se focalizan en la seguridad, protocolos de aplicación o los datos intrínsecos.

Cuando hablamos de transmisión de datos en una red, hablamos de 'paquetes' es decir conjuntos organizados de datos. Estos paquetes se organizan a capas siguiendo las funciones de la pila OSI, por lo tanto para cada nivel existen una cabeceras y una carga útil de datos que contiene niveles superiores.

Un punto importante a entender es que no existe una única red, sino multitud de redes, las cuales están interconectadas entre si. A el conjunto de redes publicas interconectadas entre si es lo que hoy en día llamamos Internet, a los subconjuntos de redes privadas no publicas se les denomina intranet o red privada, y aquellas mono-redes privas locales que disponemos en nuestro router de casa se les denomina LANs (Local Area Network).

Es importante entender que el principal objetivos de este trabajo en temática de redes es configurar adecuadamente nuestras LANs en casa en conjunción con una red privada virtual (VPN) de carácter profesional, o la interconexión de LANs mediante VPN.

D.1.1. VPN y encapsulado

Una VPN es una red privada, virtual, es decir no tiene capa física real sino que es emulada. Para emular dicha capa se puede utilizar opciones de hardware específicas de fabricantes o un software. Nuestro caso software implica que usaremos una red real, sobre la cual se utilizara un servicio o aplicación de VPN, es decir, los paquetes de datos que se envían dentro de una VPN, son paquetes encapsulados en la capa de datos de aplicación.

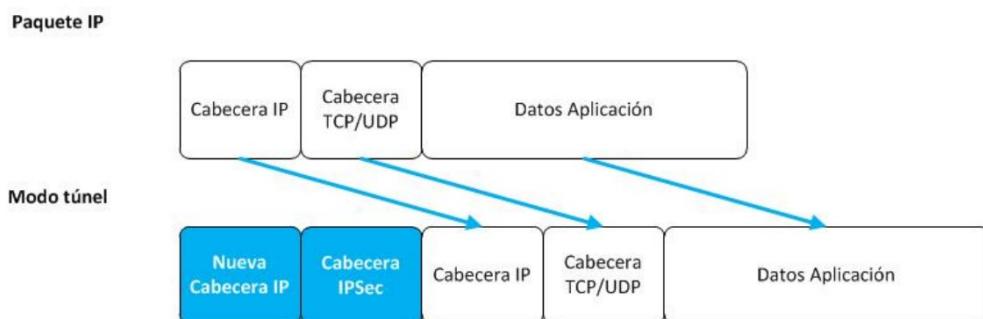


Figura D.2: Encapsulado túnel paquete de VPN en red real.

Esta virtualización puede degradar la condiciones respecto a la red real, especialmente por que los protocolos de transporte, no son capaces de ajustarse, al estar funcionando virtualizada mente sobre otra red.

D.1.2. Protocolo y tecnología

Actualmente existen múltiples opciones donde una política de confianza, mantenimiento o facilidad de configuración en uno y otro software es el desencadenante para su elección. Como soluciones validadas y evaluadas se han utilizado las siguientes tecnologías:

D.1.2.1. Openswan / libreswan

Ipsec[89] vpn, basada en un conjunto de protocolos de los 90 extendido bajo en nombre de librewan o openswan que permite hacer túneles p2p, principalmente estáticos, no dispone de interfaz gráfica y tiene una configuración compleja.

Su principal ventaja es la sencillez de ser nativo y robusto, mientras su principal desventaja es su complejidad para poder pasar un firewall o NAT. Por lo que únicamente lo utilizaremos para túneles estáticos sobre elementos públicos, aunque es posible su configuración, requiere de un conocimiento previo así como no es user-friendly . Su principal característica es rapidez y simplicidad entre ip públicas.

D.1.2.2. OpenVPN y derivados

Openvpn[87], es una conjunto de protocolos y software server/cliente con más de 15 años, en definitiva todo un estándar, estable, fácil de usar, pensado para evadir firewall y NATs. Su principal desventaja es que no es nativo, es decir requiere de instalación en ambos server y cliente, así como la velocidad del mismo no es a nivel de kernel del SO, por lo que induce un delay que degrada las condiciones de la conexión vpn, especialmente cuando utiliza TCP como protocolo de transporte. Su principal característica es su uso extensivo.

D.1.2.3. Wireguard y derivados

Wireguard[88], más moderno (2015) es prácticamente una mejora sobre los dos anteriores. Incluye la totalidad de las funcionalidades de openvpn al igual que permite una ejecución pseudo-nativa en kernel en el servidor. Requiere instalar tanto servidor como cliente ya que no es un protocolo incluido en los SO. Desde 2020 es estable y el actual reemplazo como protocolo estándar de vpn por lo que actualmente es la opción óptima.

D.1.2.4. Otros protocolos

Otros, existen software del nivel y calidad de wireguard, incluso más enfocados a conexiones p2p o descentralizadas como freelan, cjdns, o mejoras sobre protocolos antiguos vulnerados como sstp (mejora sobre pptp). Todos ellos tienen un factor en común, no son de amplio uso y por lo tanto su configuración, instalación y mantenimiento, no es simple y tiende a generar más problemas así como un soporte y documentación más reducido y menos actualizada. Por lo que no son adecuados para todos los públicos.

D.1.2.5. VPN as Service

Por último existen una multitud de servicios y software licenciado gratuito en sheft-host basado en openvpn, wireguard y combinaciones, especialmente en el área de clientes y gestión web-gráfica del servidor y las redes vpn. Sin embargo los más usados como “netmaker”[91], requieren de unos recursos y herramientas “sobredimensionadas” ya que se centran en calidad y facilidad al cliente final y exceden nuestros recursos añadiendo complejidad adicional.

Por ello este documento centra sus pruebas de concepto en estos dos conjuntos de protocolos, **openvpn** y especialmente **wireguard** por su rapidez, en combinación con interfaces gráficas o gestores sencillos, así como su gran uso, incluido por ejemplo en multitud de routers, televisores de manera nativa.

D.1.3. Enrutado

Debemos entender que todo elemento de una red es susceptible de estar interconectado en paralelo a otras redes, es decir, en un nodo con múltiples interfaces. Si dicho elemento permite el enrutado entre redes, puede recibir y reenviar paquetes entre ambas redes, esta propiedad es conocida en sistemas linux como ip forwarding.

Cuando nos conectamos a una red, se proveen de varios parámetros uno de ellos es el default gateway, es decir, la ruta por defecto. Normalmente tiene el valor de la IP del router (dentro de una LAN-NAT), puesto que para salir al exterior se debe enviar los paquetes al router. Sin embargo existe la posibilidad de añadir más rutas en “la tabla de enrutado”.

Existen multitud de protocolos (RIP, OSPF, BGP..) para transmitir qué redes son accesibles a través de qué nodo y añadir las rutas predefinidas en la tabla de enrutado. Pero no es el objetivo de este trabajo, por lo que para los casos de uso actual utilizaremos enruteamientos estáticos añadidos, es decir, modificar en los router, vpn server o clientes vpn la tabla de enrutado para permitir la intercomunicación de redes. Ejemplo de 2 sedes y red en la nube interconectadas:

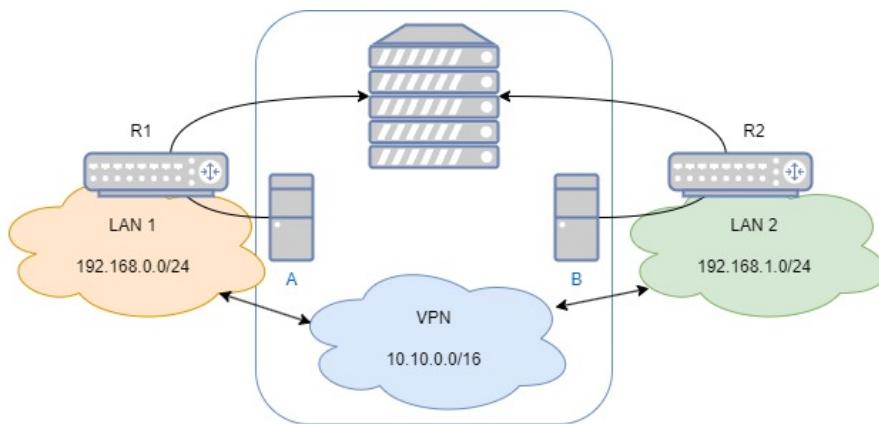


Figura D.3: Diagrama interconexión de redes por vpn.

Por ejemplo desde un elemento en la red naranja la tabla de enrutado es la siguiente: Es decir el router R1 gestiona todas las peticiones dentro y fuera de la red naranja a

Tabla D.1: Tabla enrutado elemento solo conectado a la red naranja.

Destino	Interfaz	Pasarela
default / 0.0.0.0	naranja	R1.naranja
192.168.0.0/24	naranja	destinatario.naranja
192.168.1.0/24	naranja	A.naranja
10.10.0.0/16	naranja	A.naranja

excepción de las peticiones a la red verde y azul, que son redirigidas al elemento A, conectado con la red azul(vpn) y a través de esta a vía la red verde.

Tabla D.2: Tabla enrutado desde A.

Destino	Interfaz	Pasarela
default / 0.0.0.0	A.azul	azul vpn server ip
192.168.0.0/24	A.naranja	destinatario.naranja
192.168.1.0/24	A.azul	B.azul
10.10.0.0/16	A.azul	azul vpn server ip

Queda patente que en la tabla de enrutado del elemento A este 'sale a internet' a través de la red VPN, lo cual es coherente puesto que su red principal es la VPN, aunque como se explico en figura [D.2](#), dichos paquetes son encapsulados dentro de paquetes que si utilizan la tabla de enrutado de la red naranja, por que físicamente salen a través del R1. Desde la red verde podemos hacer observaciones equivalentes:

Tabla D.3: Tabla enrutado desde un elemento solo conectado a la red verde.

Destino	Interfaz	Pasarela
default / 0.0.0.0	verde	R2.verde
192.168.0.0/24	verde	A.azul.ip
192.168.1.0/24	verde	destinatario.verde
10.10.0.0/16	verde	B.verde

Similar al caso de A, B tiene como red principal y gateway la VPN, que es por donde sale a internet. En ambos casos A y B utilizan su segunda interfaz conectada a la red azul, para enrutar paquetes dirigidos a la red azul o a redes que azul permite conectarse, y en contrapartida cuando algún paquete les llega desde la red azul hacia sus redes (naranja y verde) lo enrutan internamente. Son por lo tanto la "gateway"de interconexión en sus respectivas redes locales hacia el resto de redes de la intranet.

Tabla D.4: Tabla enrutado desde B.

Destino	Interfaz	Pasarela
default / 0.0.0.0	B.azul	azul vpn server ip
192.168.0.0/24	B.azul	A.azul
192.168.1.0/24	B.verde	destinatario.verde
10.10.0.0/16	B.azul	azul vpn server ip

Con estas configuraciones cualquier elemento que conectemos a cualquiera de las 3 redes (naranja, azul o verde) tendrá acceso a ellas y por lo tanto es similar a estar todos en una única red privada, una intranet.

Entiéndase que un elemento ajeno a las sedes (trabajador teletrabajando) que se conecte únicamente a la red azul VPN, sin exponer su propia LAN, a la vez que también tendrá acceso a las 3 redes al ser un elemento de una de ellas.

D.1.4. Nat Masquerade

Cuando una red es privada, es decir no pública, significa que la numeración IP es propia y no se expone a internet. Por lo tanto no es accesible desde fuera, y de igual manera si intentamos acceder a internet mediante un mecanismo de salida, los paquetes ip nunca regresaran puesto que nuestra dirección remitente IP es privada.

Es un problema similar a enviar una carta sin remitente porque vivimos en una zona no urbana, o de difícil acceso. Normalmente se utiliza un apartado de correos, o una dirección pública existente, donde recibir la contestación.

La solución a este problema se llama nat-masquerade, y consiste exactamente en eso, nuestro router adsl/fibra está conectado a un ISSP y el si tiene una dirección pública como nuestro “apartado de correos”. El router genera nuestra red privada LAN e implanta un mecanismo de reemplazo de las cabecera IP, con el objetivo de que toda petición a internet sea él quien aparece como remitente y cuando la contestación es recibida, la enruta hace la fuente original.

Este mecanismo es el más habitual para interconectar redes privadas y públicas manteniendo la conectividad hacia afuera, pero negando la conectividad desde fuera de la red, únicamente se puede acceder al router que genera el NAT(Network Address Translation) o contestaciones a peticiones iniciadas desde dentro de la LAN.

Un caso especial son las ISP CG-NAT[26], que indica que la red de nuestro proveedor de internet también es privada, por lo que en realidad estamos navegando desde dos capas de NAT consecutivas.

D.1.5. DNS

DNS o sistema de nombres de dominio, es el servicio de red que nos permite traducir una URL “www.dominio.es” por la correspondiente IP a la que enviar nuestros paquetes. Es de especial importancia ya que aunque estemos conectados correctamente a una red con acceso a internet, sin el apropiado acceso a un dns, creeremos no tener internet ya que las peticiones en “lenguaje humano” no serán resultas. Los DNS permiten no sólo la

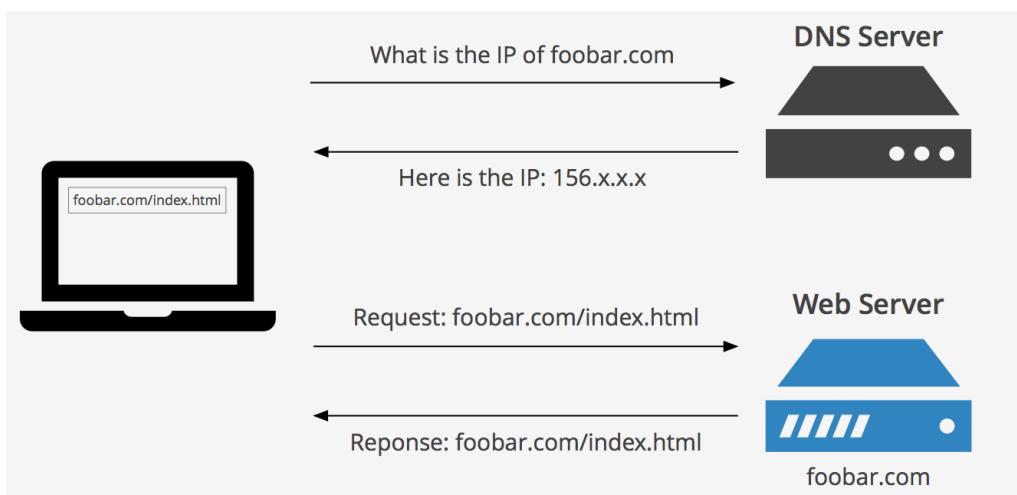


Figura D.4: Diagrama resolución de queries dns proxy[157].

traducción sino el balanceo de carga, el filtrado de peticiones y la monitorización del tipo de tráfico a través de las peticiones dns. Un DNS interno, suele ser un dns relay o proxy, es decir, un punto de repetición hacia un dns exterior que incluye el mapeado de dominios internos de nuestra red privadas, para permitir resolver dominios privados.

El objetivo principal es que aquellas peticiones tales como, ‘webempresaria.internal’, ‘dropbox.internal’, ‘email.internal’, ‘productoA.internal’ puedan resolverse solas sin necesidad de saber la ip o ips indicadas, ya que esta puede ser dinámicas y no depender de una puesta en marcha estática.

Es de especial interés ya que docker permite la generación de redes privadas entre el host y los container. Con el objetivo de evitar el uso de ip internal variables, implementa un sistema interno dns permitiendo el acceso a los container de múltiples formas, nombre, alias de red, hostname del container, identificador del container, ip internas de docker y el mapeo de puertos directamente en la host de docker.

Por ejemplo al configurar un wordpress y su db, como es una configuración que no debe depender de las ip asignadas, se utilizan estas resolución del nombre del container como ip de los servicios.

Existen herramientas como pi hole[116], adguard, nextdns que sobre escriben dominios reales o inspeccionan la queries dns bloqueando anuncios, spam o llamadas inseguras.

D.1.6. Proxy

Un Proxy es un elemento intermediario, es decir, similar al concepto de nat, utilizar un elemento de la red para que envíe y reciba los paquetes por nosotros.

Si es utilizado por los clientes, aquellos servicios que reciben las peticiones creen que es el proxy quien las realiza. Esta casuística era muy común en los 90, ya que debido a las bajas velocidades de conexión (15-64 kbps) recurrentemente se utilizaban un proxy-cache en empresas con el objetivo de obtener información previamente cacheada y optimizar tiempo y recursos. Actualmente se utiliza como método para enmascarar tu geo-ip o para tener acceso a servicios no disponibles en tu zona geográfica.

Si es utilizado por los servidores se le denomina reverse-proxy, ya que funciona de manera invertida. Los clientes realizan múltiples peticiones a un único proxy expuesto públicamente, el proxy identifica las peticiones y las redirige internamente a los servidores de los diferentes servicios. Principalmente permite balancear flujos, añadir https por redirección, centralizar las conexiones y aplicar filtros previos para rechazar peticiones sin usar recursos en los servidores.

D.1.6.1. Let's Encrypt Https

Actualmente todas las páginas web o servicios "profesionales" especialmente seguros requieren de TLS (Transport Layer Security) o focalizados en peticiones web HTTPS[121].

Para implementar el HTTPS es necesario el uso de certificados, dichos certificados si queremos que sean confiables, es decir de utilidad pública, deben estar registrados o emitidos por entes confiables. Actualmente solicitar un certificado puede costar de 6-15€, pero el servicio de Let's Encrypt[122] provee de certificados temporales (3 meses) válidos

y gratuitos para dominios concretos.

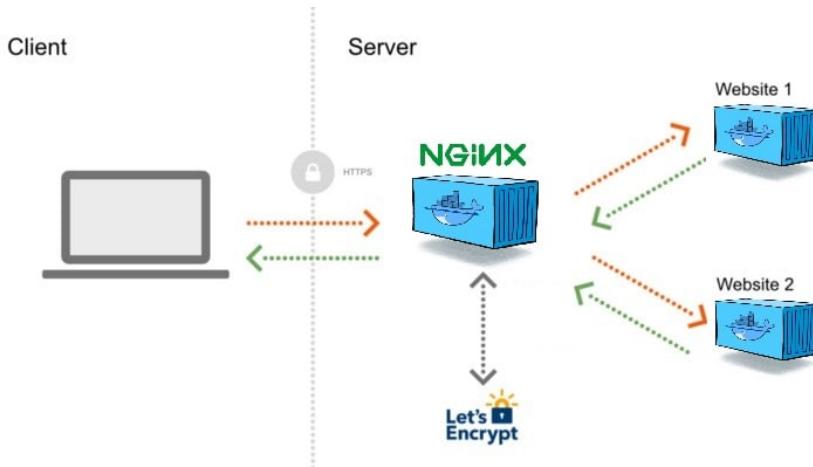


Figura D.5: Ngix-rever proxy + Let's encrypt dockerized[158].

Con el objetivo de que todas nuestras páginas públicas sea aceptadas por los navegadores y confiables se utilizará una estrategia de proxy-reverse basada en ngix y un bot automatizado de Let's encrypt que crea y actualiza los certificados que utiliza el reverse proxy.

D.2. Seguridad

La seguridad natural de una red se consigue por 3 métodos:

- Dificultad o fortaleza de acceso, aquellas redes con protocolos de autenticación más robustos (certificados, algoritmo wireless seguros WPA2-AES), uso de cifrados temporales o medios físicos (cable, roseta) limitados, dificultan o impiden un acceso no autorizado al medio.
- Limitación de acción o autorización por identificadores. Una manera de incrementar esta dificultad puede ser un filtrado de MAC (L2), o autenticación (L3 o superiores) para poder tener acceso real a los recursos de la red. Desgraciadamente existen métodos de clonado e interceptación que permiten engañar a la red, pudiendo suplantar identificadores (MAC, IP) así como suplantar elementos una vez se tiene acceso a la red.
- Limitación de comunicaciones por capas y sub-áreas de la red, el uso de VLAN, switch (L2) que limita las comunicaciones L1, router (L3) que limita las comunicaciones L2 es importante ya que una red jerárquica que aísla las diferentes capas OSI en sub-áreas es más robusta ante un ataque de suplantación, ya que únicamente un porcentaje de las comunicaciones de la red total es accesible en las diferentes capas. Por otra parte ataques masivos que bloquean las capacidades de la red, no son efectivos puesto únicamente afecta a segmentos parciales. Especialmente mencionale que elementos como WIFI, no permiten separar capas L1-L2-L3, siendo vulnerables, ya que una vez conseguido desencriptar el canal de , se pueden acceder a las otras dos capas.

Existen otros métodos activos, como escaneo y monitorización de la propia red para poder identificar intrusos pero no son objetivo de este trabajo.

D.2.1. VLAN

VLAN o virtual LAN, es un mecanismo de generaciones de LAN virtuales. Inicialmente se implementó por puerto, es decir, permitir utilizar los mismos elementos físicos (switch, router) en múltiples LAN sin interactuar entre ellas, creando una segregación de los puertos o cables (L1) que permite convivir múltiples LAN virtuales, sobre una única red física.

También existe un concepto similar a L2, por filtrado de MAC asignado en las diferentes VLAN. Usualmente VLAN se utiliza para cortar o aislar las comunicaciones L2, por ejemplo dentro de una misma LAN-router que dispone de conexiones cableada y wireless.

Desde la aparición de IPv6, también puede segregar en VLAN (L3) por protocolo IPv4 / IPv6 dentro de una misma red física y mismo elementos de hardware. O protocolos customizados Apple talk, IPX etc...

Finalmente continuando con el concepto de segregación, se pueden utilizar parámetros como subred, puertos, protocolo de servicio (FTP, multimedia,...) forma de acceso en la separación de los elementos de las diferentes VLAN.

La conclusión de este punto es que el uso de VLAN nos permite no solo aislar nuestra red cableada de vulneraciones wireless en capa L2, sino que también permite reducir los accesos a regiones concretas de nuestro cableado, por consiguiente es un elemento interesante a la hora de programar una conexión directa RouterHome-Router oficina en casa que permite aislar con mayor robustez un despacho de teletrabajo.

D.2.2. Filtrado Mac

Como se ha mencionado una configuración con filtrado de mac no es invulnerable pero sí es más robusta. Debido al reducido número de elementos que son necesario de interconectar a nuestra Red de teletrabajo, un requisito indispensable debería ser el filtrado MAC tanto de elementos cableados como wireless.

D.3. Encendido Remoto

Existen muchas estrategias de reducción de consumo, la más interesante es el encendido remoto, ya que permite tener los recursos apagados, encenderlos remotamente, lo cual implica un gasto energético únicamente por recursos en uso, con la única contraprestación de un tiempo mínimo de arranque para disponer de ellos.

Para realizar este encendido de recursos en remoto se han utilizado múltiples técnicas compatibles entre sí.

D.3.1. Wake on LAN

Wake on LAN, es un método basado en el arranque por red, es decir, existe una gran gama de elementos (pc, portátil, servidores u otros) que permiten apagar la casi totalidad de elementos de hardware a excepción del conector red. Dicho elemento permanece encendido, siendo capaz de identificar la recepción de paquetes específicos para su MAC. Si recibe un mensaje específico ('wake up') dirigido a su MAC, realizan un arranque del hardware, equivalente a pulsar el botón de power on.

Este mecanismo no solo requiere de un soporte del mismo por hardware (algo bastante común), sino de su apropiada habilitación y configuración (BIOS) y de un cableado, router o switch que lo permitan. Idealmente todos los router-switch giga ethenet, de 8 pins lo permite, pero no es un elemento usual en router-switch ethernet (100 Mbps) o de cableado o comunicación half (4 pines).

D.3.2. Clock - Scheduler wake up

Similar a 'wake on lan', es un mecanismo equivalente que se basa en los relojes internos de la placa del elemento, los cuales siempre mantienen una pequeña batería capaz de ejecutar encendidos a una hora, día, fecha concreta o cada periodos de tiempo definido.

Sigue siendo una funcionalidad explícita del hardware pero aun mas extendida en su implementación que 'wake on lan'. Por lo tanto un simple mecanismo de scripting hace un elemento externo, permite encender por alarma-reloj el dispositivo, verificar si se desea que se mantenga encendido o apagarlo, para que la siguiente alarma-reloj repita el mecanismo.

D.3.3. Smart switch, grid

Por ultimo existen mecanismos de arranque en función de la alimentación eléctrica, es decir, encender cuando se dispone de energía eléctrica, sin necesidad de pulsar power on. Aunque no es un mecanismo propio de el elemento de Red, pc server, permite combinar enchufes inteligentes, commutadores o relés controlados por terceros elementos que mediante una acción online, conecta el dispositivo.

D.4. Elementos Usados

Para la realización tanto del montaje de red de 'la oficina física' capítulo [1](#) y anexo [B](#), así como otros proyectos personales o la aplicación parcial de este documento a la empresa Elenkar se han utilizado los siguientes elementos.

D.4.1. Conexión

Se ha procedido a uso de cableado Cat5/Cat6 RJ45 para el uso de giga ethernet como plataforma física principal. Igualmente debido a las características del cable así como de

las longitudes del mismo es factible el uso de ethernet a 2.5Gbps, especialmente entre los elementos principales de la red.

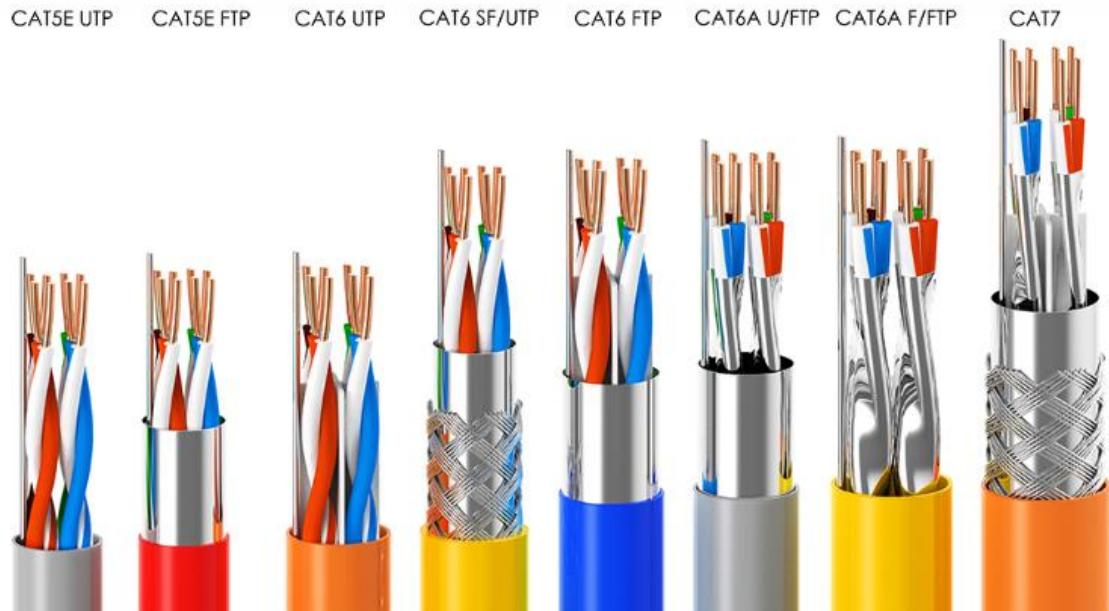


Figura D.6: Cables Cat con sus diferentes trenzados y apantallamientos[159].

Como la conexión principal con el ISP actualmente de los diferentes domicilios/locales oscila entre 300-1000 Mbps, ha prevalecido el uso de Giga ethernet sobre Ethernet para permitir un uso maximizado de la conexión, pero no se han utilizado routers o switch 2.5Gbps, ni protocolos Wifi6, que permiten velocidades superiores a Giga ethernet debido a los aumentos de los costes.

D.4.2. Switch

Tanto para la instalación de la oficina física, como el cableado de diferentes viviendas, incluso ciertos proyectos personales se han usado una multitud de switch entre los que destacan los siguientes modelos giga ethernet:

D.4.2.1. TP-LINK TL-SG108 Switch 8 Puertos

Especialmente utilizado como switch principal, administrado y configurado con VLAN, perfecto para cablear una oficina o casa, figura D.7.

D.4.2.2. TP-Link LS1005G Switch 5 Puertos Gigabit

Adecuado como una versión inferior, con menor cantidad de puertos, figura D.8.

D.4.2.3. Mercusys MS105G Switch 5 Puertos Gigabit

Low cost giga switch, perfecto para redistribución de elementos conectado en la propia oficina, así como wake up o alimentación por ethernet, figura D.9.



Figura D.7: Imagen TP-LINK TL-SG108 Switch 8 Puertos (PCComponent imágenes).

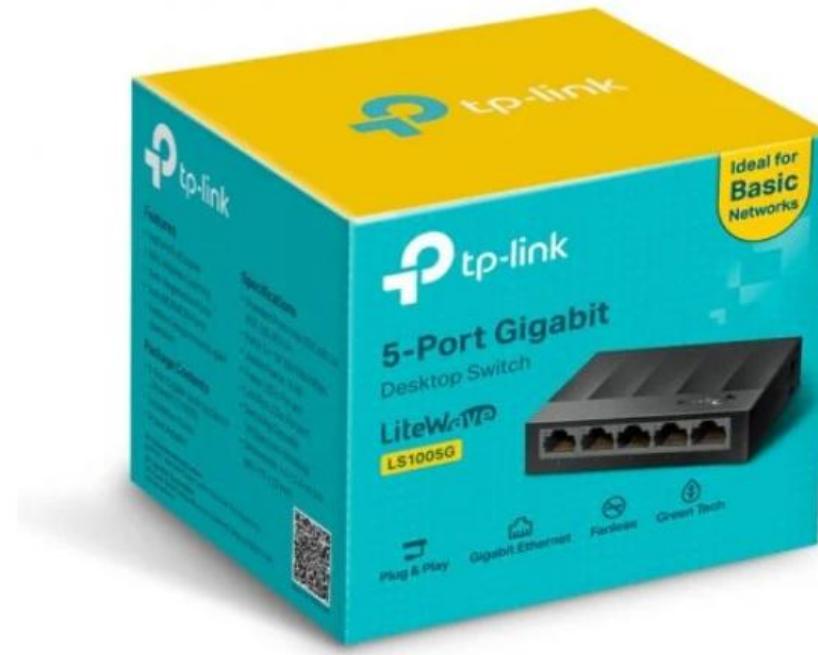


Figura D.8: Imagen TP-Link LS1005G Switch 5 Puertos (PCComponent imágenes).



Figura D.9: Imagen Mercusys MS105G Switch 5 Puertos (PCComponent imágenes).

D.4.3. Router

Actualmente cualquier router WLAN-LAN en un presupuesto 20€-60€, incluye un puerto WLAN giga ethernet(azul), al menos 4 o mas puertos giga ethernet LAN, un hotspot wifi con protocolo 802.11n o superiores y permite la configuración de múltiples modos:

- Modo Router-Broker, genere comunicación entre WLAN, LAN y la WLAN generadas. Tiene servicio DHCP y permite la configuración estática manual (filtros, IPs, tabla de rutas).
- Modo NAT-LAN, similar al anterior genera un LAN y un WLAN interconectadas pero separadas por VLAN, todas ellas con una NAT-MASQUERADE hacia la WLAN que provee de internet. Es equivalente a un router ADSL o fibra.
- Modo wireless WAN, es decir conectarse a otro router de manera inalámbrica que actúa de WAN.
- Modo Repetidor, permite extender las funcionalidades de una LAN/WLAN ya existente (vía cable o wireless).
- Muchos de ellos incluyen 1 o 2 USB, con perspectivas a servicios que requieren almacenamiento o para la interconexión de router-3G/4G como fuente de conectividad en caso de desconexión del puerto WLAN.

Por otra parte suelen incluir múltiples servicios externos tales como:

- Clientes VPN, especialmente openVPN y wireguard o derivados. Algunos también incluyen API de clientes VPN de proveedores externos basados en protocolos estándares.
- Clientes DynamicDNS, son principales clientes de redirección de dominios dns, que se actualizan para permitir apuntar a un IP dinámica desde internet.
- DMZ, firewall filtrado o redirección de puertos, IPv6, servidor multimedia (ftp, vídeo), e incluso servidores vozip configuración de asterix y otros servicios relacionados con multimedia dentro de una LAN.

Se han utilizado múltiples modelos de routers especialmente Linksys E5400-EU Router WiFi AC, Xiaomi Mi Router WiFi AC y Asus RT-AX53U Router WiFi 6, así como algún que otro modelo genérico chino de aliexpress.

Las principales funcionalidades de estos router son basadas en software, es decir, en muchos casos con formatear y cambiar el software embeded del router a versiones como dd-wrt[133] habilita una mejor gestión y mas posibilidades especialmente en clientes VPN y enrutado customizable.

D.4.3.1. Router Portable

Un router portable es básicamente un dispositivo similar a una raspberry pi o placas similares o un router especialmente reducido. Usualmente cuenta con 1 o 2 puertos ethernet,



Figura D.10: Router portable GL-MT300N-V2 (amazon products).

un usb para interconectar router 3G/4G o directamente slot para sim. Similar a los router WAN, permite la interconexión por cable, wireless o modem telefónico y dispone especialmente de conectores VPN-cliente, NAT-Masquerade y la configuración customizada de filtros, rutas y otras herramientas de diagnostico. En muchos casos se alimenta vía USB o 5V, con reducidos consumos, la antenas son co-planar embeded y no tiene una cobertura muy amplio (10-20 m).

Su principal función es ser configurados (sus clientes y parámetros internos) para ser 'desplegados' de manera portable usualmente por Cable Ethernet o conexión móvil (3G/4G) aunque también permite conectarse a hotspot wireless. Su bajo consumo permite a su vez desplegarlo con un powerbank durante varias horas.

La principal ventaja de este router es su portabilidad y movilidad, es decir, permite de una manera práctica utilizarlo como router con vpn integrada en cualquier LAN a la que se conecte físicamente, facilitando la configuración de un entorno seguro en cualquier acceso con a internet median rj45.

D.4.4. Servidor autocrático

Los servidores autocráticos son por definición servidores en casa o en la oficina, permiten un uso mayor de recursos aunque la 'availability' de sus recursos se ve resentida por la red eléctrica y la conexión de comunicaciones de la sede en la que se sitúan.

Por otra parte esta el tema de gasto eléctrico; para evitar un consumo eléctrico mayor al coste de un servidor en la nube o dedicado es necesario limitar la potencia del servidor o las horas encendido.

Existen varias estrategias:

- Hardware especializado, que tiene bajo consumos en momento de baja actividad.
- Utilización de mini-pc con bajos consumos 6-10 W (CPU N100 , N300) o plataformas arm como raspberry pi o similares 1-6W.

- Encendidos y apagados programados basado en un scheduler y demanda de trabajo por realizar.

En todas ellas el objetivo es un coste reducido o en función del trabajo realizado, y es factible la combinación de más de una solución con el fin de no perjudicar el principal activo que es la mayor disponibilidad de recursos en el servidor autocrático que un VPS.

D.4.4.1. Arm servers

Debido a la naturaleza de bajo coste y especialmente bajo consumo las placas basados en arm son perfectas para servidores low cost con consumos de 1-6 w.



Figura D.11: Placa Rasberry pi (amazon products imagen).



Figura D.12: Placa Orange Pi PC 3 (Aliexpress products imagen).

En las implementaciones de los diversos proyectos donde se ha aplicado este documento se han utilizado múltiples versiones, todas ellas basadas en rasberri pi OS, raspbian (cus-

tom debian para raspberry pi), ubuntu/fedora/debian arm versión, armbian (custom debian para arm boards) entre otros.

Así mismo se han utilizado placas Rasberry pi V1, V2, V3, junto a otras placas similares OrangePi Pc3, BananaPI M2+, en cada uno de los domicilios o sedes donde existen servidores autocráticos, elementos de red, interconexiones LAN-VPN o directamente como servidor autocrático en si.

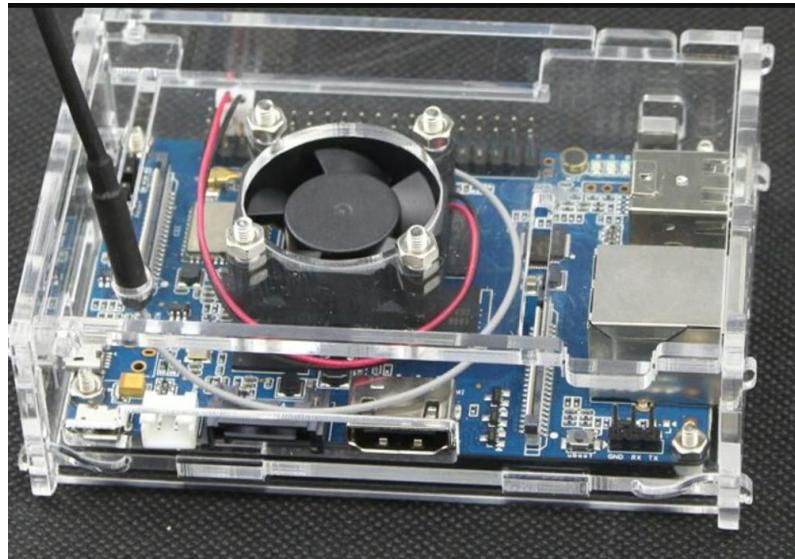


Figura D.13: Placa Banana Pi M2+(Aliexpress products imagen).

D.4.4.2. Noc server

Aquellos casos que se requiere de un servidor de recursos aceptables y especialmente procesador X86-64, es decir, para ejecutar servicios dockerizados no especializados en plataforma arm. Se han seleccionado mini-pc con los CPU N100-N305 con consumo de 6-15W pero la potencia en recursos de un ordenador de 2010-15.



Figura D.14: EQ12 Pro y Chuwi LarkBox X (Aliexpress products imagen).

D.4.4.3. PC y portátiles viejos

Un gran abanico de antiguos portátiles o torres, semi actualizadas pueden ser utilizados como servidores temporales, es decir fuente de recursos encendidos remotamente para por ejemplo mediante ansible setear un cluste de docker swarm o kubernetes con propósitos de pruebas y desarollo, para ser apagado después de su uso.

D.4.5. Camaras IP

Se dispone de una amalgama de cámaras IP la gran mayoría con imagen normal, infrarrojo, sonido y altavoz sobre el actuar. Así como motorizadas 360º o con seguimiento de humanos, tanto interiores como exteriores; todas con costes entre los 10€-60€ (low cost).

La principal característica dentro de este documento es el uso de VPS-VPN para centralizar y permitir un acceso ágil a las IP cam, permitiendo usar software dockerizado para la monitorización y generación de flujos de vídeo, tanto para almacenamiento como para procesado de imágenes. Todas las cámaras IP se sitúan en zonas privadas, por lo que la regulación de no es aplicable y tiene un carácter preventivo de seguridad.

El uso de esta VPN e interconexión de fuentes ipcam con servicios dockerizados que las utilizan, permite evadir el uso de las aplicaciones nativas de dichas cámaras, usualmente con fallos de seguridad y blindar el firewall de las diferentes LAN, para evitar tener ataques reversos a través de las cámaras ip, con software no parcheado.

Por ultimo, el relego de la generación de diferentes señales Wireless en cada LAN, especialmente en el caso de las ipcam, con el objetivo de separar compartimentando, pero especialmente evitando degradar la señal de la WLAN como el protocolo usado debido a enlaces de datos degradados (ipcam-router).

D.4.6. Smart Devices

Se han utilizado diferentes elementos inteligentes o red de sensores, focalizado en demórica o acciona-dores. Como elementos principales destacan el uso de Alexa / Google Home como interfaz audio-vocal y gateways vía bluetooth o WIFI, así como la instalación de gateways de zigbee, z-wave o sensores a 144 Mhz.

Inicialmente se buscaba una interacción entre diversos elementos, luces, sensores y acciones para elementos de la oficina tales como, luz, temperatura, y automatismos. Sin embargo en una segunda revisión como proyecto personal involucró la seguridad y monitorización de elementos como el consumo eléctrico, para finalmente también añadir actuadores.

Como conclusión se han utilizado una plataforma de sensores y actuadores (incluyendo alarma y centralita) basada en GauTone, plataformas Alexa / google home, la interconexión de diversos electrodomésticos todo ello centralizado por una raspberry pi con Home Assistant, que es el elemento que a través de autenticación de token API en las otras plataformas, permite unir transversalmente todos los elementos, aísla o federaliza los recursos de las API bajo un criterio más seguro y privado.

Destaco por su sencillez y bajo coste:

- Sensores magnéticos, sensores de presencia, sensor de temperatura y humedad.
- Sensor-adicionador termostato (accesible y programable vía nube), luces, enchufes inteligentes o conmutadores, así como PIA conectados.
- La facilidad de integración de sensores propios vía placas ESP32 o raspberry en API de terceros.

APÉNDICE E. PRUEBAS DE CONCEPTO Y MVP

En este anexo se citas las principales y mas útiles elementos de pruebas de concepto así como Mínimo Producto Viable MVP basico de cara a usar en proyecto o necesidades de una pequeña empresa.

E.1. Pruebas de Concepto

Las pruebas de conceptos, son docker-compose en donde se instancian los servicios auxiliares y servicios a probar. Aunque no requiere de las restricciones o interconexiones con otras redes usualmente se tiene a definir redes internas de docker si es necesario así como volúmenes en las mismas carpeta del docker-compose. Muchas de estas pruebas se basan en la versión oficial o ejemplos proveídos por su principal comunidad mantenedora.

E.1.1. Comunicación

Existen multitud de servicios de comunicación revisadas incluso probados durante la realización de este documento tabla C.4. En muchos casos especialmente en los relacionados con email la configuración es bastante compleja o en muchos sistemas similares a slack usan demasiados recursos o subsistemas auxiliares. Uno de los sistemas mas aceptables de ser usado es rocket chat[78] con una base de datos auxiliar en este caso mongodb.

```
version: '3.3'
services:
  rocketchat:
    image: registry.rocket.chat/rocketchat/rocket.chat:6.1.0
    restart: unless-stopped
    volumes:
      - rocketchat-uploads:/app/uploads
    environment:
      MONGO_URL: mongodb://mongodb:27017/rocketchat?replicaSet=rs0
      MONGO_OPLOG_URL: mongodb://mongodb:27017/local?replicaSet=rs0
      ROOT_URL: http://localhost:3000 # https://rocketchat.example.com
      PORT: 3000
      DEPLOY_METHOD: docker
      Accounts_UseDNSDomainCheck: 'false'
    depends_on:
      - mongodb
    expose:
      - 3000
    ports:
      - "3000:3000"

  mongodb:
    image: bitnami/mongodb:4.4
    restart: unless-stopped
    volumes:
      - mongodb_data:/bitnami/mongodb
    environment:
      MONGODB_REPLICA_SET_MODE: primary
      MONGODB_REPLICA_SET_NAME: rs0
      MONGODB_PORT_NUMBER: 27017
```

```

MONGODB_INITIAL_PRIMARY_HOST: mongodb
MONGODB_INITIAL_PRIMARY_PORT_NUMBER: 27017
MONGODB_ADVERTISED_HOSTNAME: mongodb
MONGODB_ENABLE_JOURNAL: 'true'
ALLOW_EMPTY_PASSWORD: 'yes'

volumes:
  mongodb_data:
  rocketchat-uploads:

```

Code E.1: docker-compose.yml Rocket Chat prueba de concepto.

E.1.2. Ejemplo web

La herramienta mas fácil de generar una web dinámica así como gestionarla es utilizar un framework, especialmente si hablamos de un framework que debe ser gestionado por personas no técnicas sin tocar código. Wordpress es una excelente opción, es mayoritario, existen una gran cantidad de plugins y módulos prefabricados y guías extensas que permiten crear una web rápida y fácilmente.

```

version: "3.5"
services:
#
# Wordpress blog page
#
mariadb:
  container_name: mariadb
  image: docker.io/bitnami/mariadb:11.0
  ports:
    - '23306:3306'
  volumes:
    - './mariadb /:/bitnami/mariadb'
  environment:
    #ALLOW_EMPTY_PASSWORD: "yes"
    MARIADB_USER: ${MARIADB_USER:-bn_wordpress}
    MARIADB_PASSWORD: ${MARIADB_PASSWORD:-p4ssw0rd_wordpress}
    MARIADB_DATABASE: ${MARIADB_DATABASE:-bitnami_wordpress}
    MARIADB_ROOT_PASSWORD: ${MARIADB_ROOT_PASS:-root_pass}

wordpress:
  container_name: wordpress
  image: docker.io/bitnami/wordpress:6
  ports:
    - '28080:8080'
    - '28443:8443'
  volumes:
    - './wordpress /:/bitnami/wordpress'
  depends_on:
    - mariadb
    - nginx
  environment:
    VIRTUAL_HOST: blog.programing.es
    VIRTUAL_PORT: 8080
    LETSENCRYPT_HOST: blog.programing.es
    WORDPRESS_DATABASE_HOST: mariadb
    WORDPRESS_DATABASE_PORT_NUMBER: 3306

```

```

WORDPRESS_DATABASE_NAME: ${MARIADB_DATABASE:-bitnami_wordpress}
WORDPRESS_DATABASE_USER: ${MARIADB_USER:-bn_wordpress}
WORDPRESS_DATABASE_PASSWORD: ${MARIADB_PASSWORD:-p4ssw0rd_wordpress}
WORDPRESS_USERNAME: ${WORDPRESS_USER:-ropnom}
WORDPRESS_PASSWORD: ${WORDPRESS_PASSWORD:-tfg}
WORDPRESS_EMAIL: ${WORDPRESS_EMAIL:-rodrigo.sc@programing.es}
WORDPRESS_FIRST_NAME: ${WORDPRESS_FIRST_NAME:-Rodrigo}
WORDPRESS_LAST_NAME: ${WORDPRESS_LAST_NAME:-Sampedro}
WORDPRESS_BLOG_NAME: ${WORDPRESS_BLOG_NAME:-"Programar Ingenieria"}
WORDPRESS_PLUGINS: ${WORDPRESS_PLUGINS}
WORDPRESS_SMTP_HOST: ${WORDPRESS_SMTP_HOST}
WORDPRESS_SMTP_PORT: ${WORDPRESS_SMTP_PORT}
WORDPRESS_SMTP_USER: ${WORDPRESS_SMTP_USER}
WORDPRESS_SMTP_PASSWORD: ${WORDPRESS_SMTP_PASSWORD}
WORDPRESS_ENABLE_REVERSE_PROXY: "yes"

```

Code E.2: docker-compose.yml Wordpress prueba de concepto.

E.1.3. Ejemplo wiki

Para la evaluación del uso de wiki o documentación existen múltiples alternativas. Dependiendo de la casuística, documentación de software, pagina de documentación e información colectiva, foros de debate y reporte, existen múltiples herramientas aplicables. Con el fin de cubrir diversas casuísticas especialmente en nivel de recursos se han evaluado 3 opciones, bajos recursos para documentación PineDocs[61], intermedio en recursos y comunitario MediaWiki[59], y casos customizados para desarrollo sin usar excesivos recursos BookStack[60].

E.1.3.1. PineDocs

```

version: '3'
services:
  web:
    image: xy2z/pinedocs:1.2.5
    ports:
      - 3000:80
    volumes:
      - ./data:/data/pinedocs

```

Code E.3: docker-compose.yml PineDocs prueba de concepto.

E.1.3.2. MediaWiki

```

version: '3'
services:
  mediawiki:
    image: mediawiki
    restart: always
    ports:
      - 8080:80
    links:
      - database

```

```

volumes:
- ./images:/var/www/html/images
  # After initial setup, download LocalSettings.php to the same directory
  # as
  # this yaml and uncomment the following line and use compose to restart
  # the mediawiki service
  #- ./LocalSettings.php:/var/www/html/LocalSettings.php
# This key also defines the name of the database host used during setup
# instead of the default "localhost"
database:
  image: mariadb
  restart: always
  environment:
    # @see https://phabricator.wikimedia.org/source/mediawiki/browse/master/
    includes/DefaultSettings.php
    MYSQL_DATABASE: my_wiki
    MYSQL_USER: wikiuser
    MYSQL_PASSWORD: example
    MYSQL_RANDOM_ROOT_PASSWORD: 'yes'
  volumes:
    - ./db:/var/lib/mysql

```

Code E.4: docker-compose.yml MediaWiki prueba de concepto.

E.1.3.3. BookStack

```

version: "3"
services:
  bookstack:
    image: ghcr.io/linuxserver/bookstack
    container_name: bookstack
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=Europe/Madrid
      - APP_URL=http://localhost:6875
      - DB_HOST=bookstack_db
      - DB_USER=bookstack
      - DB_PASS=p4ssw0rd
      - DB_DATABASE=bookstackapp
    volumes:
      - ./app-data:/config
    ports:
      - "6875:80"
    restart: unless-stopped
    depends_on:
      - bookstack_db

  bookstack_db:
    image: ghcr.io/linuxserver/mariadb
    container_name: bookstack_db
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=Europe/Madrid
      - MYSQL_ROOT_PASSWORD=p4ssw0rd
      - MYSQL_DATABASE=bookstackapp

```

```

    - MYSQL_USER=bookstack
    - MYSQL_PASSWORD=p4ssw0rd
  volumes:
    - ./db-data:/config
  restart: unless-stopped

```

Code E.5: docker-compose.yml BookStack prueba de concepto.

E.1.4. Cloud Storage

Se han evaluado diferentes opciones de bajos y medianos recursos como servicios de almacenamiento de datos extendido y sincronizado (cloud).

E.1.4.1. Sea File

```

version: '2.0'
services:
  db:
    image: mariadb:10.6
    container_name: seafiler-mysql
    environment:
      - MYSQL_ROOT_PASSWORD=db_dev # Requested, set the root's password of MySQL service.
      - MYSQL_LOG_CONSOLE=true
    volumes:
      - ./db:/var/lib/mysql # Requested, specifies the path to MySQL data persistent store.
    networks:
      - seafiler-net

  memcached:
    image: memcached:1.6.18
    container_name: seafiler-memcached
    entrypoint: memcached -m 256
    networks:
      - seafiler-net

  seafiler:
    image: seafilerd/seafiler-mc:latest
    container_name: seafiler
    ports:
      - "80:80"
      - "443:443" # If https is enabled, cancel the comment.
    volumes:
      - ./seafiler-data:/shared # Requested, specifies the path to Seafiler data persistent store.
    environment:
      - DB_HOST=db
      - DB_ROOT_PASSWORD=db_dev # Requested, the value shuold be root's password of MySQL service.
      - TIMEZONE=Etc/UTC # Optional, default is UTC. Should be uncomment and set to your local time zone.
      - SEAFILE_ADMIN_EMAIL=me@example.com # Specifies Seafiler admin user, default is 'me@example.com'.

```

```

    - SEAFILE_ADMIN_PASSWORD=asecret      # Specifies Seafile admin password,
    default is 'asecret'.
    - SEAFILE_SERVER.LETSENCRYPT=false   # Whether to use https or not.
    - SEAFILE_SERVER.HOSTNAME=docs.seafile.com # Specifies your host name if
    https is enabled.
depends_on:
    - db
    - memcached
networks:
    - seafile-net

networks:
    seafile-net:

```

Code E.6: docker-compose.yml SeaFile prueba de concepto.

E.1.4.2. Samba server

```

version: '3.4'
services:
  samba:
    image: dperson/samba
    environment:
      TZ: 'EST5EDT'
    networks:
      - default
    ports:
      - "137:137/udp"
      - "138:138/udp"
      - "139:139/tcp"
      - "445:445/tcp"
    read_only: true
    tmpfs:
      - /tmp
    restart: unless-stopped
    stdin_open: true
    tty: true
    volumes:
      - /mnt:/mnt:z
      - /mnt2:/mnt2:z
    command: '-s "Mount;/mnt" -s "Bobs Volume;/mnt2;yes;no;bob" -u "bob;
    bobspasswd" -p'

networks:
  default:

```

Code E.7: docker-compose.yml Samba prueba de concepto.

E.1.4.3. NextCloud

```

version: "3.4"
#https://gist.github.com/mrzapp/08947dd861bc826c4e02cee2d4da51ae
networks:
  postgres: ~
  redis: ~

```

```

services:
  # PostgreSQL
  postgres:
    container_name: postgres
    environment:
      - POSTGRES_PASSWORD=somepassword
      - POSTGRES_USER=someuser
    image: postgres:latest
    restart: always
    volumes:
      - "./postgres/init:/docker-entrypoint-initdb.d/"
      - "./postgres/data:/var/lib/postgresql/data"
      - "/etc/localtime:/etc/localtime:ro"
    networks:
      - postgres

  # NextCloud
  nextcloud:
    container_name: nextcloud
    depends_on:
      - postgres
    image: nextcloud:latest
    restart: always
    ports:
      - 1000:80
    volumes:
      - "./nextcloud/data:/var/www/html/data"
      - "./nextcloud/config:/var/www/html/config"
      - "./nextcloud/themes:/var/www/html/themes"
      - "./nextcloud/custom_apps:/var/www/html/custom_apps"
      - "/etc/localtime:/etc/localtime:ro"
    networks:
      - postgres
      - redis

  # Redis
  redis:
    container_name: redis
    image: redis:latest
    restart: always
    networks:
      - redis

  # Collabora
  collabora:
    container_name: collabora
    image: collabora/code:latest
    cap_add:
      - MKNOD
    environment:
      - domain=test\.local:9980
      - username=someuser
      - password=somepassword
    ports:
      - 9980:9980
    restart: always
    volumes:

```

```
      - "/etc/localtime:/etc/localtime:ro"
```

Code E.8: docker-compose.yml NextCloud prueba de concepto.

E.1.5. Reverse Proxy y https

Aunque se ha evaluado Traefik[123] y Ngix[124], pero debido a una mayor cantidad de soporte y documentación, se ha entendido que Ngix es mas apropiado como prueba de concepto resolutiva.

```
version: "3.5"
services:
  nginx:
    container_name: nginx
    image: nginxproxy/nginx-proxy
    restart: unless-stopped
    labels:
      - com.github.nginx-proxy.nginx
    ports:
      - 80:80
      - 443:443
    volumes:
      - /var/run/docker.sock:/tmp/docker.sock:ro
      - ./nginx/html:/usr/share/nginx/html
      - ./nginx/certs:/etc/nginx/certs
      - ./nginx/vhost:/etc/nginx/vhost.d
    logging:
      options:
        max-size: "10m"
        max-file: "3"

  letsencrypt-companion:
    container_name: letsencrypt-companion
    image: jrcs/letsencrypt-nginx-proxy-companion
    restart: unless-stopped
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
      - ./nginx/html:/usr/share/nginx/html
      - ./nginx/certs:/etc/nginx/certs
      - ./nginx/vhost:/etc/nginx/vhost.d
      - ./nginx/acme:/etc/acme.sh
    environment:
      DEFAULT_EMAIL: ropnom5291@gmail.com
      NGINX_PROXY_CONTAINER: nginx

  hello-world:
    container_name: hello-world
    image: kornkitti/express-hello-world
    expose:
      - "8080"
    environment:
      VIRTUAL_HOST: test.programing.es
      LETSENCRYPT_HOST: test.programing.es
```

Code E.9: docker-compose.yml Ngix proxy y Let's Encrypt prueba de concepto.

E.1.6. Ticketing

Después de evaluar diferentes softwares dockerizados de ticketing, he llegado a la conclusión que la gran mayoría de servicios externos gratuitos son bastante adecuados para un equipo reducido.

Por otra parte la mayor parte de servicios de ticketing son muy costosos en recursos por lo que el único ejemplo valido bajo en recursos es Planka[93].

```
version: '3'
services:
  planka:
    image: ghcr.io/plankanban/planka:latest
    command: >
      bash -c
      "for i in `seq 1 30`; do
        ./start.sh &&
        s=$$? && break || s=$$?;
        echo \"Tried $$i times. Waiting 5 seconds...\";
        sleep 5;
      done; (exit $$s)"
    restart: unless-stopped
    volumes:
      - ./user-avatars:/app/public/user-avatars
      - ./project-background-images:/app/public/project-background-images
      - ./attachments:/app/private/attachments
    ports:
      - 3000:1337
    environment:
      - BASE_URL=http://localhost:3000
      - TRUST_PROXY=0
      - DATABASE_URL=postgresql://postgres@postgres/planka
      - SECRET_KEY=notsecretkey

      # Can be removed after installation
      - DEFAULT_ADMIN_EMAIL=demo@demo.demo # Do not remove if you want to
        prevent this user from being edited/deleted
      - DEFAULT_ADMIN_PASSWORD=demo
      - DEFAULT_ADMIN_NAME=Demo Demo
      - DEFAULT_ADMIN_USERNAME=demo

      # related: https://github.com/knex/knex/issues/2354
      # As knex does not pass query parameters from the connection string we
      # have to use environment variables in order to pass the desired values,
      e.g.
      # - PGSSLMODE=<value>

      # Configure knex to accept SSL certificates
      # - KNEX_REJECT_UNAUTHORIZED_SSL_CERTIFICATE=false
    depends_on:
      - postgres

  postgres:
    image: postgres:14-alpine
    restart: unless-stopped
    volumes:
      - ./db-data:/var/lib/postgresql/data
    environment:
```

- POSTGRES_DB=planka
- POSTGRES_HOST_AUTH_METHOD=trust

Code E.10: docker-compose.yml Planka prueba de concepto.

E.1.7. SSO Ldap y Keycloak

Con el fin de implementar un SSO, se ha utilizado LDAP[85] y una interfaz gráfica de configuración y un servicio de Keycloak[86].

```
version: '3.8'
services:
  mysql:
    image: mysql:5.7.43
    container_name: mysql
    ports:
      - "3306:3306"
    environment:
      - MYSQL_DATABASE=keycloak
      - MYSQL_USER=keycloak
      - MYSQL_PASSWORD=password
      - MYSQL_ROOT_PASSWORD=root_password
    healthcheck:
      test: "mysqladmin ping -u root -p${MYSQL_ROOT_PASSWORD}"
  keycloak:
    image: quay.io/keycloak/keycloak:22.0.3
    container_name: keycloak
    environment:
      - KEYCLOAK_ADMIN=admin
      - KEYCLOAK_ADMIN_PASSWORD=admin
      - KC_DB=mysql
      - KC_DB_URL_HOST=mysql
      - KC_DB_URL_DATABASE=keycloak
      - KC_DB_USERNAME=keycloak
      - KC_DB_PASSWORD=password
      - KC_HEALTH_ENABLED=true
    ports:
      - "8080:8080"
    command: start-dev
    depends_on:
      - mysql
    healthcheck:
      test: "curl -f http://localhost:8080/health/ready || exit 1"
  openldap:
    image: osixia/openldap:1.5.0
    container_name: openldap
    environment:
      - LDAP_ORGANISATION="MyCompany Inc."
      - LDAP_DOMAIN=mycompany.com
    ports:
      - "389:389"
  phpldapadmin:
    image: osixia/phpldapadmin:0.9.0
```

```

container_name: phpldapadmin
environment:
  - PHPLDAPADMIN_LDAP_HOSTS=openldap
ports:
  - "6443:443"
depends_on:
  - openldap

```

Code E.11: docker-compose.yml SSO LDAP-Keycloak prueba de concepto.

E.1.8. VPN

En el caso de VPN se han evaluado las dos opciones mas utilizadas OpenVPN y wi-reguard, sin embargo se han descartado los clientes multiprotocolo y comerciales por opciones ligeras, eficientes e integradas con una UI simple.

E.1.8.1. OpenVPN

Se ha buscado una versión especialmente ligera de openvpn y una interfaz web sencilla y fácil de gestionar:

```

---
version: "3.5"
#define networks
networks:
  openvpn:
    name: openvpn
    #external: true
    internal: false
    driver: bridge
    driver_opts:
      com.docker.network.bridge.name: openvpn
    ipam:
      config:
        - subnet: 192.168.88.0/24
services:
  openvpn:
    container_name: openvpn
    image: d3vilh/openvpn-server:latest-amd64
    privileged: true
    ports:
      - "1194:1194/udp"
    environment:
      TRUST_SUB: 10.0.70.0/24
      GUEST_SUB: 10.0.71.0/24
      HOME_SUB: 192.168.88.0/24
    volumes:
      - ./pki:/etc/openvpn/pki
      - ./clients:/etc/openvpn/clients
      - ./config:/etc/openvpn/config
      - ./staticclients:/etc/openvpn/staticclients
      - ./log:/var/log/openvpn
      - ./fw-rules.sh:/opt/app/fw-rules.sh
    cap_add:
      - NET_ADMIN

```

```

networks:
  - openvpn
  restart: always

openvpn-ui:
  container_name: openvpn-ui
  image: d3vilh/openvpn-ui:latest-amd64
  environment:
    - OPENVPNADMIN_USERNAME=admin
    - OPENVPNADMIN_PASSWORD=gagaZush
  privileged: true
  ports:
    - "8080:8080/tcp"
  volumes:
    - ./etc/openvpn
    - ./db:/opt/openvpn-ui/db
    - ./pki:/usr/share/easy-rsa/pki
    - /var/run/docker.sock:/var/run/docker.sock:ro
  restart: always
  networks:
    - openvpn

```

Code E.12: docker-compose.yml OpenVPN con UI.

E.1.8.2. Wireguard

En el caso de wireguard se han encontrado dos soluciones alternativas que han sido utilizadas para crear el ejemplo de múltiples layer de vpn.

E.1.8.3. Wireguard UI

```

version: "3.8"
networks:
  external_net:
    external: true
    name: external_net
services:
  # WireGuard VPN service
  wireguard:
    image: linuxserver/wireguard:latest
    container_name: wireguard
    cap_add:
      - NET_ADMIN
      - SYS_MODULE
    volumes:
      - ./config:/config
    ports:
      # Port of the WireGuard VPN server
      - 51820:51820/udp

  # WireGuard-UI service
  wireguard-ui:
    image: ngoduykhanh/wireguard-ui:latest
    container_name: wireguard-ui
    depends_on:

```

```

    - wireguard
  cap_add:
    - NET_ADMIN
  # Use the network of the 'wireguard' service
  # This enables to show active clients in the status page
  #network_mode: service:wireguard
  environment:
    - SENDGRID_API_KEY
    - EMAIL_FROM_ADDRESS
    - EMAIL_FROM_NAME
    - SESSION_SECRET
    - WGUI_USERNAME=admin
    - WGUI_PASSWORD=password
    - WG_CONF_TEMPLATE
    - WGUI_MANAGE_START=true
    - WGUI_MANAGE_RESTART=true
  logging:
    driver: json-file
    options:
      max-size: 50m
  volumes:
    - ./db:/app/db
    - ./config:/etc/wireguard
  ports:
    # Port for WireGuard-UI
    - 5000:5000
  http:
    image: nginxdemos/hello:latest
    labels:
      SERVICE_80_NAME: http
      SERVICE_TAGS: production
    ports:
      - 8080:80

```

Code E.13: docker-compose.yml Wireguard prueba de concepto.

E.1.8.4. Wireguard Easy

```

version: "3.5"
networks:
  external_front:
    name: external_front
    #external: true
    internal: false
    driver: bridge
    driver_opts:
      com.docker.network.bridge.name: docker_external
    ipam:
      config:
        - subnet: 192.168.99.0/24
services:
  wg-easy:
    environment:
      - WG_HOST=programming.es
      # Optional:
      - PASSWORD=pass
      # - WG_PORT=21821

```

```

      - WG_DEFAULT_ADDRESS=10.252.1.x
      # - WG_DEFAULT_DNS=1.1.1.1
      # - WG_MTU=1420
      # - WG_ALLOWED_IPS=192.168.15.0/24, 10.0.1.0/24
      # - WG_PRE_UP=echo "Pre Up" > /etc/wireguard/pre-up.txt
      # - WG_POST_UP=echo "Post Up" > /etc/wireguard/post-up.txt
      # - WG_PRE_DOWN=echo "Pre Down" > /etc/wireguard/pre-down.txt
      # - WG_POST_DOWN=echo "Post Down" > /etc/wireguard/post-down.txt

image: weejewel/wg-easy
container_name: wg-easy
volumes:
  - .config:/etc/wireguard
ports:
  - "51820:51820/udp" # 51820/udp"
  - "51821:51821/tcp"
restart: unless-stopped
cap_add:
  - NET_ADMIN
  - SYS_MODULE
sysctls:
  - net.ipv4.ip_forward=1
  - net.ipv4.conf.all.src_valid_mark=1
networks:
  - external_front

http2:
image: nginxdemos/hello:latest
labels:
  SERVICE_80_NAME: http
  SERVICE_TAGS: production
hostname: demo2.internal
# ports:
#   - 80
networks:
  - external_front

```

Code E.14: docker-compose.yml Wireguard Easy prueba de concepto.

E.1.9. CI / CD

Se han realizado diferentes pruebas de CI/CD especialmente con jenkins sin embargo el ejemplo mas simple y dockerizable con bajos recursos es drone, así mismo se ha utilizado gitea[68] como principal repositorio de control de versiones.

```

version: '3.6'
services:
  gitea:
    container_name: gitea
    image: gitea/gitea:${GITEA_VERSION:-1.20}
    restart: unless-stopped
    environment:
      # https://docs.gitea.io/en-us/install-with-docker/#environments-variables
      - APP_NAME="Gitea"
      - USER_UID=1000
      - USER_GID=1000

```

```

- RUN_MODE=prod
- DOMAIN=${IP_ADDRESS}
- SSH_DOMAIN=${IP_ADDRESS}
- HTTP_PORT=3000
- ROOT_URL=http://${IP_ADDRESS}:3000
- SSH_PORT=222
- SSH_LISTEN_PORT=22
- DB_TYPE=sqlite3
ports:
- "3000:3000"
- "222:22"
networks:
- cicd_net
volumes:
- ./gitea:/data

drone:
  container_name: drone
  image: drone/drone:${DRONE_VERSION:-2.20}
  restart: unless-stopped
  depends_on:
    - gitea
  environment:
    # https://docs.drone.io/server/provider/gitea/
    - DRONE_DATABASE_DRIVER=sqlite3
    - DRONE_DATABASE_DATASOURCE=/data/database.sqlite
    - DRONE_GITEA_SERVER=http://${IP_ADDRESS}:3000/
    - DRONE_GIT_ALWAYS_AUTH=false
    - DRONE_RPC_SECRET=${DRONE_RPC_SECRET}
    - DRONE_SERVER_PROTO=http
    - DRONE_SERVER_HOST=${IP_ADDRESS}:3001
    - DRONE_TLS_AUTOCERT=false
    - DRONE_USER_CREATE=${DRONE_USER_CREATE}
    - DRONE_GITEA_CLIENT_ID=${DRONE_GITEA_CLIENT_ID}
    - DRONE_GITEA_CLIENT_SECRET=${DRONE_GITEA_CLIENT_SECRET}
  ports:
    - "3001:80"
    - "9001:9000"
  networks:
    - cicd_net
  volumes:
    - /var/run/docker.sock:/var/run/docker.sock
    - ./drone:/data

drone-runner:
  container_name: drone-runner
  image: drone/drone-runner-docker:${DRONE_RUNNER_VERSION:-1.8}
  restart: unless-stopped
  depends_on:
    - drone
  environment:
    # https://docs.drone.io/runner/docker/installation/linux/
    # https://docs.drone.io/server/metrics/
    - DRONE_RPC_PROTO=http
    - DRONE_RPC_HOST=drone
    - DRONE_RPC_SECRET=${DRONE_RPC_SECRET}
    - DRONE_RUNNER_NAME="${HOSTNAME}-runner"
    - DRONE_RUNNER_CAPACITY=2

```

```

  - DRONE_RUNNER_NETWORKS=cicd_net
  - DRONE_DEBUG=false
  - DRONE_TRACE=false
  ports :
    - "3002:3000"
  networks :
    - cicd_net
  volumes :
    - /var/run/docker.sock:/var/run/docker.sock

networks :
  cicd_net:
    name: cicd_net

```

Code E.15: docker-compose.yml Gitea-Drone CI prueba de concepto.

E.2. Wireguard LAN-VPN-LAN

Se ha seleccionado wireguard como vpn principal, sin embargo la configuración de wireguard tanto en servidor como en cliente se basa en script estáticos de re-direccionamiento y enrutado para las diferentes LAN, redes docker internas y VPN.

En este apartado se muestra ejemplo de configuración de ambos, servidor y clientes para que el usuario a través de la UI, pueda configurar manual y estáticamente sus decisiones.

E.2.1. Nat masquerade hace red docker

Para permitir la comunicación entre la red docker interna principal y la red VPN se puede utilizar un nat masquerada como indica la siguiente configuración de ejemplo:

```
[Interface]
Address = 10.10.1.0/24 # vpn range
ListenPort = 51820 # vpn server port
PrivateKey = <private key> # vpn server private key
MTU = 1450 # vpn MTU size (1500 - overhead bytes encapsulated)
PostUp = iptables -A FORWARD -i %i -j ACCEPT; iptables -A FORWARD -o %i
        -j ACCEPT; iptables -t nat -A POSTROUTING -s 10.10.1.0/24 -o eth0
        -j MASQUERADE; iptables -A INPUT -p udp -m udp --dport 21820 -j
        ACCEPT
PostDown = iptables -D FORWARD -i %i -j ACCEPT; iptables -D FORWARD -o
           %i -j ACCEPT; iptables -t nat -D POSTROUTING -s 10.10.1.0/24 -o
           eth0 -j MASQUERADE; iptables -D INPUT -p udp -m udp --dport 21820 -
           j ACCEPT
Table = auto
```

Code E.16: Ejemplo de configuracion servidor Nat masquerade

E.2.2. Script en cliente

Uno de los punto manuales en este documentos, es la configuración de un elemento de la RED VPN y LAN como nodo intermedio de comunicaciones. Debido a la naturaleza del elemento normalmente router portable o una raspberry pi, requiere en todo momento de:

- Tener el ip forwarding activado para permitir el enrutado de paquetes entre redes.
- Definir su tabla de rutas, añadiendo manualmente que otras Redes permite enrutar y por que elemento.
- Incluir la configuración pertinente de su IP en la VPN y rango de su LAN para ser ofertada en otros elementos de la red

En un proceso iterativo, es decir, una vez definida el 'mapa' de las redes que vamos a interconectar y que elementos participan en dicha interconexión, es necesario agrupar el conjunto de LAN a ofrecerse al resto de clientes VPN, y manipular dicha configuración para obviar las redes LAN en los elementos interconectadores.

Debido a la naturaleza de Wireguard UI, no es un mecanismo automatizado y debe hacerse a mano cliente a cliente. De igual manera aunque los elementos interconectores funcionen y la configuración de la VPN, es necesario editar manualmente los routers de las LAN para que indiquen adecuadamente a través de que elementos se puede acceder tanto a la VPN como al resto de LAN.

```
[Interface]
PrivateKey = <private key>
Address = <client ip>
DNS = <internal dns proxy relay>
PostUp = DROUTE=$(ip route | grep default | awk '{print $3}'); HOMENET
=192.168.0.0/16; HOMENET2=10.0.0.0/8; HOMENET3=172.16.0.0/12; ip
route add $HOMENET3 via $DROUTE; ip route add $HOMENET2 via $DROUTE;
ip route add $HOMENET via $DROUTE; iptables -I OUTPUT -d $HOMENET -
j ACCEPT; iptables -A OUTPUT -d $HOMENET2 -j ACCEPT; iptables -A
OUTPUT -d $HOMENET3 -j ACCEPT; iptables -A OUTPUT ! -o %i -m mark
! --mark $(wg show %i fwmark) -m addrtype ! --dst-type LOCAL -j
REJECT
PreDown = HOMENET=192.168.0.0/16; HOMENET2=10.0.0.0/8; HOMENET3
=172.16.0.0/12; ip route del $HOMENET3 via $DROUTE; ip route del
$HOMENET2 via $DROUTE; ip route del $HOMENET via $DROUTE; iptables
-D OUTPUT ! -o %i -m mark ! --mark $(wg show %i fwmark) -m addrtype
! --dst-type LOCAL -j REJECT; iptables -D OUTPUT -d $HOMENET -j
ACCEPT; iptables -D OUTPUT -d $HOMENET2 -j ACCEPT; iptables -D
OUTPUT -d $HOMENET3 -j ACCEPT
```

Code E.17: Ejemplo de configuración servidor para homenets

Aunque es factible automatizar con protocolos estándar OSPF o RIP, la interconexión automática y anunciado de estas redes, requiere de funcionalidades a típicas de un router LAN por lo que se han descartado esta opción ante la falta de soporte en muchos routers habituales.

En el ejemplo E.17 podemos observar como el cliente cuando inicia la vpn wireguard, añade rutas estáticas de destinación al vpn server, si estas son adecuadamente seteadas con las IP o elementos que interconectan las LAN, los clientes podrán hacer peticiones a las LAN interconectadas por la VPN.

E.3. Mínimo Viable Producto

El MVP es una prueba de concepto que valida tanto las funcionalidades de servicios dockerizados en VPS, el rever proxy con https funciona, la generación de un dns interno y el uso de VPN para acceder a los servicios no públicos, así como una prueba de dos capas de VPN para servicios específicos.

Al ser un propósito general de prueba, muchos servicios web como pueden ser una wiki, un repositorio espejo etc se han sustituido por meros mock web, que permiten visionar un mensaje de 'hello word' o propiedades del servidor web que los ejecuta.

```
version: "3.5"

#define networks
networks:
  external:
    name: z_external
    #external: true
    internal: false
    driver: bridge
    driver_opts:
      com.docker.network.bridge.name: external
    ipam:
      config:
        - subnet: 192.168.66.0/24
  vpn_layer1:
    name: vpn
    #external: true
    internal: false
    driver: bridge
    driver_opts:
      com.docker.network.bridge.name: vpn_layer1
    ipam:
      config:
        - subnet: 192.168.99.0/24
  vpn_layer2:
    name: depp_vpn
    #external: true
    internal: false
    driver: bridge
    driver_opts:
      com.docker.network.bridge.name: vpn_layer2
    ipam:
      config:
        - subnet: 192.168.100.0/24

services:

# _____
# Public services
# _____
# all manage by nginx-proxy and let's encrypt
#
# reverse proxy service
nginx:
  container_name: nginx-proxy
```

```

image: nginxproxy/nginx-proxy
restart: unless-stopped
labels:
  - com.github.nginx-proxy.nginx
ports:
  - 80:80
  - 443:443
volumes:
  - /var/run/docker.sock:/tmp/docker.sock:ro
  - ./nginx/html:/usr/share/nginx/html
  - ./nginx/certs:/etc/nginx/certs
  - ./nginx/vhost:/etc/nginx/vhost.d
logging:
  options:
    max-size: "10m"
    max-file: "3"
networks:
  - external

# Let's encrypt service (tls certificates for https service)
letsencrypt-companion:
  container_name: letsencrypt-companion
  image: jrcs/letsencrypt-nginx-proxy-companion
  restart: unless-stopped
  volumes:
    - /var/run/docker.sock:/var/run/docker.sock
    - ./nginx/html:/usr/share/nginx/html
    - ./nginx/certs:/etc/nginx/certs
    - ./nginx/vhost:/etc/nginx/vhost.d
    - ./nginx/acme:/etc/acme.sh
  environment:
    DEFAULT_EMAIL: ropnom5291@gmail.com
    NGINX.PROXY.CONTAINER: nginx
  networks:
    - external

#
# Wordpress blog page
#
mariadb:
  container_name: mariadb
  image: docker.io/bitnami/mariadb:11.0
  volumes:
    - './mariadb' :/bitnami/mariadb'
  environment:
    #ALLOW.EMPTY.PASSWORD: "yes"
    MARIADB_USER: ${MARIADB_USER:-bn_wordpress}
    MARIADB_PASSWORD: ${MARIADB_PASSWORD:-p4ssw0rd_wordpress}
    MARIADB_DATABASE: ${MARIADB_DATABASE:-bitnami_wordpress}
    MARIADB_ROOT_PASSWORD: ${MARIADB_ROOT_PASS:-root_pass}
  networks:
    - external

wordpress:
  container_name: wordpress_blog
  image: docker.io/bitnami/wordpress:6
  # ports:
  #   - '28080:8080'

```

```

#      - '28443:8443'
expose:
  - '8080'
volumes:
  - './wordpress /:/ bitnami/wordpress'
depends_on:
  - mariadb
  - nginx
environment:
  VIRTUAL_HOST: blog.programing.es
  VIRTUAL_PORT: 8080
  LETSENCRYPT_HOST: blog.programing.es
  WORDPRESS_DATABASE_HOST: mariadb
  WORDPRESS_DATABASE_PORT_NUMBER: 3306
  WORDPRESS_DATABASE_NAME: ${MARIADB_DATABASE:-bitnami_wordpress}
  WORDPRESS_DATABASE_USER: ${MARIADB_USER:-bn_wordpress}
  WORDPRESS_DATABASE_PASSWORD: ${MARIADB_PASSWORD:-p4ssw0rd_wordpress}
  WORDPRESS_USERNAME: ${WORDPRESS_USER:-ropnom}
  WORDPRESS_PASSWORD: ${WORDPRESS_PASSWORD:-tfg}
  WORDPRESS_EMAIL: ${WORDPRESS_EMAIL:-rodrigo.sc@programing.es}
  WORDPRESS_FIRST_NAME: ${WORDPRESS_FIRST_NAME:-Rodrigo}
  WORDPRESS_LAST_NAME: ${WORDPRESS_LAST_NAME:-Sampedro}
  WORDPRESS_BLOG_NAME: ${WORDPRESS_BLOG_NAME:-"Programar Ingenieria"}
  WORDPRESS_PLUGINS: ${WORDPRESS_PLUGINS}
  WORDPRESS_SMTP_HOST: ${WORDPRESS_SMTP_HOST}
  WORDPRESS_SMTP_PORT: ${WORDPRESS_SMTP_PORT}
  WORDPRESS_SMTP_USER: ${WORDPRESS_SMTP_USER}
  WORDPRESS_SMTP_PASSWORD: ${WORDPRESS_SMTP_PASSWORD}
  WORDPRESS_ENABLE_REVERSE_PROXY: "yes"
networks:
  - external

#
# Other services like next cloud service
#
#
# VPN services
#
# WireGuard VPN service
wireguard:
  container_name: wireguard_vpn1
  image: linuxserver/wireguard:latest
  sysctls:
    - net.ipv4.ip_forward=1
    - net.ipv4.conf.all.src_valid_mark=1
  cap_add:
    - NET_ADMIN
    - SYS_MODULE
  volumes:
    - ./vpn/config:/config
environment:
  VIRTUAL_HOST: vpn.programing.es
  LETSENCRYPT_HOST: vpn.programing.es
  VIRTUAL_PORT: 5000
ports:
  # Port of the WireGuard VPN server
  - 21820:21820/udp

```

```

# UI port
# - 21500:5000
networks:
  vpn_layer1:
    external:

# WireGuard-UI service
wireguard-ui:
  container_name: wireguard-ui-vpn1
  image: ngoduykhanh/wireguard-ui:latest
  depends_on:
    - wireguard
  cap_add:
    - NET_ADMIN
  environment:
    # SENDGRID_API_KEY:
    # EMAIL_FROM_ADDRESS:
    # EMAIL_FROM_NAME:
    # SESSION_SECRET:
    WGUIL_USERNAME: admin
    WGULPASSWORD: pass
    # WG_CONF_TEMPLATE:
    WGUILMANAGE_START: "true"
    WGUILMANAGE_RESTART: "true"
  logging:
    driver: json-file
    options:
      max-size: 50m
  volumes:
    - ./vpn/ui/db:/app/db
    - ./vpn/config:/etc/wireguard
  # Use the network of the 'wireguard' service
  # This enables to show active clients in the status page
  network_mode: service:wireguard
  # ports: #using service:wireguard
  #   # Port for WireGuard-UI
  #   - 21500:5000
  # networks:
  #   - external

wg-easy:
  image: weejewel/wg-easy
  container_name: wg-easy
  volumes:
    - ./config:/etc/wireguard
  # ports:
  #   - "51820:51820/udp"
  #   - "51821:51821/tcp"
  restart: unless-stopped
  environment:
    WG_HOST: vpn2.internal
    PASSWORD: pass
    WG_PORT: 51820
    WG_DEFAULT_ADDRESS: 10.252.1.x
    WG_DEFAULT_DNS: 192.168.100.53
    WG_PERSISTENT_KEEPALIVE: 20
    WG_MTU: 1400
    WG_ALLOWED_IPS: "10.252.1.0/24,192.168.100.0/24"

```

```

# - WG_PRE_UP=echo "Pre Up" > /etc/wireguard/pre-up.txt
WG_POST_UP: "DETH=$(ip route | grep default | awk '{ print $$5}')";
iptables -A FORWARD -i %i -j ACCEPT; iptables -A FORWARD -o %i -j ACCEPT;
iptables -t nat -A POSTROUTING -s 10.252.1.0/24 -o $$DETH -j MASQUERADE"
# - WG_PRE_DOWN=echo "Pre Down" > /etc/wireguard/pre-down.txt
WG_POST_DOWN: "DETH=$(ip route | grep default | awk '{ print $$5}')";
iptables -D FORWARD -i %i -j ACCEPT; iptables -D FORWARD -o %i -j ACCEPT;
iptables -t nat -D POSTROUTING -s 10.252.1.0/24 -o $$DETH -j MASQUERADE"
cap_add:
- NET_ADMIN
- SYS_MODULE
sysctls:
- net.ipv4.ip_forward=1
- net.ipv4.conf.all.src_valid_mark=1
networks:
vpn_layer1:
  ipv4_address: 192.168.99.100
  aliases:
  - vpn2.internal
vpn_layer2:

# VPN dns service
dns-proxy:
  container_name: dns-proxy-mageddo
  image: defreitas/dns-proxy-server:3.14.5
  # ports:
  #   - "192.168.99.53:53:53/udp"
  #   - "5380:5380"
  volumes:
  - /var/run/docker.sock:/var/run/docker.sock
  - /etc/resolv.conf:/etc/resolv.conf
  - ./dns/config/:/app/conf/
  environment:
    MG_REGISTER_CONTAINER_NAMES: "true"
  networks:
    vpn_layer1:
      ipv4_address: 192.168.99.53 #fix ip for relay dns
      aliases:
      - "dns1.internal"
    vpn_layer2:
      ipv4_address: 192.168.100.53 #fix ip for relay dns
      aliases:
      - "dns2.internal"

dns-proxy2:
  container_name: dns-proxy-dynamic
  image: carlsverre/damn-simple-dns-proxy:latest
  # ports:
  #   - "192.168.99.53:53:53/udp"
  networks:
    vpn_layer1:
      ipv4_address: 192.168.99.54 #fix ip for main dns

# demo web
http:
  image: nginxdemos/hello:latest
  labels:

```

```

SERVICE_80_NAME: http
SERVICE_TAGS: production
hostname: demo.internal
# ports:
#   - 80
networks:
  vpn_layer1:
    aliases:
      - webtest.internal
depends_on:
  - dns-proxy
  - dns-proxy2

http2:
  image: nginxdemos/hello:latest
  labels:
    SERVICE_80_NAME: http
    SERVICE_TAGS: production
  hostname: demo2.internal
  # ports:
  #   - 80
  networks:
    vpn_layer2:
      aliases:
        - webtest2.internal
  depends_on:
    - dns-proxy
    - dns-proxy2

hello-world:
  container_name: hello-world
  image: kornkitti/express-hello-world
  expose:
    - "8080"
  environment:
    VIRTUAL_HOST: test.programing.es
    LETSENCRYPT_HOST: test.programing.es
  networks:
    - external

```

Code E.18: docker-compose.yml MVP prueba de concepto.

E.4. Aprovisionamiento básico Ansible

Debido a restricciones de seguridad y privacidad, así como evitar futuros ataques a la plataforma de Elenkar únicamente se muestran los procesos de abastecimiento junto a segurización básica del VPS.

Véase código del proyecto [160].

APÉNDICE F. PROYECTOS DE APLICACIÓN

Este anexo explica los usos o aplicaciones reales de este documento.

Aunque existe una implementación real del Capítulo 1, la realidad es que mi casuística personal de teletrabajo, toda la parte virtual o software no es auto-gestionada, sino que sigue los estándares directrices y decisiones de la empresa en la que trabajo.

En conclusión en mi realidad diaria, no es aplicable. Sin embargo existen otras realidades en las que si se ha implementado los temas 2 y 3 como parte de la flexibilización o trabajo remoto parcial para mi pareja, donde si tengo margen de maniobra para decidir y aplicar este documento.

Finalmente en casos como el inicio de un negocio (autónomo o startup) requiere de unos patrones o automatizaciones muy similares, por lo tanto he colaborado y ayudado a varios amigos o ex-compañeros de trabajo en una aplicación customizada de 'la nube virtual' para beneficio en sus proyectos personales, así como queda claro que lo utilizo parcialmente en mis proyectos amateur, o del ámbito casero-familiar.

F.1. Elenkar

Elenkar S.L es una pequeña empresa de 3-4 trabajadores enfocada en servicios inmobiliarios y servicios exclusivos relacionados, afincada en el baix penedes.

Sus principales necesidades eran Web (captación de clientes), mail (método de comunicación vía internet), capacidad de almacenaje y compartición de documentos (dropbox / drive) con otras inmobiliarias y un acceso externo remoto cuando no se esta en la oficina.

El servicio web son externalizados actualmente especialmente por la intermodalidad de portales por ciertos CRM específicos para inmobiliarias y aunque en el periodo 2013-2017 este autor genero un software Java completo para la gestión de la empresa, quedo totalmente desfasado tras el boom de portales inmobiliarios 2017-19 y su concentración posterior.

El servicio mail externalizado en este caso no solo por un tema de 'availability' sino por términos legales y espacio, para almacenar y guardar los mails durante un tiempo estipulado, no es viable o excesivamente complejo auto gestionar el servicio de mail.

El servicio de almacenamiento, compartido en muchos casos con otras empresas obliga a la contratación de dropbox bussines (200-250€ la año) ha sido sustituido por un servicio Nextcloud mas adecuado a las necesidades y con múltiples extensiones no plausibles en dropbox, por 48€ / año. La implementación de un servidor samba interno dentro de la LAN de la oficina principal también ha permitido el traspaso de ficheros de una manera ágil y eficaz, evitando la sobrecarga de nextcloud. Por otra parte la introducción de otras herramientas como weTransfer[132] y [46] ha permitido un canal seguro de comunicación externa especialmente para información confidencial.

Uno de los puntos mas interesantes ha sido la combinación de un servicio de VPN, cámaras ip, sensores y interruptores/enchufes inteligentes. La necesidad de permitir un trabajo flexible y remoto a la oficina requiere de una ágil conexión remota a los PC de la ofici-

na, únicamente proveída por una VPN. Por otra parte la VPN facilito el uso en remoto de impresora y escáneres, así como el acceso al servidor samba. Destaca el uso de una banana pi, no solo como servidor autócrata, sino como elemento de 'wake on lan' para el arranque de los pc y el acceso en remoto directamente al pc de trabajo (véase D.3.1.).

Finalmente se instaló un conjunto de ipcam, sensores y alarma que son accesibles desde la LAN-VPN permitiendo un acceso ágil sin necesidad de exponer dichas ipcam a aplicaciones oficiales made in china de escasa ciberseguridad o confianza. Integración y utilización de una raspberry pi con home assistan junto a un gateway zigbee, que permite monitorizar en paralelo sensores de alarma(magnéticos y de presencia), grabaciones cámaras IP, así como automatizaciones de luz, enchufes, calefactores o aire acondicionado.

Desgraciadamente debido a cuestiones de seguridad así como privacidad, Elenkar no ha permitido la publicación detallada o al pormenor tanto de los scripts, mapa de red o diagramas de interacción de sus elementos inteligentes.

F.2. Casos menores, emprendimiento

Principalmente proveer de un multi-hosting, vpn personal a un coste económico a amigos y conocidos, reduciendo sus gasto de emprendedores.

Normalmente la contratación de un hosting para Wordpress, prestashop necesario para cualquier negocio, requiere de un coste 10-15€ / mes, dominio 6-15€/año, certificado SSL (6-15 €/año), mail (10-50€/año) y usualmente los honorarios de un profesional que nos realiza dichas tareas, normalmente customizando el wordpress, plugins etc... En resumen mantener una web de un pequeño negocio puede costar entorno a 150€/año y su montaje asumiendo casos sencillos de 300-800€ según el profesional y la calidad del trabajo desarrollado.

Este proyecto no solo permite reducir los costes de 150€/año a un precio entorno 60€/año sino que permite una actualización continua sin mantenimiento de terceros (al menos en periodos inferiores a un año), así como facilita la gestión y la introducción de los propietarios a docker-compose, con su ágil e intuitivo funcionamiento, para permitir desplegar elementos autónomicamente.

Es de especial utilidad la facilidad de desplegar nuevos contenedores para web auxiliares o derivadas, que pueden auto gestionarse con certificados https, sin necesidad de ningún profesional.

Por ultimo remarco la usabilidad de la VPN, ya que aunque puede que solo sea útil para pymes o negocios digitalizados, elementos claves como seguridad, domótica, que permiten tener servicios propios que externamente son excesivamente caros de mantener o de instalar.

F.3. Mi uso personal

Mi mayor uso personal radica en la VPN y la accesibilidad a las diferentes LAN familiares así como para mis proyectos personales como monitorización de todo tipo de sensores.

Verdaderamente dispongo de múltiples ordenadores, mini pc, raspberry pi y banana pi diseminados entre diferentes domicilios y localizaciones, centralizar esta infraestructura gracias al VPS y la VPN así como disponer de un lugar públicos para desplegar mis proyectos, blog personal y mail es una infraestructura considerables raramente asumible a tan ínfimo coste.

Multitud de proyectos de índole de sensores, smarthings y en especial de procesado de imagen basado en ipcams, son de extrema dificultad si se desea interconectar las fuentes, y micro servicios dockerizados que procesen dicha información, y ejecuten acciones sobre una infraestructura unificada.