

I.E.S Fernando Aguilar Qignon



PROYECTO FIN DE GRADO

HEIMDALL

Realizado por
Alberto Martínez de Pinillos Verches

Para la obtención del título de
Grado en Administración de Sistemas Informáticos en Red

Dirigido por
Domingo Anillo Álvarez

Año 2023/24.

Agradecimientos

En primer lugar, quiero expresar mi más sincero agradecimiento a mi tutor Domingo Anillo Alvarez por su invaluable orientación, paciencia y apoyo durante la realización de este trabajo. Sus conocimientos y consejos han sido fundamentales para el desarrollo de este proyecto.

Agradezco también a Jaime Canillas Galiano, Jorge Juan Delgado Durán, Javier García Estévez, Ricardo Mariscal Quintero y Juan Diego Gonzalez Ramos por sus enseñanzas y consejos a lo largo de mi carrera. Su dedicación y pasión por la enseñanza me han inspirado a esforzarme cada día más.

También quiero agradecer a mis compañeros de clase que me han acompañado en estos años, gracias por su colaboración y apoyo. Sus comentarios y críticas constructivas han sido esenciales para mejorar mi trabajo.

A mis padres y a mi familia, les agradezco de corazón por su amor, comprensión y constante apoyo. Sin su sacrificio este logro no habría sido posible.

A mis amigos, la familia que he ido escogiendo a lo largo de mi corta vida, gracias por estar siempre ahí para mí, brindándome apoyo emocional y motivación. Agradezco poder jactarme de tener una familia tan rica y extensa.

Quiero hacer una mención especial a mi tutor de prácticas de FCT, Darío Bruzón, por su apoyo y por proporcionarme los recursos necesarios para llevar a cabo este proyecto además de brindarme sus conocimientos ya que ha marcado una diferencia significativa en los resultados obtenidos. Su colaboración ha sido de gran ayuda para la culminación de este proyecto.

”Viaje antes que destino, Fuerza antes que debilidad, Vida antes que muerte.”

~ The Stormlight Archive - The Way of Kings ~ Brandon sanderson

Resumen

Durante el periodo de prácticas en un entorno real como parte de mi Formación en Centros de Trabajo (FCT), hemos decidido desarrollar una aplicación web utilizando Django. Esta iniciativa tiene como objetivo principal mejorar la gestión de incidencias dentro de la organización mediante un sistema integral de gestión de tickets y comunicación interna.

Para lograr esto, la aplicación permitirá a los usuarios registrar incidencias de manera estructurada, asignar prioridades, dar seguimiento al estado de cada ticket y registrar las acciones tomadas para su resolución. Se implementará un flujo de trabajo automatizado que mejorará la eficiencia operativa, reduciendo los tiempos de respuesta y aumentando la satisfacción del cliente interno.

Un componente clave de la aplicación será un panel de control interactivo creado con Plotly, que proporcionará análisis detallados de estadísticas como el número de tickets abiertos, resueltos y los tiempos de resolución promedio. Este panel de control permitirá a los administradores y responsables tomar decisiones informadas mediante la visualización en tiempo real de tendencias de incidencias por departamento o área específica, facilitando así la identificación proactiva de problemas y la implementación de soluciones preventivas.

Además de la gestión de tickets, la aplicación integrará un sistema de correo electrónico interno utilizando tecnologías como Postfix, Dovecot y Roundcube. Esto mejorará significativamente la comunicación entre los equipos dentro de la organización, facilitando el intercambio rápido de información crucial para la resolución efectiva de incidencias y la colaboración entre departamentos.

Desde el punto de vista de la infraestructura, tanto la aplicación Django como el servidor de correos se desplegarán en un entorno Linux. Esto garantizará la estabilidad y seguridad necesarias para operaciones críticas. Para la gestión centralizada de usuarios, se utilizará un servidor Windows Server con Active Directory (AD), aprovechando LDAP para la sincronización automática de credenciales entre el AD y la aplicación Django. Esta integración asegurará un acceso seguro y eficiente a la plataforma, simplificando la administración de usuarios y manteniendo la coherencia en los permisos de acceso.

En resumen, el proyecto tiene como resultado esperado la implementación de una solución integral que no solo optimice la gestión de incidencias y la comunicación interna, sino que también promueva una cultura organizacional más colaborativa y eficiente. Al proporcionar herramientas avanzadas para la toma de decisiones basadas en datos y mejorar la eficiencia operativa, la aplicación contribuirá directamente a la mejora continua y al éxito a largo plazo de la organización.

Índice general

1. Capítulo I: Introducción	1
1.1. Contextualización	1
1.2. Alcance del Proyecto	1
1.3. Objetivos del proyecto	2
1.4. Planificación del proyecto	3
1.4.1. Desglose de tareas	3
1.4.2. Comunicación y Seguimiento	3
2. Capítulo II: Marco teórico	4
2.1. Python	4
2.2. Django	4
2.3. Plotly	5
2.4. Postfix	6
2.5. Dovecot	6
2.6. Roundcube	7
2.7. Active Directory	8
2.7.1. Dominio	8
2.7.2. Objeto	9
2.7.3. Controlador de dominio	9
2.7.4. Árboles	9
2.7.5. Bosque	10
2.7.6. Unidad Organizativa	11
2.7.7. Relaciones de confianza	11
2.8. Apache	12
2.9. Nginx	12
2.10. MySQL	13
2.11. Certbot	13
3. Capítulo III: Fase de diseño	14
3.1. Recursos humanos.	14
3.2. Recursos hardware.	15
3.3. Recursos software.	15
3.4. Postfix	16
3.5. Roundcube	18
3.5.1. Dovecot	25
3.6. Django	29
3.6.1. Modelos	30
3.6.2. Vistas	33
3.6.3. URL	35
3.6.4. Settings	36
3.6.5. Señales	42
3.7. Nginx	46
3.8. Certificados	48
3.9. Active Directory	51

4. Capítulo IV: Fase de pruebas	58
4.1. Creación de usuarios en active directory	58
4.2. Distribución de usuarios para pruebas	58
4.3. Resultados de las pruebas y corrección de errores	59
4.4. Mejora continua	59
5. Capítulo V: Documentación del sistema	60
5.1. Manual de superadministrador	60
5.1.1. Windows server	60
5.1.2. Linux Server	60
5.1.3. Manual del Administrador	61
5.2. Manual del técnico	64
5.3. Manual del trabajador	68
6. Capítulo VI: Conclusiones finales	70
7. Anexo I: Documento de pruebas aportado por Carlos Montesino	72

Índice de figuras

2.1. Python. Fuente: picpng.com	4
2.2. Django. Fuente: 1000logos.net	4
2.3. Plotly. Fuente: wikipedia.org	5
2.4. Postfix. Fuente: wikipedia.org	6
2.5. Dovecot. Fuente: wikipedia.org	6
2.6. Roundcube. Fuente: roundcube.net	7
2.7. Árboles. Fuente: somebooks.es	10
2.8. Bosque. Fuente: somebooks.es	10
2.9. Unidad Organizativa. Fuente: somebooks.es	11
2.10. Apache. Fuente: wikipedia.org	12
2.11. Apache. Fuente: wikipedia.org	12
2.12. MySQL. Fuente: mysql.com	13
2.13. Cerbot. Fuente: github.com/certbot	13
 3.1. Dell PowerEdge R740	15
3.2. Instalación de Postfix 03	16
3.3. Instalación de Postfix 04	16
3.4. Instalación de Postfix 05	17
3.5. Instalación de Postfix 06	17
3.6. Instalación de Postfix 07	17
3.7. Instalación de Postfix 08	17
3.8. Instalación de Roundcube 01	18
3.9. Instalación de Roundcube 02	19
3.10. Instalación de Roundcube 03	19
3.11. Instalación de Roundcube 04	19
3.12. Instalación de Roundcube 05	20
3.13. Instalación de Roundcube 06	20
3.14. Instalación de Roundcube 07	20
3.15. Instalación de Roundcube 08	21
3.16. Instalación de Roundcube 09	21
3.17. Instalación de Roundcube 10	21
3.18. Instalación de Roundcube 11	21
3.19. Instalación de Roundcube 12	22
3.20. Instalación de Roundcube 13	22
3.21. Instalación de Roundcube 14	22
3.22. Instalación de Roundcube 15	23
3.23. Instalación de Roundcube 16	23
3.24. Instalación de Roundcube 17	24
3.25. Instalación de Roundcube 18	24
3.26. Instalación de Roundcube 19	25
3.27. Instalación de Roundcube 20	25
3.28. Instalación de dovecot 01	25
3.29. Configuración de dovecot, main.cf 02	26
3.30. Instalación de dovecot master.cf 03	26

3.31. Configuración de dovecot, dovecot.conf 04	26
3.32. Configuración de dovecot, dovecot.conf 05	26
3.33. Configuración de dovecot, 10-auth.conf 06	27
3.34. Configuración de dovecot, 10-master.conf 07	27
3.35. Instalación de Roundcube 21	27
3.36. Instalación de Roundcube 24	28
3.37. Instalación de Roundcube 25	28
3.38. Instalación de Roundcube 26	28
3.39. Arbol de directorios mostrado usando "tree"	30
3.40. MR heimdall	31
3.41. Django url 01	35
3.42. Django url 02	35
3.43. Django url 03	36
3.44. Django configuración settings.py 01	37
3.45. Django configuración settings.py 02	38
3.46. Django configuración settings.py 03	38
3.47. Django configuración settings.py 04	39
3.48. Django configuración settings.py 05	39
3.49. Django configuración settings.py 06	40
3.50. Django configuración settings.py 07	40
3.51. Django configuración settings.py 08	41
3.52. Django configuración settings.py 09	41
3.53. Configuración de señales 01	42
3.54. Configuración de señales 02	43
3.55. Configuración de señales 03	43
3.56. Configuración de señales 04	44
3.57. Configuración de señales 05	45
3.58. Configuración de señales 06	45
3.59. Nginx configuración RoundCube.	46
3.60. Nginx configuración previa para Django 01	46
3.61. Nginx configuración previa para Django 02	47
3.62. Nginx configuración Django	47
3.63. Configuración de certificados 01	48
3.64. Subdominios Ionos	48
3.65. Configuración de certificados 02	49
3.66. Configuración de certificados 03	49
3.67. Configuración de certificados 04	50
3.68. Configuración de certificados 05	50
3.69. Instalación de Active directory 01	51
3.70. Instalación de Active directory 02	51
3.71. Instalación de Active directory 03	52
3.72. Instalación de Active directory 04	52
3.73. Instalación de Active directory 05	53
3.74. Instalación de Active directory 06	53
3.75. Instalación de Active directory 07	54
3.76. Instalación de Active directory 08	54
3.77. Configuración del Active directory 01	55
3.78. Configuración del Active directory 02	56
3.79. Configuración del Active directory 03	56

3.80. Configuración del Active directory 04	57
5.1. Manual superadministrador 01	61
5.2. Manual técnico 02	62
5.3. Manual técnico 02	62
5.4. Manual técnico 02	63
5.5. Manual técnico 02	63
5.6. Manual técnico 02	63
5.7. Manual técnico 01	64
5.8. Manual técnico 02	64
5.9. Manual técnico 03	65
5.10. Manual técnico 04	65
5.11. Manual técnico 05	65
5.12. Manual técnico 06	66
5.13. Manual técnico 07	66
5.14. Manual técnico 08	66
5.15. Manual técnico 09	67
5.16. Manual técnico 10	67
5.17. Manual trabajador 01	68
5.18. Manual trabajador 02	68
5.19. Manual trabajador 03	69
5.20. Manual trabajador 04	69
5.21. Manual trabajador 05	69

1. Capítulo I: Introducción

1.1. Contextualización

En el entorno de trabajo, se observa que la administración de las cuatro empresas que forman parte del grupo empresarial (Para no usar el nombre de las empresas reales se ha decidido crear cuatro empresas ficticias), SolTeck, CyberWave, CoreRift y EmeralSolutions, no contaba con un sistema eficiente para el control de incidencias de las diferentes empresas. La falta de una herramienta centralizada para gestionar y dar seguimiento a los tickets de soporte, reportes de incidencias y solicitudes de servicio ha resultado en una gestión fragmentada y poco efectiva de los problemas que surgen en el día a día operativo de cada empresa.

Esta carencia ha generado inconvenientes significativos, como la falta de visibilidad sobre el estado actual de las incidencias, tiempos prolongados de resolución debido a la falta de seguimiento estructurado, y una comunicación limitada entre los departamentos y equipos involucrados en la solución de problemas. Además, la ausencia de datos organizados y analíticos sobre las incidencias ha dificultado la toma de decisiones informadas para mejorar los procesos y optimizar recursos.

Motivado por estos problemas, se plantea el desarrollo de una aplicación personalizada para gestionar tickets. El objetivo principal es implementar una solución integral que no solo centralice la gestión de incidencias de las empresas del grupo, sino que también mejore la eficiencia en la resolución de problemas, promueva una comunicación más fluida y transparente entre los departamentos y contribuya a una toma de decisiones basada en datos.

1.2. Alcance del Proyecto

El proyecto tiene como objetivo principal desarrollar una aplicación web personalizada para la gestión de incidencias, dirigida específicamente a las cuatro empresas del grupo empresarial ficticio (SolTeck, CyberWave, CoreRift y EmeralSolutions). El alcance del proyecto abarcará los siguientes aspectos:

1. Desarrollo de la Aplicación:

- Implementación de un sistema de gestión de tickets que permita registrar, asignar prioridades, dar seguimiento y cerrar incidencias de manera eficiente.
- Desarrollo de una interfaz de usuario intuitiva y amigable que facilite la navegación y el uso del sistema por parte de los usuarios finales.

2. Funcionalidades Específicas:

- Creación de un dashboard de análisis utilizando Plotly, que proporcione estadísticas detalladas sobre el número de tickets abiertos, cerrados, tiempos de resolución promedio y otras métricas relevantes.

- Integración de un sistema de correo electrónico interno utilizando tecnologías como Postfix y Roundcube, permitiendo la comunicación directa dentro de la aplicación entre los usuarios.

3. Infraestructura y Tecnología:

- Despliegue de la aplicación en un entorno basado en Linux para asegurar estabilidad, seguridad y escalabilidad.
- Integración con un servidor Windows Server utilizando Active Directory (AD) para la gestión centralizada de usuarios y la sincronización de credenciales.

4. Delimitaciones:

- El proyecto no incluirá el desarrollo de funcionalidades fuera del ámbito de gestión de incidencias, como gestión de recursos humanos o contabilidad.
- La integración con sistemas externos más allá de Active Directory no será parte del alcance inicial del proyecto.

1.3. Objetivos del proyecto

El presente proyecto tiene como objetivo principal desarrollar una solución integral de gestión de incidencias adaptada a las necesidades específicas de las empresas del grupo empresarial. A continuación se detallan los objetivos específicos que guiarán el desarrollo y la implementación de la aplicación:

- **Gestión personalizada de tickets:** Desarrollar una aplicación de gestión de tickets personalizada, adaptada a las herramientas, tecnologías y necesidades específicas de las empresas del grupo Heimdall.
- **Análisis de incidencias:** Implementar un dashboard para obtener análisis detallados de las incidencias a lo largo del tiempo, permitiendo una mejor gestión y toma de decisiones.
- **Registro de usuarios:** Utilizar el servicio de Active Directory para llevar a alta de usuarios en las diferentes empresas y comunicarlos directamente con la aplicación.
- **Manual de resolución:** Incluir manuales confeccionados por los técnicos para resolver problemas repetitivos de manera más eficiente, ahorrando tiempo y recursos.
- **Canal de comunicación:** Habilitar un canal de comunicación mediante un servicio de correos. Este sistema enviará correos automáticamente a los usuarios cuando se cree, confirme y resuelva un problema, mejorando la comunicación y seguimiento de incidencias.

Estos objetivos están diseñados para optimizar la gestión de incidencias dentro del grupo empresarial, mejorando la eficiencia operativa y facilitando una comunicación efectiva entre los equipos involucrados.

1.4. Planificación del proyecto

1.4.1. Desglose de tareas

En el proyecto se desglosará en las siguientes tareas y actividades principales:

1. **Análisis y Requisitos:** Realizar un relevamiento exhaustivo de los requisitos del proyecto, incluyendo los sistemas y servicios involucrados, la cantidad de logs generados, las necesidades de almacenamiento y las funcionalidades requeridas.
2. **Diseño de la Aplicación:** Diseño del sistema y la base de datos. Además del desarrollo de la interfaz de usuario y experiencia de usuario (UI/UX).
3. **Desarrollo e Implementación:** Implementación del sistema de gestión de tickets y dashboard de análisis junto al sistema de correo electrónico interno y configuración de seguridad.
4. **Pruebas y Ajustes:** Realización de pruebas unitarias, de integración y de aceptación. Seguido de ajustes basados en retroalimentación del usuario y pruebas de rendimiento.
5. **Despliegue y Puesta en Marcha:** Preparación del entorno de producción y configuración de servidores para el lanzamiento oficial de la aplicación.

1.4.2. Comunicación y Seguimiento

Se establecerán las siguientes prácticas de comunicación y seguimiento durante el proyecto:

- Reuniones regulares de seguimiento del equipo de proyecto.
- Uso de herramientas de gestión de proyectos (como Jira o Trello) para la asignación de tareas y seguimiento del progreso.
- Comunicación constante con los stakeholders para la validación de requisitos y avances del proyecto.
- Informes periódicos de avance y cumplimiento de hitos establecidos en el cronograma.

2. Capítulo II: Marco teórico

2.1. Python

[9]Python es un lenguaje de alto nivel de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma.

Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos).

Otro objetivo del diseño del lenguaje es la facilidad de extensión. Se pueden escribir nuevos módulos fácilmente en C o C++. Python puede incluirse en aplicaciones que necesitan una interfaz programable.

Aunque la programación en Python podría considerarse en algunas situaciones hostil a la programación funcional tradicional expuesta por Lisp, existen bastantes analogías entre Python y los lenguajes minimalistas de la familia Lisp (como Scheme).

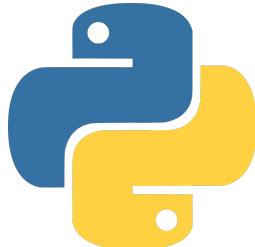


Figura 2.1: Python. Fuente: picpng.com

2.2. Django

[2]Django es un framework web Python de alto nivel que fomenta el desarrollo rápido y el diseño limpio y pragmático. Construido por desarrolladores experimentados, se encarga de gran parte de las complicaciones del desarrollo web, para que pueda centrarse en escribir su aplicación sin necesidad de reinventar la rueda. Es gratuito y de código abierto.



Figura 2.2: Django. Fuente: 1000logos.net

2.3. Plotly

[7] Plotly es una empresa de computación técnica con sede en Montreal, Quebec, que desarrolla herramientas en línea de análisis y visualización de datos. Plotly ofrece herramientas en línea de gráficos, análisis y estadísticas para individuos y colaboración, así como bibliotecas de gráficos científicos para Python, R, MATLAB, Perl, Julia, Arduino, JavaScript y REST.

Plotly ofrece productos de código abierto y empresariales.

- Dash es un framework de código abierto para Python, R y Julia que permite construir aplicaciones analíticas basadas en la web. Existen muchas bibliotecas especializadas de Dash, diseñadas para crear componentes y aplicaciones específicas de un dominio. Algunos ejemplos son Dash DAQ, para construir interfaces de adquisición de datos para usar con instrumentos científicos, y Dash Bio, que permite a los usuarios crear tipos de gráficos personalizados, herramientas de análisis de secuencias y herramientas de renderizado 3D para aplicaciones de bioinformática.
- Dash Enterprise es el producto de pago de Plotly para construir, probar, desplegar, gestionar y escalar aplicaciones Dash a nivel organizativo.
- Chart Studio Cloud es una herramienta gratuita en línea para crear gráficos interactivos. Tiene una interfaz gráfica de usuario de apuntar y hacer clic para importar y analizar datos en una cuadrícula y usar herramientas estadísticas. Los gráficos pueden ser incrustados o descargados.
- Chart Studio Enterprise es un producto de pago que permite a los equipos crear, estilizar y compartir gráficos interactivos en una sola plataforma. Ofrece opciones ampliadas de autenticación y exportación de archivos, y no limita el compartir y visualizar.
- Las bibliotecas de visualización de datos incluyen Plotly.js, una biblioteca de JavaScript de código abierto para crear gráficos, que también potencia Plotly.py para Python, así como Plotly.R para R, MATLAB, Node.js, Julia, Arduino y una API REST. Plotly también puede ser usado para estilizar gráficos interactivos con Jupyter notebook.
- Plotly ofrece convertidores de figuras que convierten gráficos de matplotlib, ggplot2 e IGOR Pro en gráficos interactivos y en línea.



Figura 2.3: Plotly. Fuente: wikipedia.org

2.4. Postfix

[8]Postfix es un servidor de correo de software libre, un programa informático para el enrutamiento y envío de correo electrónico, creado con la intención de que sea una alternativa más rápida, fácil de administrar y segura al ampliamente utilizado Sendmail. Anteriormente conocido como VMailer e IBM Secure Mailer, fue originalmente escrito por Wietse Venema durante su estancia en el Thomas J. Watson Research Center de IBM, y continúa siendo desarrollado activamente.

Postfix es el agente de transporte por omisión en diversas distribuciones de Linux y en las últimas versiones del Mac OS X. Para 2016 un estimado de 25 % de los servidores de correo electrónico a lo largo de todo el mundo utilizaba Postfix.



Figura 2.4: Postfix. Fuente: wikipedia.org

2.5. Dovecot

[3]Dovecot es un servidor de IMAP y POP3 de código abierto para sistemas GNU/Linux / UNIX-like, escrito teniendo en mente la seguridad. Desarrollado por Timo Sirainen, fue publicado por primera vez en julio del año 2002. Los objetivos principales de Dovecot es ser ligero, rápido, fácil de instalar y sobre todo seguro.

Dovecot puede trabajar con el estándar mbox, Maildir y sus propios formatos nativos dbox de alto rendimiento. Es completamente compatible con implementaciones de servidores UW IMAP y Courier IMAP, así como con clientes que accedan directamente a los buzones de correo.

Soporta compresión de mensajes (zlib y bzlib) así como deduplicación de adjuntos a través de la funcionalidad SiS (Single Instance Storage)

Se utiliza junto a Postfix para proporcionar autenticación de usuarios y autenticación de SMTP. Esto permite a los usuarios acceder a sus buzones de correo electrónico de manera segura y autenticada.



Figura 2.5: Dovecot. Fuente: wikipedia.org

2.6. Roundcube

[10]Roundcube es un cliente de correo para ver los mensajes de correo (correo electrónico) a través de una página web, pudiendo acceder desde cualquier navegador con acceso a internet. Desde él es posible realizar todas las operaciones necesarias para gestionar los correos e incluso usarlo como agenda de contactos y calendario.

Alguna de sus principales características son:

- Disponible en 65 idiomas.
- Drag-&-drop para gestionar los e-mails.
- Soporte completo para mensajes MIME y HTML.
- Sistema de protección de la privacidad sofisticado.
- Redactar mensajes con archivos adjuntos.
- Múltiples identidades del remitente.
- Composición de e-mails en HTML enriquecido.
- Reenviar mensajes con archivos adjuntos.
- Búsqueda de mensajes y contactos.
- Corrección ortográfica.
- Administración de carpetas IMAP.
- Soporte para servidores SMTP externos.
- Caché de acceso al buzón rápido.
- Número ilimitado de usuarios y mensajes.
- Plantilla sistema de skins personalizados.

Roundcube incluye otras sofisticadas librerías de código abierto como PEAR, una librería IMAP derivada de IlohaMail, el editor de texto enriquecido TinyMCE, la librería Googiespell para la corrección ortográfica y el saneador HTML5-PHP de Masterminds.

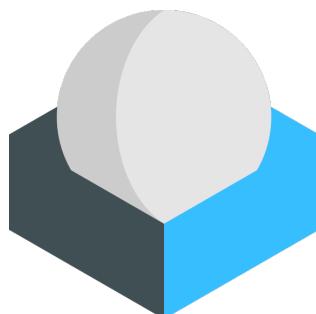


Figura 2.6: Roundcube. Fuente: roundcube.net

2.7. Active Directory

[4] Un directorio es una estructura jerárquica que almacena información sobre objetos en la red. Un servicio de directorio, como Active Directory Domain Services (AD DS), proporciona los métodos para almacenar datos de directorio y poner dichos datos a disposición de los usuarios y administradores de la red. Por ejemplo, AD DS almacena información acerca de las cuentas de usuario, como nombres, contraseñas, números de teléfono, etc., y permite que otros usuarios autorizados de la misma red tengan acceso a dicha información.

Active Directory almacena información acerca de los objetos de una red y facilita su búsqueda y uso por parte de los usuarios y administradores. Active Directory usa un almacén de datos estructurado como base para una organización jerárquica lógica de la información del directorio.

Este almacén de datos, también conocido como directorio, contiene información sobre los objetos de Active Directory. Estos objetos suelen incluir recursos compartidos como servidores, volúmenes, impresoras y cuentas de usuario y equipo de red.

La seguridad se integra en Active Directory mediante la autenticación de inicio de sesión y el control de acceso a los objetos del directorio. Con un único inicio de sesión de red, los administradores pueden administrar los datos del directorio y la organización a través de su red, y los usuarios de red autorizados pueden tener acceso a los recursos en cualquier parte de la red. La administración basada en directiva facilita la administración de incluso las redes más complejas.

Active Directory también incluye:

- Un conjunto de reglas, el esquema, que define las clases de objetos y atributos incluidos en el directorio, las restricciones y límites de las instancias de estos objetos y el formato de sus nombres.
- Un catálogo global que contiene información acerca de todos los objetos del directorio. Esto permite a los usuarios y administradores buscar información del directorio con independencia del dominio en que el directorio tiene los datos.
- Un mecanismo de consulta e índice para poder publicar los objetos y sus propiedades, y buscar por usuarios o aplicaciones de red.
- Un servicio de replicación que distribuye los datos de directorio en una red. Todos los controladores de dominio de un dominio participan en la replicación y contienen una copia completa de toda la información de directorio del dominio. Cualquier cambio en los datos del directorio se replica en todos los controladores de dominio del dominio.

2.7.1. Dominio

[11] Un Dominio es una colección de objetos dentro del directorio que forman un subconjunto administrativo. Pueden existir diferentes dominios dentro de un bosque, cada uno de ellos con su propia colección de objetos y unidades organizativas.

Para poner nombre a los dominios se utiliza el protocolo DNS. Por este motivo, Active Directory necesita al menos un servidor DNS instalado en la red.

2.7.2. Objeto

La palabra Objeto se utiliza como nombre genérico para referirnos a cualquiera de los componentes que forman parte del directorio, como una impresora o una carpeta compartida, pero también un usuario, un grupo, etc. Incluso podemos utilizar la palabra objeto para referirnos a una Unidad organizativa.

Cada objeto dispondrá de una serie de características específicas (según la clase a la que pertenezca) y un nombre que permitirá identificarlo de forma precisa.

En general, los objetos se organizan en tres categorías:

- Usuarios: identificados a través de un nombre (y, casi siempre, una contraseña), que pueden organizarse en grupos, para simplificar la administración.
- Recursos: que son los diferentes elementos a los que pueden acceder, o no, los usuarios según sus privilegios. Por ejemplo, carpetas compartidas, impresoras, etc.
- Servicios: que son las diferentes funciones a las que los usuarios pueden tener acceso. Por ejemplo, el correo electrónico.

Existen objetos que pueden contener a su vez otros objetos, como es el caso de los grupos de usuarios y de las unidades organizativas.

2.7.3. Controlador de dominio

Un Controlador de dominio (domain controller) contiene la base de datos de objetos del directorio para un determinado dominio, incluida la información relativa a la seguridad. Además, será responsable de la autenticación de objetos dentro de su ámbito de control.

En un dominio dado, puede haber varios controladores de dominio asociados, de modo que cada uno de ellos represente un rol diferente dentro del directorio. Sin embargo, a todos los efectos, todos los controladores de dominio, dentro del mismo dominio, tendrán la misma importancia.

2.7.4. Árboles

Un Árbol es simplemente una colección de dominios que dependen de una raíz común y se encuentran organizados como una determinada jerarquía. Dicha jerarquía también quedará representada por un espacio de nombres DNS común.

El objetivo de crear este tipo de estructura es fragmentar los datos del Directorio Activo, replicando sólo las partes necesarias y ahorrando ancho de banda en la red.

Si un determinado usuario es creado dentro de un dominio, éste será reconocido automáticamente en todos los dominios que dependan jerárquicamente del dominio al que pertenece.

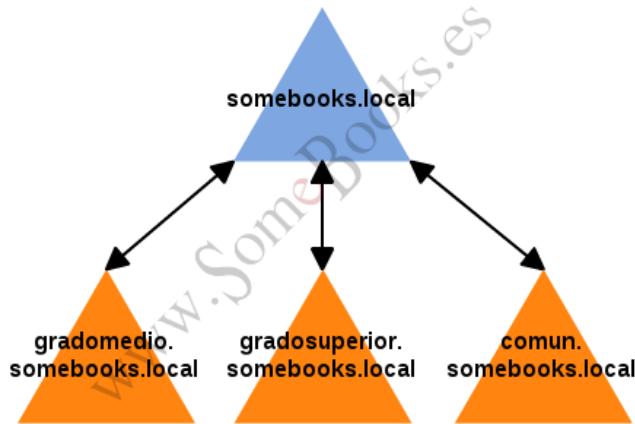


Figura 2.7: Árboles. Fuente: somebooks.es

2.7.5. Bosque

El Bosque es el mayor contenedor lógico dentro de Active Directory, abarcando a todos los dominios dentro de su ámbito. Los dominios están interconectados por Relaciones de confianza transitivas que se construyen automáticamente (consultar más adelante el concepto de Relación de confianza). De esta forma, todos los dominios de un bosque confían automáticamente unos en otros y los diferentes árboles podrán compartir sus recursos.

Como ya hemos dicho, los dominios pueden estar organizados jerárquicamente en un árbol que comparte un espacio de nombres DNS común. A su vez, diferentes árboles pueden estar integrados en un bosque. Al tratarse de árboles diferentes, no compartirán el mismo espacio de nombres.

De forma predeterminada, un bosque contiene al menos un dominio, que será el dominio raíz del bosque. En otras palabras: cuando instalamos el primer dominio, además del propio dominio, estamos creando la raíz de un nuevo árbol y también la raíz de un nuevo bosque.

El dominio raíz del bosque contiene el Esquema del bosque, que se compartirá con el resto de dominios que formen parte de dicho bosque (consultar el concepto de Esquema más adelante).

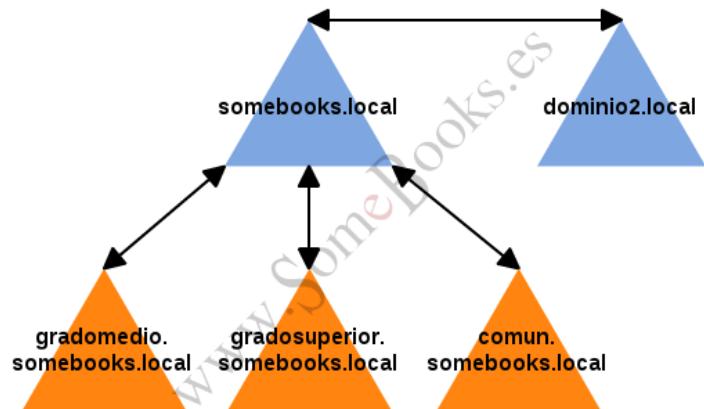


Figura 2.8: Bosque. Fuente: somebooks.es

2.7.6. Unidad Organizativa

Una Unidad Organizativa es un contenedor de objetos que permite organizarlos en subconjuntos, dentro del dominio, siguiendo una jerarquía. De este modo, podremos establecer una estructura lógica que represente de forma adecuada nuestra organización y simplifique la administración.

Otra gran ventaja de las unidades organizativas es que simplifican la delegación de autoridad (completa o parcial) sobre los objetos que contienen, a otros usuarios o grupos. Esta es otra forma de facilitar la administración en redes de grandes dimensiones.

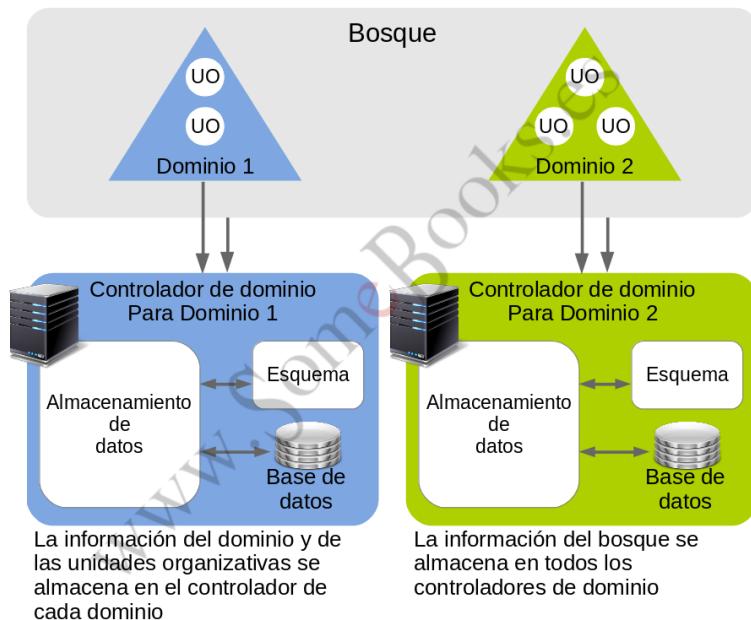


Figura 2.9: Unidad Organizativa. Fuente: somebooks.es

2.7.7. Relaciones de confianza

Las relaciones de confianza en Active Directory permiten a los usuarios de un dominio acceder a recursos en otro dominio y se clasifican en varios tipos con características específicas.

Las relaciones de confianza internas incluyen las predeterminadas, como las relaciones Padre-Hijo y Árbol-Raíz, que se crean automáticamente dentro de un bosque de AD y son bidireccionales y transitivas. Las relaciones de confianza externas conectan diferentes bosques de AD o dominios de Windows NT. Pueden ser bidireccionales y transitivas o no transitivas y unidireccionales o bidireccionales.

Las relaciones de confianza de respaldo (Shortcut Trusts) reducen los tiempos de autenticación entre dos dominios dentro del mismo bosque y son transitivas y pueden ser unidireccionales o bidireccionales.

Las características de las relaciones de confianza incluyen la bidireccionalidad, donde ambos dominios confían mutuamente, y la transitividad, que permite extender la confianza más allá de los dos dominios directamente involucrados.

2.8. Apache

[12] El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual según la normativa RFC 2616.

Apache es usado principalmente para enviar páginas web estáticas y dinámicas en la World Wide Web. Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web.

Apache es el componente de servidor web en la popular plataforma de aplicaciones LAMP, junto a MySQL y los lenguajes de programación PHP/Perl/Python (y ahora también Ruby). También es usado para muchas otras tareas donde el contenido necesita ser puesto a disposición en una forma segura y confiable. Un ejemplo es al momento de compartir archivos desde una computadora personal hacia Internet. Un usuario que tiene Apache instalado en su escritorio puede colocar arbitrariamente archivos en la raíz de documentos de Apache, desde donde pueden ser compartidos.



Figura 2.10: Apache. Fuente: wikipedia.org

2.9. Nginx

[6] Nginx es un servidor web/Proxy inverso ligero de alto rendimiento y un proxy para protocolos de correo electrónico (IMAP/POP3).

Nginx fue inicialmente desarrollado con el fin explícito de superar el rendimiento ofrecido por el servidor web Apache. Sirviendo archivos estáticos, Nginx usa dramáticamente menos memoria que Apache, y puede manejar aproximadamente cuatro veces más solicitudes por segundo. Este aumento de rendimiento viene con un costo de disminuida flexibilidad, como por ejemplo la capacidad de anular las configuraciones de acceso del sistema por archivo.



Figura 2.11: Apache. Fuente: wikipedia.org

2.10. MySQL

[5]MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, todo para entornos de desarrollo web. Es muy utilizada en aplicaciones web, como Joomla, Wordpress, Drupal o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL.

Se trata una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Sea cual sea el entorno en el que va a utilizar MySQL, es importante monitorizar de antemano el rendimiento para detectar y corregir errores tanto de SQL como de programación.



Figura 2.12: MySQL. Fuente: mysql.com

2.11. Certbot

[1]Certbot es parte del esfuerzo de la EFF para cifrar toda la Internet. La comunicación segura en la Web depende de HTTPS, que requiere el uso de un certificado digital que permite a los navegadores verificar la identidad de los servidores web. Los servidores web obtienen sus certificados de terceros de confianza llamados autoridades de certificación (CAs). Certbot es un cliente fácil de usar que obtiene un certificado de Let's Encrypt—una autoridad de certificación abierta lanzada por la EFF, Mozilla y otros—y lo despliega en un servidor web. Certbot y Let's Encrypt pueden automatizar el proceso de configurar un sitio web seguro y permitir activar y gestionar HTTPS con comandos simples.



Figura 2.13: Certbot. Fuente: github.com/certbot

3. Capítulo III: Fase de diseño

En esta fase se llevará a cabo un proceso meticuloso y estructurado orientado a la creación de una solución robusta y eficiente. Esta etapa es fundamental para asegurar que la aplicación no solo cumpla con los requisitos funcionales y técnicos establecidos, sino que también optimice la experiencia del usuario y maximice la eficiencia operativa dentro de la organización.

El diseño de la aplicación se centrará en varios aspectos clave: interfaz de usuario intuitiva que facilite la navegación y la interacción con el sistema de gestión de tickets, y la integración fluida de herramientas de análisis y comunicación interna. Cada decisión de diseño ha sido cuidadosamente evaluada para asegurar que contribuya a la mejora de la productividad y la efectividad en la resolución de incidencias.

Antes de adentrarnos en la fase de diseño de este proyecto es importante considerar los recursos humanos, hardware y software necesarios. Estos elementos son fundamentales para establecer una base sólida que respalde el éxito y la eficiencia del desarrollo.

En cuanto a los recursos humanos, se identificarán los roles claves necesarios para llevar a cabo el proyecto. En términos de hardware se procederá con la evaluación y selección de servidores adecuados para el despliegue de nuestra aplicación y el sistema de correo considerando recursos como la capacidad de procesamiento, almacenamiento y redundancia.

Atendiendo al software se planificará el uso de herramientas específicas para el desarrollo backend de la aplicación, generación de las gráficas interactivas, creación de un servidor de correos. La elección de estas herramientas se basará en su capacidad para cumplir con los requisitos funcionales y técnicos del proyecto, así como en su compatibilidad con la infraestructura existente y las capacidades de integración.

3.1. Recursos humanos.

Para el desarrollo del proyecto el equipo ideal debe de incluir:

- **Director de proyecto:** Encargado de la toma de decisiones estratégicas basadas en las opiniones del equipo y de asegurar el cumplimiento de los objetivos del proyecto.
- **Desarrollador Backend:** Responsable de la lógica de la aplicación Django y la integración de servicios.
- **Administrador de Base de Datos:** Encargado de diseñar y administrar la base de datos, asegurando la eficiencia y la integridad de los datos.
- **Desarrollador Frontend:** Responsable de la interfaz de usuario y la experiencia de usuario (UX/UI), utilizando tecnologías como HTML, CSS y posiblemente JavaScript.
- **Administrador de Sistemas:** Encargado de configurar y mantener el servidor de correos, el Windows server, la propia aplicación en django y el servicio de Open LDAP.

3.2. Recursos hardware.

Para este proyecto, a fin de hacerlo lo mas fiel a la realidad posible usare un servidor semejante al encontrado en las practicas, este sera: Dell PowerEdge T40, tambien contaremos con un SAI/UPS para evitar un repentino cese de los servidores y poder hacer un apagado ordenado del sistema, este sera CyberPower CP1500PFCLCD PFC Sinewave UPS System el cual cuenta con 25 min de autonomia tras una interrupcion de la corriente.

Las caracteristicas del servidor Dell PowerEdge T40 son:

- CPU: Intel Xeon E-2224G (4 núcleos, 3.5 GHz)
- RAM: 32 GB DDR4
- Almacenamiento: 1 Disco de 1 TB SSD
- Virtualización: VMware ESXi



Figura 3.1: Dell PowerEdge R740

Ademas este servidor cuenta con una gran escalabilidad y flexibilidad, permitiendo agregar más memoria y almacenamiento conforme crecen tus necesidades.

3.3. Recursos software.

Como sistemas operativos para los servidores virtualizados utilizaremos utilizado:

- **Servidor Linux:** Ubuntu 22.04 Live Server.
- **Servidor Windows:** Windows Server 2016.

Las herramientas y servicios software que serán necesarias para este proyecto son:

- Plotly [2.3]
- Nginx [2.9]
- Postfix [2.4]
- Roundcube [2.6]
- Dovecot [2.5]
- Django [2.2]
- Certbot [2.11]
- Active Directory [2.7]

3.4. Postfix

Para llevar a cabo la comunicación interna usare postfix, ya que este el servidor de correos utilizado en las practicas laborales.

Este nos pedirá una serie de configuraciones que deberemos de especificar, nuestras elecciones se guardaran en el fichero de configuración “/etc/postfix/main.cf”. Las configuraciones mas interesantes son las siguientes:

- “myhostname”: Nombre del host del servidor de correo.
- “myorigin”: Dirección de origen del correo enviado.
- “mydestination”: Dominios para los que Postfix aceptará correos.
- “mynetworks”: Redes que tienen permitido enviar correo a través de este servidor.
- “home_mailbox”: Define la ubicación del buzón de correo en el directorio de inicio del usuario.

```
GNU nano 6.2                                     /etc/postfix/main.cf

# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_tls_security_level=may

smtp_tls_CApth=/etc/ssl/certs
smtp_tls_security_level=may
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_unauth_destination
myhostname = dendorgames.es
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = $myhostname, dendorgames.es, localhost.localdomain, localhost
relayhost =
mynetworks = 127.0.0.0/8 [:ffff:127.0.0.0]/104 [:1]/128 192.168.1.0/24
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all
home_mailbox = Maildir/
```

Figura 3.2: Instalación de Postfix 03

A continuación seguiremos instalando S-nail^[1] el cual es un cliente de correo de línea de comandos utilizado para enviar y recibir correos electrónicos desde el terminal.

```
root@heimdall:~# apt install s-nail -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  s-nail
0 upgraded, 1 newly installed, 0 to remove and 10 not upgraded.
Need to get 572 kB of archives.
After this operation, 1149 kB of additional disk space will be used.
```

Figura 3.3: Instalación de Postfix 04

[1] S-nail: <https://wiki.archlinux.org/title/S-nail>

Tras instalarlo, reiniciamos el servicio de postfix.

```
root@heimdall:~# systemctl restart postfix
root@heimdall:~# systemctl status postfix
● postfix.service - Postfix Mail Transport Agent
  Loaded: loaded (/lib/systemd/system/postfix.service; enabled; vendor preset: enabled)
  Active: active (exited) since Thu 2024-06-06 18:16:30 UTC; 3s ago
    Docs: man:postfix(1)
   Process: 4921 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
 Main PID: 4921 (code=exited, status=0/SUCCESS)
    CPU: 3ms

Jun 06 18:16:30 heimdall systemd[1]: Starting Postfix Mail Transport Agent...
Jun 06 18:16:30 heimdall systemd[1]: Finished Postfix Mail Transport Agent.
root@heimdall:~# |
```

Figura 3.4: Instalación de Postfix 05

Para garantizar que la variable “MAIL” está configurada independientemente de cómo accede a su cuenta (ya sea a través de ssh, su, su- o sudo), deberá establecer la variable en el archivo “/etc/bash.bashrc” y añadirlo a un archivo en “/etc/profile.d” para asegurar que está configurado para todos los usuarios por defecto.

```
1 echo 'export MAIL=~/.Maildir' | sudo tee -a /etc/bash.bashrc | sudo tee -a /etc/profile.d/mail.sh
2 source /etc/profile.d/mail.sh
```

Comprobamos que la variable ha sido cargada correctamente.

```
albertomave@heimdall:~$ source /etc/profile.d/mail.sh
albertomave@heimdall:~$ echo $MAIL
/home/albertomave/.Maildir
albertomave@heimdall:~$ |
```

Figura 3.5: Instalación de Postfix 06

Antes de ejecutar el cliente, hay algunos ajustes que deberemos de configurar. Estos se encuentran en el archivo “/etc/s-nail.rc” al final del este, añadimos las siguientes opciones:

```
GNU nano 6.2                                     /etc/s-nail.rc
# s-it-mode
set emptystart
set folder=.Maildir
set record=+sent|
```

Figura 3.6: Instalación de Postfix 07

Probaremos a enviar un correo mediante s-nail y entraremos en el propio cliente para ver si el correo ha llegado correctamente.

```
albertomave@heimdall:~$ echo 'init' | s-nail -s 'init' -Snorecord albertomave
albertomave@heimdall:~$ s-nail
s-nail version v14.9.23. Type '?' for help
/home/albertomave/.Maildir: 1 message 1 new
►N 1 Alberto          2024-06-06 18:22  14/436  init
? |
```

Figura 3.7: Instalación de Postfix 08

3.5. Roundcube

Una vez que tenemos el servidor de correos instalado y configurado deberemos de ser capaces de servir los buzones en la web, para ello usaremos Roundcube, el cual usaban en mis prácticas en combinación con postfix.

Para llevar a cabo esta instalación deberemos de instalar primero LAMP-server.

LAMP es un acrónimo que se refiere a un conjunto de tecnologías de software libre que se utilizan para crear un servidor web.

- Linux: Un sistema operativo de código abierto y gratuito.
- Apache: Un servidor web de código abierto y gratuito, conocido por ser uno de los más populares y utilizados en el mundo.
- MySQL: Un gestor de bases de datos de código abierto y gratuito, ampliamente utilizado para almacenar y manejar grandes cantidades de datos.
- PHP: Un lenguaje de programación de código abierto y gratuito, ampliamente utilizado para crear aplicaciones web dinámicas y estáticas.

```
root@heimdall:~# apt install lamp-server"
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'libevent-pthreads-2.1-7' for task 'lamp-server'
Note, selecting 'libperl5.34' for task 'lamp-server'
Note, selecting 'libcurl4' for task 'lamp-server'
Note, selecting 'libtimedate-perl' for task 'lamp-server'
Note, selecting 'libhtml-tagset-perl' for task 'lamp-server'
Note, selecting 'libgdbm-compat4' for task 'lamp-server'
Note, selecting 'libssh-4' for task 'lamp-server'
Note, selecting 'ssl-cert' for task 'lamp-server'
Note, selecting 'libfcgi-perl' for task 'lamp-server'
Note, selecting 'perl' for task 'lamp-server'
Note, selecting 'libldap-common' for task 'lamp-server'
Note, selecting 'libaio1' for task 'lamp-server'
Note, selecting 'libaprutil1-dbd-sqlite3' for task 'lamp-server'
```

Figura 3.8: Instalación de Roundcube 01

Deberemos de instalar algunas librerías de PHP extras para que funcione correctamente, estos son:

```
1 sudo apt install php-xml php-json php-dom php-mbstring
```

Configuraremos el “data.timezone” para que este en nuestra zona (Europe/Madrid), esto lo configuraremos en el archivo “/etc/php/8.1/apache2/php.ini”.

```

GNU nano 6.2                                     /etc/php/8.1/apache2/php.ini *
;extension=xsl

zend_extension=opcache

; Module Settings ;

[CLI Server]
; Whether the CLI web server uses ANSI color coding in its terminal output.
cli_server.color = On

[Date]
; Defines the default timezone used by the date functions
; https://php.net/date.timezone
date.timezone = "Europe/Madrid"

; https://php.net/date.default-latitude
date.default_latitude = 31.7667

```

Figura 3.9: Instalación de Roundcube 02

Pasaremos ahora a instalar y configurar la base de datos. En este caso sera una base de datos Mysql.

```

root@heimdall:~# mysql_secure_installation
Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No:

Skipping password set for root as authentication with auth_socket is used by default.
If you would like to use password authentication instead, this can be done with the "ALTER_USER" command.
See https://dev.mysql.com/doc/refman/8.0/en/alter-user.html#alter-user-password-management for more information.

By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.

```

Figura 3.10: Instalación de Roundcube 03

```

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
- Dropping test database...
Success.

- Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
root@heimdall:~#

```

Figura 3.11: Instalación de Roundcube 04

Una vez instalado nuestro sistema gestor de bases de datos Mysql, nos conectaremos y crearemos una base de datos nueva, para el uso de RoundCube, de la siguiente manera:

```
root@heimdall:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.36-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> |
```

Figura 3.12: Instalación de Roundcube 05

```
mysql> create database roundcubedb;
Query OK, 1 row affected (0.01 sec)

mysql> createuser 'mailadmin' identified by 'mailadmin';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version
for the right syntax to use near 'createuser 'mailadmin' identified by 'mailadmin'' at line 1
mysql> create user 'mailadmin' identified by 'mailadmin';
Query OK, 0 rows affected (0.03 sec)

mysql> grant all privileges on roundcubedb.* to 'mailadmin';
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> |
```

Figura 3.13: Instalación de Roundcube 06

Después nos moveremos al directorio “/temp/” para mediante “wget” descargar el repositorio actualizado de roundcube y una vez descomprimido eliminarlo.

```
root@heimdall:~# cd /tmp/
root@heimdall:/tmp# wget https://github.com/roundcube/roundcubemail/releases/download/1.6.7/roundcubemail-1.6.7-complete
.tar.gz
--2024-06-06 18:34:54-- https://github.com/roundcube/roundcubemail/releases/download/1.6.7/roundcubemail-1.6.7-complete
.tar.gz
Resolving github.com (github.com)... 140.82.121.3
Connecting to github.com (github.com)|140.82.121.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/4224042/328c41e9-0555-4359-8924-6
5dbabb56b5b?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20240606%2Fus-east-1%2Fs3%2Faws4_
request&X-Amz-Date=20240606T183455Z&X-Amz-Expires=300&X-Amz-Signature=fbf8ee0645747fcc4ad3a15a51cd59f53aeecb751f1c086ac0
c4669b6dd0dd13c6X-Amz-SignedHeaders=host&actor_id=4224042&response-content-disposition=attachment%3B%2
0filename%3Droundcubemail-1.6.7-complete.tar.gz&response-content-type=application%2Foctet-stream [following]
--2024-06-06 18:34:55-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/4224042/328c41e9-0
555-4359-8924-65dbabb56b5b?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20240606%2Fus-east
-1%2Fs3%2Faws4_request&X-Amz-Date=20240606T183455Z&X-Amz-Expires=300&X-Amz-Signature=fbf8ee0645747fcc4ad3a15a51cd59f53ae
eeb751f1c086ac0c4669b6dd0dd13c6X-Amz-SignedHeaders=host&actor_id=4224042&response-content-disposition=
attachment%3B%20filename%3Droundcubemail-1.6.7-complete.tar.gz&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.109.133, 185.199.108.133, 185.199.111
.133...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5899345 (5.6M) [application/octet-stream]
Saving to: 'roundcubemail-1.6.7-complete.tar.gz'

roundcubemail-1.6.7-complete. 100%[=====] 5.63M 16.8MB/s in 0.3s
2024-06-06 18:34:56 (16.8 MB/s) - 'roundcubemail-1.6.7-complete.tar.gz' saved [5899345/5899345]
root@heimdall:/tmp# |
```

Figura 3.14: Instalación de Roundcube 07

Ahora vamos a descomprimirlo mediante “tar” en el directorio “/var/www/” de la siguiente manera.

```
root@heimdall:/tmp# tar -xzvf roundcubemail-1.6.7-complete.tar.gz -C /var/www/
roundcubemail-1.6.7/
roundcubemail-1.6.7/public_html/
roundcubemail-1.6.7/public_html/program/
roundcubemail-1.6.7/public_html/program/resources
roundcubemail-1.6.7/public_html/program/js
roundcubemail-1.6.7/public_html/skins
roundcubemail-1.6.7/public_html/plugins
roundcubemail-1.6.7/public_html/index.php
roundcubemail-1.6.7/public_html/.htaccess
roundcubemail-1.6.7/README.md
roundcubemail-1.6.7/SECURITY.md
roundcubemail-1.6.7/vendor/
```

Figura 3.15: Instalación de Roundcube 08

A continuación le cambiaremos el nombre a “webmail” y le daremos propietario al usuario “www-data” el cual nos permitirá servir la aplicación web con apache o nginx.

```
root@heimdall:/tmp# mv /var/www/roundcubemail-1.6.7/ /var/www/webmail
root@heimdall:/tmp# chown -R www-data:www-data /var/www/webmail/
root@heimdall:/tmp# chown -R www-data:www-data /var/www/webmail/*
root@heimdall:/tmp# ls -lart /var/www/webmail/
total 412
drwxrwxr-x 3 www-data www-data 4096 May 19 08:27 skins
drwxrwxr-x 3 www-data www-data 4096 May 19 08:27 public_html
drwxrwxr-x 8 www-data www-data 4096 May 19 08:27 program
drwxrwxr-x 37 www-data www-data 4096 May 19 08:27 plugins
drwxrwxr-x 2 www-data www-data 4096 May 19 08:27 logs
drwxrwxr-x 3 www-data www-data 4096 May 19 08:27 installer
```

Figura 3.16: Instalación de Roundcube 09

Ahora cargamos en la base de datos “roundcubedb”, la cual creamos anteriormente, las diferentes tablas que necesitamos para que nuestro webmail funcione correctamente.

```
root@heimdall:/tmp# mysql -u root -p roundcubedb < /var/www/webmail/SQL/mysql.initial.sql
Enter password:
root@heimdall:/tmp# systemctl restart apache2
root@heimdall:/tmp# |
```

Figura 3.17: Instalación de Roundcube 10

Podremos comprobar que todo ha salido bien entrando en la dirección web, pero nos saldrá este error:



Figura 3.18: Instalación de Roundcube 11

Esto es debido a que no hemos configurado en el archivo “000-default.conf” la ubicación “/etc/apache2/sites-available/”.

```
GNU nano 6.2                               /etc/apache2/sites-available/000-default.conf *
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/
```

Figura 3.19: Instalación de Roundcube 12

Simplemente deberemos de cambiar la variable “DocumentRoot” y establecerla como en la figura 3.19. Una vez llevado a cabo este paso, podremos acceder correctamente al webmail, para instalarlo, donde observaremos que hace falta instalar la librería php-Intl, proceso que llevaremos a cabo utilizando el gestor de paquetes “apt”.

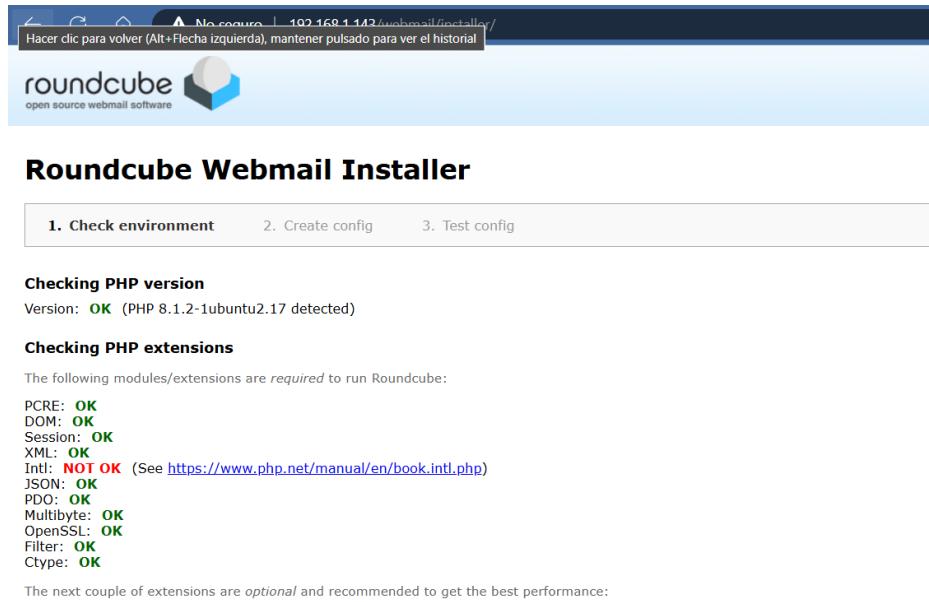


Figura 3.20: Instalación de Roundcube 13

Configuramos la base de datos MySQL del webmail.

Database setup

db_dsnw

Database settings for read/write operations:

MySQL	Database type	localhost	Database server (omit for sqlite)
localhost	Database name (use absolute path and filename for sqlite)	roundcubedb	
mailadmin	Database user name (needs write permissions) (omit for sqlite)	mailadmin	
mailadmin	Database password (omit for sqlite)	password	

db_prefix

Optional prefix that will be added to database object names (tables and sequences).

Figura 3.21: Instalación de Roundcube 14

Seguido de esto, estableceremos los servidores de IMAP y SMTP.

IMAP Settings

imap_host
localhost:143

The IMAP host(s) chosen to perform the log-in
Leave blank to show a textbox at login. To use SSL/STARTTLS connection add ssl:// or tls:// prefix. It can also contain the port number, e.g. tls://smtp.domain.tld:587.

username_domain

Automatically add this domain to user names for login
Only for IMAP servers that require full e-mail addresses for login

auto_create_user
 Automatically create a new Roundcube user when log-in the first time
A user is authenticated by the IMAP server but it requires a local record to store settings and contacts. Without this option the user must be created manually.
If this option is disabled, the login only succeeds if there's a matching user-record in the local Roundcube database.

sent_mbox
Sent

Store sent messages in this folder
Leave blank if sent messages should not be stored. Note: folder must include namespace prefix if any.

trash_mbox
Trash

Move messages to this folder when deleting them
Leave blank if they should be deleted directly. Note: folder must include namespace prefix if any.

drafts_mbox
Drafts

Store draft messages in this folder
Leave blank if they should not be stored. Note: folder must include namespace prefix if any.

junk_mbox
Junk

Store spam messages in this folder
Note: folder must include namespace prefix if any.

Figura 3.22: Instalación de Roundcube 15

SMTP Settings

smtp_host
localhost:25
Use this host for sending mails
To use SSL/STARTTLS connection add ssl:// or tls:// prefix. It can also contain the port number, e.g. tls://smtp.domain.tld:587.

smtp_user/smtp_pass
%u %p
SMTP username and password (if required)

Use the current IMAP username and password for SMTP authentication

smtp_log
 Log sent messages in {log_dir}/sendmail or to syslog.

Figura 3.23: Instalación de Roundcube 16

En el ultimo apartado de la instalación se verificará toda la configuración.

The screenshot shows the Roundcube Webmail Installer interface. At the top, there are navigation icons (back, forward, home) and a status bar indicating 'No seguro' and the URL '192.168.1.143/webmail/installer/index.php?_step=3'. Below this is the Roundcube logo with the text 'open source webmail software'. The main title 'Roundcube Webmail Installer' is displayed. A horizontal menu bar below the title has three items: '1. Check environment', '2. Create config', and '3. Test config', with '3. Test config' being the active tab. The main content area starts with a section titled 'Check config file' which lists two files: 'defaults.inc.php' and 'config.inc.php', both marked as 'OK'. A note below says 'OK, lazy people can download the updated config file here: [config.inc.php](#)'. The next section is 'Check if directories are writable', listing '/var/www/webmail/temp/' and '/var/www/webmail/logs/' both as 'OK'. The 'Check DB config' section lists 'DSN (write)', 'DB Schema', 'DB Write', and 'DB Time' all as 'OK'. Finally, the 'Test filetype detection' section lists 'Fileinfo/mime_content_type configuration' and 'Mimetype to file extension mapping' both as 'OK'.

Figura 3.24: Instalación de Roundcube 17

Hacemos un test de la configuración SMTP.

The screenshot shows a 'Test SMTP config' form. It has fields for 'Host' (localhost:25), 'Username' (albertomave), and 'Password' (empty). Below the form, it says 'Trying to send email...' followed by 'SMTP send: OK'. There are also fields for 'Sender' (albertomave@dendorgame) and 'Recipient' (pruebas@dendorgames.es), and a 'Send test mail' button.

Figura 3.25: Instalación de Roundcube 18

Comprobamos que llega al buzón, mediante S-nail.

```
pruebas@heimdall:~$ s-nail
s-nail version v14.9.23. Type `?' for help
/home/pruebas/Maildir: 1 message 1 new
►N 1 albertomave@dendor 2024-06-06 18:45 15/566 Test message from Roundcube
?
[-- Message 1 -- 15 lines, 566 bytes --]:
From: albertomave@dendorgames.es
To: pruebas@dendorgames.es
Subject: Test message from Roundcube
Message-Id: <20240606184523.EC1AA6077E@dendorgames.es>
Date: Thu, 6 Jun 2024 18:45:23 +0000 (UTC)

This is a test to confirm that Roundcube can send email.

? |
```

Figura 3.26: Instalación de Roundcube 19

A la hora de intentar iniciar sesión desde RoundCube usando IMAP este nos devolverá un error el cual podremos solucionar instalando dovecot [3.5.1](#).

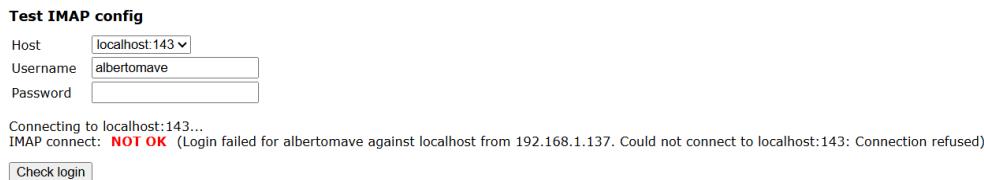


Figura 3.27: Instalación de Roundcube 20

3.5.1. Dovecot

Para solucionar el problema de iniciar sesión a través de RoundCube debemos de usar dovecot, software que instalaremos usando el gestor de paquetes “apt” [\[2\]](#).

```
root@heimdall:~# apt install dovecot-core dovecot-imapd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libexttextcat-2.0-0 libexttextcat-data
Suggested packages:
  dovecot-gssapi dovecot-ldap dovecot-lmtpd dovecot-lucene dovecot-managesieved dovecot-mysql dovecot-pgsql
  dovecot-pop3d dovecot-sieve dovecot-solr dovecot-sqlite dovecot-submissiond ntp ufw
The following NEW packages will be installed:
  dovecot-core dovecot-imapd libexttextcat-2.0-0 libexttextcat-data
0 upgraded, 4 newly installed, 0 to remove and 8 not upgraded.
Need to get 3705 kB of archives.
After this operation, 11.8 MB of additional disk space will be used.
Do you want to continue? [Y/n] |
```

Figura 3.28: Instalación de dovecot 01

Debemos de añadir a dovecot en el archivo de configuración de postfix, “/etc/postfix/-main.cf” y configurar el “master.cf” que se encuentra en el mismo directorio.

[2] APT: Advanced Package Tool.

```

smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
smtpd_sasl_local_domain = $myhostname
broken_sasl_auth_clients = yes
smtpd_recipient_restrictions = permit_sasl_authenticated, permit_mynetworks, reject_unauth_destination

```

Figura 3.29: Configuración de dovecot, main.cf 02

```

GNU nano 6.2                                     /etc/postfix/master.cf *
# =====#
# service type  private unpriv  chroot   wakeup   maxproc command + args
#   (yes)   (yes)     (no)    (never) (100)
# =====#
smtp      inet  n      -       y      -      -      smtpd
#smtp      inet  n      -       y      -      1      postscreen
#smtpd     pass  -      -       y      -      -      smtpd
#dnsblog   unix  -      -       y      -      0      dnsblog
#tlsproxy   unix  -      -       y      -      0      tlsproxy
# Choose one: enable submission for loopback clients only, or for any client.
#127.0.0.1:submission  inet n - y - - smtpd
submission inet n      -       y      -      -      smtpd
-o syslog_name=postfix/submission
-o smtpd_tls_security_level=encrypt
-o smtpd_sasl_auth_enable=yes
-o smtpd_client_restrictions=permit_sasl_authenticated,reject
-o milter_macro_daemon_name=ORIGINATING

smtps     inet  n      -       y      -      -      smtpd
-o syslog_name=postfix/smtps
-o smtpd_tls_wrappermode=yes
-o smtpd_sasl_auth_enable=yes
-o smtpd_client_restrictions=permit_sasl_authenticated,reject
-o milter_macro_daemon_name=ORIGINATING

```

Figura 3.30: Instalación de dovecot master.cf 03

Añadimos las siguientes líneas a los distintos archivos de configuración de dovecot, estos se encuentran en “/etc/dovecot/...”.

```

GNU nano 6.2                                     /etc/dovecot/dovecot.conf *
# source/destination IPs by placing the settings inside sections, for example:
# protocol imap { }, local 127.0.0.1 { }, remote 10.0.0.0/8 { }

# Default values are shown for each setting, it's not required to uncomment
# those. These are exceptions to this though: No sections (e.g. namespace {})
# or plugin settings are added by default, they're listed only as examples.
# Paths are also just examples with the real defaults being based on configure
# options. The paths listed here are for configure --prefix=/usr
# --sysconfdir=/etc --localstatedir=/var

# Enable installed protocols
!include_try /usr/share/dovecot/protocols.d/*.protocol

# A comma separated list of IPs or hosts where to listen in for connections.
# "*" listens in all IPv4 interfaces, "::" listens in all IPv6 interfaces.
# If you want to specify non-default ports or anything more complex,
# edit conf.d/master.conf.
[listen = *]

```

Figura 3.31: Configuración de dovecot, dovecot.conf 04

```

GNU nano 6.2                                     /etc/dovecot/dovecot.conf
!include conf.d/*.conf

# A config file can also tried to be included without giving an error if
# it's not found:
!include_try local.conf
mail_location = maildir:~/Maildir|

```

Figura 3.32: Configuración de dovecot, dovecot.conf 05

```

GNU nano 6.2                               /etc/dovecot/conf.d/10-auth.conf *
## Authentication processes
##
# Disable LOGIN command and all other plaintext authentications unless
# SSL/TLS is used (LOGINDISABLED capability). Note that if the remote IP
# matches the local IP (ie. you're connecting from the same computer), the
# connection is considered secure and plaintext authentication is allowed.
# See also ssl=required setting.
disable_plaintext_auth = no
auth_mechanisms = plain login

```

Figura 3.33: Configuración de dovecot, 10-auth.conf 06

```

GNU nano 6.2                               /etc/dovecot/conf.d/10-master.conf *
service auth-worker {
    # Auth worker process is run as root by default, so that it can access
    # /etc/shadow. If this isn't necessary, the user should be changed to
    # $default_internal_user.
    #user = root
}

service dict {
    # If dict proxy is used, mail processes should have access to its socket.
    # For example: mode=0660, group=vmail and global mail_access_groups=vmail
    unix_listener dict {
        #mode = 0600
        #user =
        #group =
    }
}

service auth {
    unix_listener /var/spool/postfix/private/auth {
        mode = 0660
        user = postfix
        group = postfix
    }
}

```

Figura 3.34: Configuración de dovecot, 10-master.conf 07

Una vez que hemos terminado de configurar dovecot, si volvemos a hacer la prueba de autentificación, esta tendrá éxito.

Test IMAP config

Host	localhost:143
Username	albertomave
Password	

Connecting to localhost:143...
IMAP connect: **OK** (SORT capability: yes)

Figura 3.35: Instalación de Roundcube 21

Para que en vez de la dirección ip aparezca nuestro dominio “dendorgames.es” deberemos de añadirlo en las configuraciones anteriores de smtp y imap.

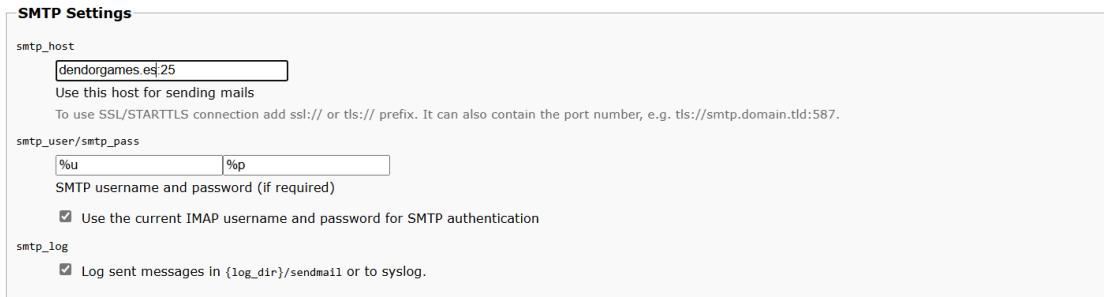


Figura 3.36: Instalación de Roundcube 24

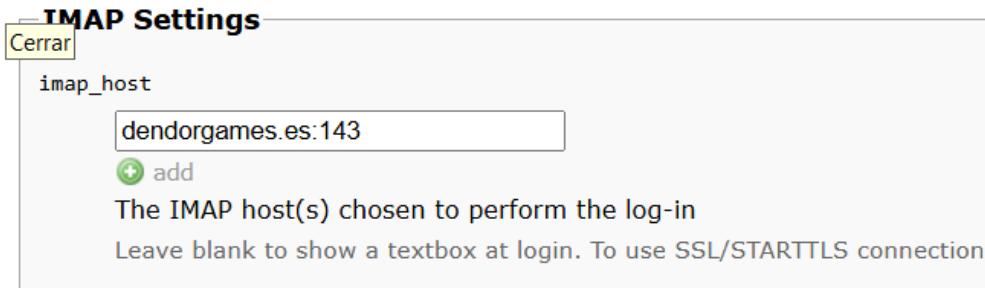


Figura 3.37: Instalación de Roundcube 25

Ahora podemos acceder a través de la dirección de “correo.dendorgames.es” y podremos ver nuestro buzón.

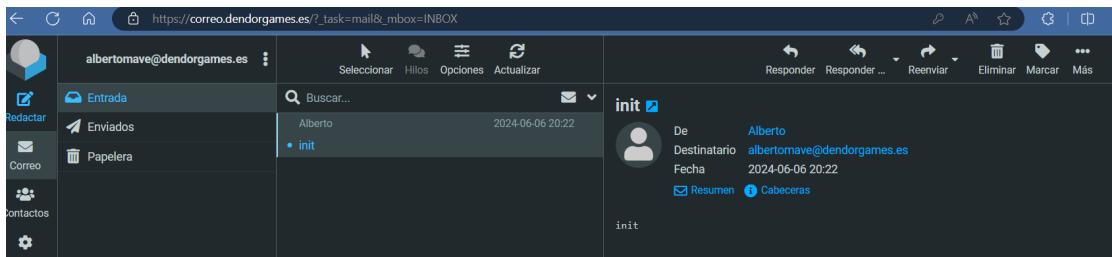


Figura 3.38: Instalación de Roundcube 26

Debemos de tener en cuenta que todas las direcciones, es decir, “correo.dendorgames.es”, “heimdall.dendorgames.es” y “dendorgames.es” se han dado de alta en el dominio y apuntan a las direcciones correspondientes, todo esto se configurará en la sección 3.8.

3.6. Django

Para empezar a crear nuestra aplicación web en django sera necesario crear un entorno virtual donde poder instalar librerías sin que entren en conflicto con otros entornos, esto es algo esencial cuando se empieza un nuevo proyecto en django.

Deberemos de situarnos en una carpeta donde queremos empezar nuestro nuevo proyecto y a continuación ejecutamos los siguientes comandos, comenzando por la creación del entorno virtual, activación del mismo y la instalación de los módulos necesarios para la elaboración de nuestra aplicación, que en nuestro caso serán los siguientes:

- **Django:** Es un framework web de alto nivel en Python que permite el desarrollo rápido y limpio de aplicaciones web.
- **crispy-bootstrap4:** Es un paquete que facilita la integración de Bootstrap 4 en formularios de Django. Permite aplicar estilos de Bootstrap a los formularios de manera sencilla y personalizada.
- **django-auth-ldap:** Proporciona autenticación basada en LDAP (Lightweight Directory Access Protocol) para aplicaciones Django. Permite integrar sistemas de autenticación externos, como Active Directory.
- **django-crispy-forms:** Es una aplicación Django que ayuda a controlar la presentación y el diseño de los formularios de manera más flexible. Facilita la creación de formularios HTML atractivos y personalizados.
- **numpy:** Es una biblioteca fundamental para la computación numérica en Python.
- **pandas:** Es una biblioteca para la manipulación y el análisis de datos en Python. Ofrece estructuras de datos y herramientas para trabajar con datos etiquetados y relacionales de manera eficiente.
- **plotly:** Es una biblioteca para crear gráficos interactivos y visualizaciones de datos.

```
1 python3 -m venv entorno_v
2 source entorno_v/bin/activate
3
4 pip3 install django django-crispy-forms crispy-bootstrap4 plotly pandas
5 numpy phonenumberslite django-auth-ldap
6
7 django-admin startproject heimdall
8 cd heimdall/
9 python3 manage.py startapp heimdallapp
10 python3 manage.py startapp Dashboard
```

Este sera el árbol de directorios que debemos encontrar al inicio de este nuevo proyecto:

```
(entorno_v) toor@empresa01:~/PFG$ tree heimdall/
heimdall/
├── Dashboard
│   ├── admin.py
│   ├── apps.py
│   ├── __init__.py
│   └── migrations
│       └── __init__.py
├── heimdall
│   ├── asgi.py
│   ├── __init__.py
│   ├── __pycache__
│   │   └── __init__.cpython-310.pyc
│   ├── settings.py
│   ├── urls.py
│   └── wsgi.py
└── heimdallapp
    ├── admin.py
    ├── apps.py
    ├── __init__.py
    ├── migrations
    │   └── __init__.py
    ├── models.py
    ├── tests.py
    └── views.py

    manage.py

6 directories, 22 files
(entorno_v) toor@empresa01:~/PFG$
```

Figura 3.39: Arbol de directorios mostrado usando "tree"

3.6.1. Modelos

A continuación tendremos que tener claro las tablas que vamos a crear en la base de datos, aunque estas podrán ir variando a medida que la aplicación vaya madurando, teniendo que añadir, quitar o modificar sus distintos campos ya que la deberemos de adaptar a todas nuestras necesidades.

En primer lugar comenzaremos usando la el fichero denominado “db.sqlite3” que se trata de un archivo de base de datos almacenado en el formato SQLite3, que es un motor de base de datos basado en SQL ligero y autónomo, haciendo uso de este fichero llevaremos a cabo el desarrollo y las distintas pruebas.

Una vez que el proyecto esté listo para el despliegue usaremos el sistema gestor de bases de datos MySQL. Este sistema gestor lo hemos utilizado para nuestra instalación de RoundCube y teniendo en cuenta que Django nos ofrece un soporte para diferentes sistemas gestores de bases de datos donde incluimos el mencionado anteriormente, evitamos instalar y configurar otro sistema gestor. Además, MySQL es una de las bases de datos más populares y rápidas para el desarrollo de aplicaciones web, siendo esta utilizada en proyectos famosos como Wikipedia, Google, Twitter, YouTube, etc. Cabe destacar que también MySQL es muy utilizado en aplicaciones web, como Joomla, Wordpress, Drupal o phpBB y por herramientas de seguimiento de errores como Bugzilla.

El esquema resultante para nuestra aplicación será el siguiente:

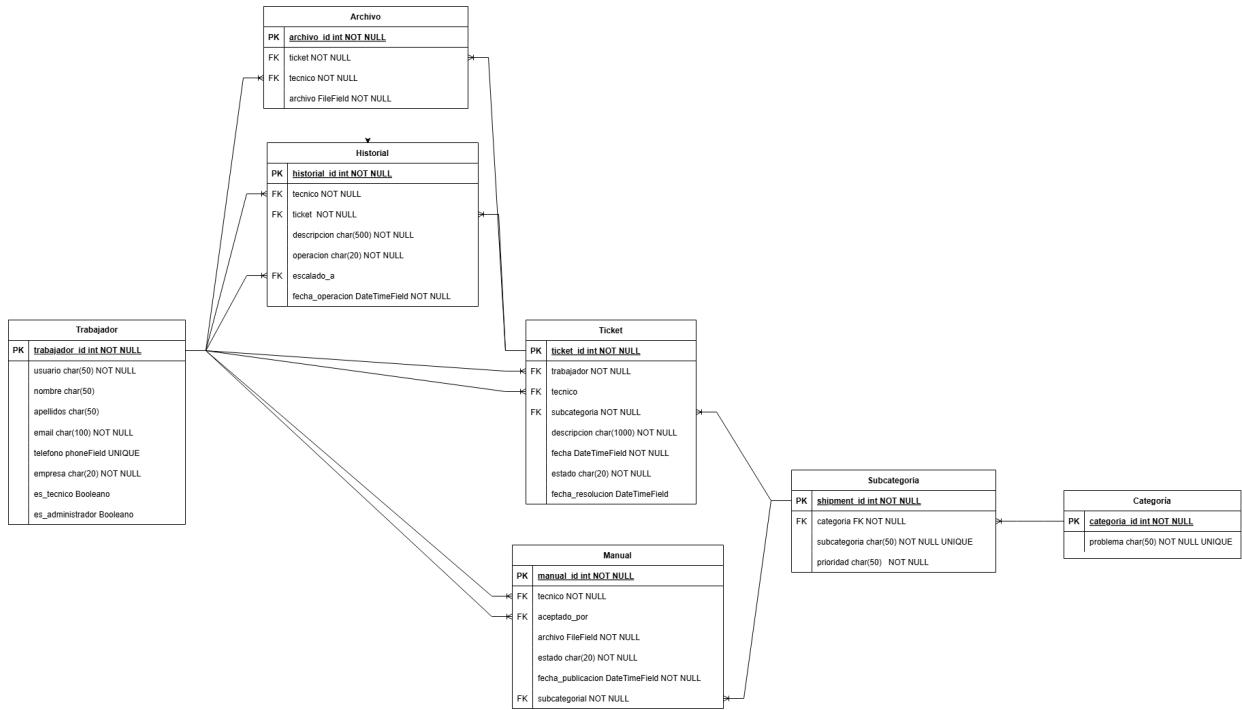


Figura 3.40: MR heimdall

Trabajador

En primer lugar, tenemos la entidad de Trabajador, este se extiende del “AbstractUser”, lo que significa que hereda las funcionalidades del modelo de usuario por defecto de Django.

- `es_tecnico`: Indica si el trabajador es un técnico.
- `es_administrador`: Indica si el trabajador es un administrador.
- `telefono`: Campo para el número de teléfono, del trabajador.
- `empresa`: Campo para seleccionar la empresa a la que pertenece el trabajador.

Categoría

Define las diferentes categorías en las que podrá situarse un ticket, esta tendrá un único campo que se denominará “problema” que indicará el nombre de la categoría.

Sub_categoria

Define subcategorías dentro de una categoría y su prioridad.

- `categoria`: Clave externa que hace referencia a la categoría perteneciente.
- `subcategoria`: Indica el nombre de la subcategoría.

- prioridad: Define el nivel de prioridad que va a asignar a la subcategoría, el valor de este campo podrá ser: Baja, Media o Alta.

Ticket

Esta entidad representará un ticket de incidencias, los campos que poseerá son:

- trabajador: Relación con el trabajador que ha creado el ticket.
- técnico: Relación con el técnico al cual se le asigna el ticket.
- subcategoría: Clave externa para identificar a que subcategoría y categoría pertenece el Ticket dando así la prioridad a este.
- descripción: Se trata de un campo donde un trabajador redactará el problema para que un técnico posea un contexto de la situación.
- fecha: Fecha de creación del ticket (se establece automáticamente cuando se crea).
- estado: Estado actual del ticket, nos indica la situación en la que se encuentra, las opciones están predefinidas y son: Creado, Confirmado, Aceptado, Escalado o Resuelto.
- fecha_resolución: Fecha de resolución del ticket (este campo se establecerá de forma automática cuando el técnico resuelva el ticket y lo marque como Resuelto).

Archivo

Representa los archivos adjuntos a los tickets.

- ticket: Relación con el ticket correspondiente.
- tecnico: Relación con el técnico que subió el archivo.
- archivo: Campo de archivo para subir documentos.

Historial

Gracias a esta entidad podremos llevar un control de las distintas operaciones que se han realizado sobre ticket y que técnicos han ayudado en su resolución.

- técnico: Relación con el técnico que realizó la operación.
- ticket: Relación con el ticket correspondiente.
- descripción: Descripción de la operación.
- operación: Tipo de operación realizada (Aceptado, Escalado, Resuelto).
- escalado_a: Técnico al que se ha escalado el ticket (opcional, se establece de forma automática si el técnico a llevado a cabo la operación de escalar).
- fecha_operación: Fecha de la operación (se establece automáticamente).

Documento

Representa documentos, estos servirán como manuales para los técnicos ya que contendrán información de como resolver ciertas incidencias. Hay que tener en cuenta que los documentos deben de ser validados por un administrador.

- técnico: Relación con el técnico que subió el documento.
- aceptado_por: Relación con el administrador que aceptó o rechazó el documento.
- archivo: Campo de archivo para subir el documento.
- estado: Estado actual del documento con opciones predefinidas (Creado, Aceptado, Rechazado).
- fecha_publicación: Fecha de publicación del documento (se establece automáticamente).
- subcategoría: Relación con la subcategoría del problema asociado al documento.

Es importante mencionar que este modelo ha ido variando a lo largo del desarrollo del proyecto, teniendo así que ajustarlo a las necesidades que iban surgiendo.

3.6.2. Vistas

En el framework Django, el archivo “views.py” es uno de los componentes más importantes de una aplicación. Su función principal es manejar la lógica de la vista, que se refiere a cómo se procesan las solicitudes HTTP (GET, POST, etc.) y cómo se preparan y envían las respuestas HTTP al navegador o al cliente que hizo la solicitud.

A continuación mostraremos las vistas mas importantes de la aplicación:

- Gestión de tickets
 - Punto de vista de trabajador: Los trabajadores y técnicos tienen la capacidad de crear tickets, proporcionando detalles como la subcategoría y el estado inicial del ticket. Una vez creado, el sistema asigna automáticamente el ticket a un técnico disponible.

Los trabajadores pueden ver una lista de sus tickets y tienen la opción de eliminar aquellos que aun no han sido confirmados, en el caso de que el ticket haya sido confirmado pierde el control sobre este, esto quiere decir que no puede realizar ninguna modificación adicional.

- Punto de vista de técnico: Estos tendrán un apartado para poder gestionar todos los tickets que se le asignen, también dispondrá de un historial donde pueden buscar y filtrar tickets utilizando diversos criterios. Además, podrán ver los detalles completos de los tickets.

Si el ticket se le ha asignado al técnico este podrá realizar acciones como adjuntar archivos relevantes, escalar el ticket a otro técnico y, finalmente, cerrar los tickets una vez resueltos, proporcionando una descripción de la solución implementada.

- Punto de vista del administrador: Además de tener acceso a todos los apartados correspondiente a los técnicos, los administradores tienen una visión completa

de todos los tickets pertenecientes a su empresa y pueden buscar y filtrar tickets utilizando diversos criterios, además tienen la capacidad de eliminar los tickets (pertenecientes a su empresa) si estos no son adecuados o están duplicados.

- Punto de vista del súper administrador: Este podrá ver todos los tickets activos en el sistema pudiendo eliminarlos, también contará con la posibilidad de gestionar tickets al igual que los técnicos, la diferencia reside en que los tickets que el súper administrador debe gestionar son los que se sitúan en la categoría “Incidencias Locales”.

- Gestión de documentos

- La aplicación permite la subida de documentos técnicos, que pueden estar relacionados con subcategorías específicas. Los técnicos pueden subir documentos que necesitan pero estos deben de ser aprobados por un administrador o súper administrador antes de estar disponibles para otros usuarios. Los administradores pueden gestionar estos documentos, aprobándolos o rechazándolos según su relevancia y calidad.

- Permisos de acceso

- Trabajadores: Pueden crear, confirmar, listar y eliminar sus propios tickets sin confirmar.
- Técnicos: Tienen capacidades más avanzadas, como aceptar, escalar y cerrar tickets, además de adjuntar archivos y ver información de otros técnicos.
- Administradores: Tienen control total sobre la gestión de tickets de su empresa, también tienen control total sobre los documentos, incluyendo la capacidad de aprobar o rechazar documentos subidos por técnicos.

- Roles

- Trabajadores: Se centran en la creación y gestión básica de sus propios tickets.
- Técnicos: Manejan la resolución de tickets, escalado y gestión de archivos adjuntos, asegurando que los tickets se resuelvan de manera eficiente.
- Administradores: Supervisan la gestión global de tickets pertenecientes a su empresa y documentos, garantizando que se sigan los estándares de calidad y que todos los documentos relevantes estén disponibles
- Súper Administrador: Se encarga de manejar todo el sistema de tickets, también gestionará el servidor de correos y la gestión de usuarios en Active Directory.

Añadir que tanto los técnicos, administradores y el superadministrador tendrán acceso al dashboard, donde podrán controlar la cantidad de tickets sin confirmar, sin resolver y resueltos. Además, podrán sacar conclusiones utilizando las diferentes gráficas disponibles en este.

El dashboard permitirá un monitoreo en tiempo real, análisis detallado y personalización de vistas, mejorando la eficiencia operativa y la toma de decisiones informadas.

3.6.3. URL

En el desarrollo de aplicaciones web, la gestión de las URLs (Uniform Resource Locators) juega un papel fundamental para asegurar una navegación eficiente y coherente. En el contexto de una aplicación gestora de tickets desarrollada con Django, las URLs son cruciales para mapear las solicitudes del usuario a las vistas adecuadas, permitiendo así la creación, visualización, actualización y eliminación de tickets de manera intuitiva y ordenada.

Django nos facilita la gestión de URLs mediante un sistema potente y flexible. Este sistema permite definir rutas claras y legibles, lo que mejora tanto la experiencia del usuario como la mantenibilidad del código. En nuestra aplicación gestora de tickets, hemos estructurado las URLs de manera que reflejen las diferentes funcionalidades disponibles, como la creación de nuevos tickets, la visualización de la lista de tickets existentes, la edición de detalles de un ticket específico y la eliminación de tickets no deseados.

Cada URL en nuestra aplicación se asocia con una vista particular, que es responsable de manejar la lógica correspondiente. Por ejemplo, una URL destinada a la creación de un nuevo ticket podría estar vinculada a una vista que despliega un formulario para ingresar los detalles del ticket, mientras que una URL destinada a la visualización de un ticket específico estará vinculada a una vista que recupera y muestra los detalles de ese ticket en particular.

A continuación, se presentan las principales URLs utilizadas en nuestra aplicación, junto con una breve descripción de su propósito y la vista a la que están asociadas:

- Estas son las URLs generales del proyecto, las cuales nos llevará al panel administrativo de django, de las landing page de los usuarios, acceso a la autentificación de django y al manejo de las contraseñas de django.

```
urlpatterns = [
    path('puente/admin/', admin.site.urls),
    path('', include('heimdallapp.urls','heim'),namespace="heim"),
    path('', include('Dashboard.urls','dash'),namespace="dash"),
    path("accounts/", include("django.contrib.auth.urls")),

    path('password_reset/done/', RedirectPasswordResetDoneView.as_view(), name='password_reset_done'),
    path('password_change/done/', RedirectPasswordChangeDoneView.as_view(), name='password_change_done'),
] + static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)

urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

Figura 3.41: Django url 01

- En el Dashboard solo tendremos la url de la raíz, que sera usada como landing page de los técnicos y administradores.

```
urlpatterns = [
    #DASHBOARD
    path('', Dash.as_view(), name='dashboard'),
]
```

Figura 3.42: Django url 02

- Esta es la parte mas interesante del proyecto, ya que tienen las URLs a través de las cuales los usuarios viajaran para hacer uso de la aplicación. Entre ellas tenemos para ver, crear, escalar, modificar y eliminar tanto tickets, como archivos, como manuales. Una url que al final descartamos fue la de “usuarios/gestión/” esta permitía al administrador de django modificar directamente los permisos de un trabajador, con la implementación de los usuarios en el AD esta función de la aplicación ya no tenia razón para existir.

```

urlpatterns = [
    # FUNCIONES
    path('informacion/obtener_subcategorias/', obtener_subcategorias, name='informacion_subcategorias'),
    # PERFILES
    path('perfil/<slug:username>', Perfiles.as_view(), name='perfiles'),

    #TRABAJADOR
    path('trabajador/', ListadoTrabajador.as_view(), name='listado_ticket_trabajador'),
    path('crear-ticket/', CrearTicket.as_view(), name='crear_ticket'),
    path('crear-ticket/confirmar/<int:pk>', ConfirmacionTicket.as_view(), name='confirmar_ticket'),
    path('eliminar-ticket/<int:pk>', EliminarTicket.as_view(), name='eliminar_ticket'),

    # VISTA DE TECNICOS
    path('tecnico/', ListadoTecnico.as_view(), name='listado_ticket_tecnico'),
    path('ticket/<int:pk>', DetalleTicket.as_view(), name='detalle_ticket_tecnico'),
    path('ticket/escalar/<int:pk>', EscalarTicket.as_view(), name='escalar_ticket_tecnico'),
    path('ticket/registro-cambios/<int:pk>', RegistroCambios.as_view(), name='registro_ticket_tecnico'),
    path('ticket/archivos/<int:pk>', ArchivosTicket.as_view(), name='archivos_ticket_tecnico'),
    path('tickets/', HistorialTicket.as_view(), name='historial_ticket'),
    path('ticket/cerrar/<int:pk>', CerrarTicket.as_view(), name='cerrar_ticket'),
    path('ticket/archivos/eliminar/<int:pk>', EliminarArchivo.as_view(), name='eliminar_archivo_ticket'),
    path('informacion/tecnicos/', InformacionTecnicos.as_view(), name='informacion_tecnicos'),
    # DOCUMENTOS
    path('manuales/', Documentos.as_view(), name='documentos'),
    path('manuales/subir/', SubirDocumentos.as_view(), name='subir_documentos'),

    #LISTADO ADMINISTRADORES
    path('administrador/', ListadoAdministrador.as_view(), name='listado_ticket_administrador'),
    path('manuales/gestion/', GestionarDocumentos.as_view(), name='gestionar_documentos'),
    path('tickets/gestion/', AdministrarTickets.as_view(), name='gestionar_tickets'),
    path('tickets/gestion/eliminar/<int:pk>', EliminarTicketAdmin.as_view(), name='gestionar_ticket_eliminar'),
    # path('usuarios/gestion/', GestionarUsuarios.as_view(), name='gestionar_usuarios'),
]

```

Figura 3.43: Django url 03

3.6.4. Settings

El archivo “settings.py” es el núcleo de configuración de cualquier proyecto Django. Este archivo centraliza la configuración de toda la aplicación, definiendo cómo debe comportarse y qué recursos utilizar. En el contexto de una aplicación gestora de tickets, “settings.py” es fundamental para establecer las configuraciones de base de datos, rutas estáticas, autenticación, y otras configuraciones específicas que aseguran el correcto funcionamiento de la aplicación.

Django nos ofrece un sistema de configuración flexible y extensible a través del archivo “settings.py”. Este archivo incluye una variedad de parámetros que permiten personalizar aspectos críticos del proyecto, tales como:

- Configuración de la Base de Datos: Definir la base de datos que se utilizaremos MySQL, las credenciales de acceso y otros parámetros relacionados.
- Configuración de Aplicaciones Instaladas: Listar todas las aplicaciones Django y aplicaciones de terceros que están habilitadas en el proyecto.

- Gestión de Archivos Estáticos y Medios: Definir las rutas y configuraciones para manejar archivos estáticos (como CSS, JavaScript) y archivos multimedia (como imágenes subidas por usuarios).
- Configuración de Seguridad: Establecer parámetros relacionados con la seguridad, tales como las claves secretas, los dominios permitidos, y las políticas de autenticación.
- Configuración de Internacionalización: Ajustar la configuración relacionada con el idioma y la localización de la aplicación.

A lo largo de este trabajo, exploraremos las configuraciones más relevantes del archivo “settings.py” de nuestra aplicación gestora de tickets, destacando cómo se han ajustado estos parámetros para satisfacer las necesidades específicas del proyecto. Asimismo, se proporcionarán ejemplos concretos de configuraciones utilizadas, ilustrando cómo cada sección del archivo “settings.py” contribuye al funcionamiento integral de la aplicación.

A continuación, se presentan las principales configuraciones del archivo “settings.py” de nuestra aplicación, junto con una breve descripción de su propósito y su impacto en el proyecto.

```
heimdall > ⚡ settings.py > ...
1 import ldap
2 from django_auth_ldap.config import LDAPSearch, LDAPSearchUnion, GroupOfNamesType
3 from pathlib import Path
4 from os.path import join, dirname
5 import os
6 from dotenv import load_dotenv
7
8 dotenv_path = join(dirname(__file__), '.env')
9 load_dotenv(dotenv_path)
10
11 # Build paths inside the project like this: BASE_DIR / 'subdir'.
12 BASE_DIR = Path(__file__).resolve().parent.parent
13
14
15 # Quick-start development settings - unsuitable for production
16 # See https://docs.djangoproject.com/en/5.0/howto/deployment/checklist/
17
18 # SECURITY WARNING: keep the secret key used in production secret!
19 SECRET_KEY = [REDACTED]
20
21 # SECURITY WARNING: don't run with debug turned on in production!
22 DEBUG = False
23 ALLOWED_HOSTS = ["*"]
24
```

Figura 3.44: Django configuración settings.py 01

En esta figura hemos podido ver los diferentes módulos que usamos, destacando el uso de “dotenv” para cargar datos sensibles de manera segura a través de un archivo .env con los permisos mínimos necesarios. También se encuentra deshabilitado el modo debug para producción y se permite la conexión desde cualquier dirección.

```

# Application definition

INSTALLED_APPS = [
    "phonenumbers_field",
    'crispy_bootstrap4',
    'crispy_forms',
    'heimdallapp.apps.heimdallappConfig',
    'Dashboard.apps.DashboardConfig',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]

```

Figura 3.45: Django configuración settings.py 02

Hablando sobre las aplicaciones instaladas, destacar HeimdallApp y Dashboard que son las aplicaciones que hemos desarrollado, además hemos añadido ciertas aplicaciones de terceros necesarias como crispy_forms y phonenumbers_field.

```

DATABASES = {
    "default": {
        "ENGINE": "django.db.backends.mysql",
        "OPTIONS": {
            "read_default_file": "/etc/mysql/my.cnf",
        },
    }
}

```

Figura 3.46: Django configuración settings.py 03

En la figura 3.46 se puede observar como se configura Django para ser utilizado con MySQL. En el fichero “/etc/mysql/my.cnf” se deberá de establecer la base de datos junto al usuario y sus credenciales para que Django pueda hacer uso de ellos.

```

41 CRISPY_TEMPLATE_PACK = 'bootstrap4'
42 MIDDLEWARE =
43     'django.middleware.security.SecurityMiddleware',
44     'django.contrib.sessions.middleware.SessionMiddleware',
45     'django.middleware.common.CommonMiddleware',
46     'django.middleware.csrf.CsrfViewMiddleware',
47     'django.contrib.auth.middleware.AuthenticationMiddleware',
48     'django.contrib.messages.middleware.MessageMiddleware',
49     'django.middleware.clickjacking.XFrameOptionsMiddleware',
50 ]
51
52 ROOT_URLCONF = 'heimdall.urls'
53
54 TEMPLATES = [
55     {
56         'BACKEND': 'django.template.backends.django.DjangoTemplates',
57         'DIRS': ['templates'],
58         'APP_DIRS': True,
59         'OPTIONS': {
60             'context_processors': [
61                 'django.template.context_processors.debug',
62                 'django.template.context_processors.request',
63                 'django.contrib.auth.context_processors.auth',
64                 'django.contrib.messages.context_processors.messages',
65             ],
66         },
67     },
68 ]
69

```

Figura 3.47: Django configuración settings.py 04

En la figura 3.47 podemos destacar la configuración de la ubicación de las plantillas (templates). Django necesita una forma conveniente de generar HTML dinámicamente. El enfoque más común se basa en plantillas. Una plantilla contiene las partes estáticas de la salida HTML deseada, así como alguna sintaxis especial que describe cómo se insertará el contenido dinámico.

```

# Internationalization
# https://docs.djangoproject.com/en/5.0/topics/i18n/

LANGUAGE_CODE = 'es-ES'

TIME_ZONE = 'Europe/Madrid'

USE_I18N = True

USE_TZ = True

AUTH_USER_MODEL = 'heimdallapp.Trabajador'

```

Figura 3.48: Django configuración settings.py 05

Destacando más configuraciones, podemos hablar de el lenguaje y la zona horaria. También, en la figura 3.48, se establece la autenticación en Django.

```

115
116 # Static files (CSS, JavaScript, Images)
117 # https://docs.djangoproject.com/en/5.0/howto/static-files/
118
119 #STATICFILES_DIRS = [os.path.join(BASE_DIR, "src/static")]
120 STATIC_URL = 'static/'
121 STATIC_ROOT = 'src/static/'
122 MEDIA_URL = 'media/'
123 MEDIA_ROOT = 'src/media/'
124
125 LOGIN_REDIRECT_URL = "/"
126 LOGOUT_REDIRECT_URL = "/"

```

Figura 3.49: Django configuración settings.py 06

Otro paso importante en el fichero “settings.py” es establecer las diferentes variables relacionadas con los ficheros estáticos (css,js,etc) y los ficheros media. Dependiendo de como se esté desplegando la aplicación esto varía, es decir, no se deben establecer de la misma manera cuando la aplicación se sirve con nginx y cuando se sirve de forma local haciendo uso de la instrucción “runserver”.

```

EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_USE_TLS = False
EMAIL_HOST = os.environ.get("EMAIL_HOST")
EMAIL_PORT = os.environ.get("EMAIL_PORT")
EMAIL_HOST_USER = os.environ.get("EMAIL_HOST_USER")
EMAIL_HOST_PASSWORD = os.environ.get("EMAIL_HOST_PASSWORD")

DEFAULT_FROM_EMAIL = os.environ.get("DEFAULT_FROM_EMAIL")

BASE_CORREOS_TEMPLATE = os.path.join(BASE_DIR, "heimdallapp/correos/")

```

Figura 3.50: Django configuración settings.py 07

Otra configuración importante será el servidor de correo, para que Django pueda hacer uso de este y enviar correos de forma automática con la creación de tickets, documentos, etc, debemos de establecer las diferentes variables que se ven en la figura 3.50 (como se ha mencionado anteriormente, los datos sensibles permanecen ocultos).

```

143 AUTHENTICATION_BACKENDS = [
144     "django_auth_ldap.backend.LDAPBackend",
145     # "django.contrib.auth.backends.ModelBackend",
146 ]
147
148 AUTH_LDAP_SERVER_URI = os.environ.get("AUTH_LDAP_SERVER_URI")
149 AUTH_LDAP_BIND_DN = os.environ.get("AUTH_LDAP_BIND_DN")
150 AUTH_LDAP_BIND_PASSWORD = os.environ.get("AUTH_LDAP_BIND_PASSWORD")
151 AUTH_LDAP_USER_SEARCH = LDAPSearchUnion(
152     LDAPSearch("ou=Solteck,dc=dmheim,dc=dendorgames,dc=es", ldap.SCOPE_SUBTREE, "(sAMAccountName=%(user)s"),
153     LDAPSearch("ou=CoreRift,dc=dmheim,dc=dendorgames,dc=es", ldap.SCOPE_SUBTREE, "(sAMAccountName=%(user)s"),
154     LDAPSearch("ou=Emeralsolutions,dc=dmheim,dc=dendorgames,dc=es", ldap.SCOPE_SUBTREE, "(sAMAccountName=%(user)s"),
155     LDAPSearch("ou=cyberwave,dc=dmheim,dc=dendorgames,dc=es", ldap.SCOPE_SUBTREE, "(sAMAccountName=%(user)s"))
156 )

```

Figura 3.51: Django configuración settings.py 08

Como haremos uso de Active Directory tenemos que establecer la configuración de conexión y búsqueda de usuarios en las diferentes unidades organizativas. Las diferentes variables que debemos configurar se pueden observar en la figura 3.51.

```

AUTH_LDAP_USER_ATTR_MAP = {
    "email": "mail",
    "first_name": "givenName",
    "last_name": "sn",
    "telefono": "telephoneNumber"
}

AUTH_LDAP_GROUP_SEARCH = LDAPSearch(
    "dc=dmheim,dc=dendorgames,dc=es", ldap.SCOPE_SUBTREE, "(objectClass=group)"
)
AUTH_LDAP_GROUP_TYPE = GroupOfNamesType()
AUTH_LDAP_USER_FLAGS_BY_GROUP = {
    "es_tecnico": ["cn=Tecnico,dc=dmheim,dc=dendorgames,dc=es",
                   "cn=DjangoAdministrador,dc=dmheim,dc=dendorgames,dc=es"],
    "es_administrador": "cn=DjangoAdministrador,dc=dmheim,dc=dendorgames,dc=es",
    "is_staff": "cn=SuperAdministrador,dc=dmheim,dc=dendorgames,dc=es",
    "is_superuser": "cn=SuperAdministrador,dc=dmheim,dc=dendorgames,dc=es",
}
AUTH_LDAP_CACHE_GROUPS = False
AUTH_LDAP_CACHE_USERS = False
AUTH_LDAP_CACHE_TIMEOUT = 0
AUTH_LDAP_ALWAYS_UPDATE_USER = True

```

Figura 3.52: Django configuración settings.py 09

Adicionalmente, en los fragmentos de configuración de la figura 3.52, se establecen la integración con Active Directory para autenticación de usuarios y gestión de permisos. AUTH_LDAP_USER_ATTR_MAP asigna atributos como correo electrónico, nombre y apellido a campos correspondientes en el modelo de usuario de Django.

AUTH_LDAP_GROUP_SEARCH configura la búsqueda de grupos de Active Directory y AUTH_LDAP_GROUP_TYPE define el tipo de grupo que se espera encontrar.

AUTH_LDAP_USER_FLAGS_BY_GROUP asigna permisos a usuarios según los grupos de Active Directory a los que pertenecen, permitiendo gestionar roles como técnico, administrador y superadministrador de forma centralizada y basada en Active Directory.

3.6.5. Señales

En Django, las señales son una herramienta esencial para permitir la comunicación automática entre diferentes partes de una aplicación. Utilizando el patrón de diseño observador, las señales permiten desencadenar acciones en respuesta a ciertos eventos o cambios de estado en el sistema. En este proyecto, utilizaremos señales para automatizar procesos críticos como el envío de correos electrónicos y la actualización de credenciales de usuario. Esto no solo mejora la eficiencia y la capacidad de respuesta de la aplicación, sino que también garantiza que estas tareas se realicen de manera consistente y sin intervención manual. En esta sección, exploraremos cómo implementar estas señales para optimizar la funcionalidad de nuestra aplicación.

```
@receiver(post_save, sender=Ticket)
def enviar_correo(sender, instance, created, **kwargs):
    if instance.estado == 'Creado':
        with open(f'{settings.BASE_CORREOS_TEMPLATE}ticket_creado.txt', "r") as fichero:
            mensaje = fichero.read()

        enlace = f"https://heimdall.dendorgames.es/crear-ticket/confimar/{instance.id}"
        mensaje_formateado = mensaje.format(
            usuario_nombre=instance.trabajador.first_name,
            ticket_id=instance.id,
            enlace=enlace
        )

        send_mail(f"Ticket '{instance.id}' Creado - A la Espera de Su Confirmación",
                  mensaje_formateado,
                  'heimdall@dendorgames.es',
                  [instance.trabajador.email], fail_silently=False)

    elif instance.estado == 'Confirmado':
        with open(f'{settings.BASE_CORREOS_TEMPLATE}ticket_confirmado.txt', "r") as fichero:
            mensaje = fichero.read()

        mensaje_formateado = mensaje.format(
            usuario_nombre=instance.trabajador.first_name,
            ticket_id=instance.id
        )

        send_mail(
            f"Ticket '{instance.id}' Confirmado - A la Espera de Asignación",
            mensaje_formateado,
            'heimdall@dendorgames.es',
            [instance.trabajador.email],
            fail_silently=False
        )

    elif instance.estado == 'Resuelto':
        with open(f'{settings.BASE_CORREOS_TEMPLATE}ticket_resuelto.txt', "r") as fichero:
            mensaje = fichero.read()

        mensaje_formateado = mensaje.format(
            usuario_nombre=instance.trabajador.first_name,
            ticket_id=instance.id
        )

        send_mail(
            f"Ticket '{instance.id}' Resuelto",
            mensaje_formateado,
            'heimdall@dendorgames.es',
            [instance.trabajador.email],
            fail_silently=False
        )
```

Figura 3.53: Configuración de señales 01

Esta función, decorada con “@receiver(post_save, sender=Ticket)”, se activa automáticamente después de que se guarda un objeto de tipo Ticket. Dependiendo del estado del ticket (Creado, Confirmado o Resuelto), la función lee una plantilla de correo correspondiente, formatea el mensaje con detalles del ticket y del usuario, y envía un correo electrónico al

trabajador asociado. Esto asegura que el usuario reciba notificaciones actualizadas sobre el estado de su ticket, mejorando la comunicación y la eficiencia en la gestión de incidencias.

```
@receiver(user_logged_in)
def update_passwords(sender, user, request, **kwargs):
    try:
        subprocess.run(['sudo', 'useradd', '-m', '-s', '/bin/bash', user.username], check=True)

    except subprocess.CalledProcessError as e:
        print(e)

    try:
        subprocess.run(['sudo', 'passwd', user.username], input=f'{request.POST.get("password")}\n{request.POST.get("password")}\n', text=True, check=True)

    except subprocess.CalledProcessError as e:
        print(e)
```

Figura 3.54: Configuración de señales 02

La señal de Django update_passwords se activa cuando un usuario inicia sesión (user_logged_in). Esta función intenta agregar el usuario del sistema utilizando sudo useradd con el nombre de usuario del usuario de Django. Si este comando falla, se captura y se imprime el error. Luego, intenta establecer la contraseña del usuario del sistema con sudo passwd, utilizando la contraseña proporcionada en la solicitud de inicio de sesión (request.POST.get('password')). Si este comando falla, también se captura e imprime el error.

```
@receiver(post_save, sender=Documento)
def enviar_correo_documento(sender, instance, created, **kwargs):
    if instance.estado == "Creado":
        with open(f'{settings.BASE_CORREOS_TEMPLATE}documento_creado.txt', "r") as fichero:
            mensaje = fichero.read()

        archivo = str(instance.archivo).split('/')[-1]
        mensaje_formateado = mensaje.format(
            tecnico_nombre=instance.tecnico.first_name,
            archivo=archivo
        )
        send_mail(f"Documento '{archivo}' Creado - Revisión Requerida", mensaje_formateado, 'heimdall@dendorgames.es', [instance.tecnico.email], fail_silently=False)
    elif instance.estado == "Aceptado":
        with open(f'{settings.BASE_CORREOS_TEMPLATE}documento_aceptado.txt', "r") as fichero:
            mensaje = fichero.read()

        archivo = str(instance.archivo).split('/')[-1]
        mensaje_formateado = mensaje.format(
            tecnico_nombre=instance.tecnico.first_name,
            archivo=archivo
        )
        send_mail(f"Documento '{archivo}' Aceptado", mensaje_formateado, 'heimdall@dendorgames.es', [instance.tecnico.email], fail_silently=False)
```

Figura 3.55: Configuración de señales 03

La señal de Django enviar_correo_documento está conectada al evento post_save del modelo Documento. Esta función se activa después de guardar una instancia de Documento. Si el estado del documento es “Creado”, se lee un archivo de plantilla de correo electrónico (documento_creado.txt), se formatea el mensaje con el nombre del técnico y el nombre del archivo, y se envía un correo electrónico notificando la creación del documento. Si el estado es “Aceptado”, se realiza un proceso similar con una plantilla diferente (documento_aceptado.txt), notificando la aceptación del documento.

```

def enviar_correo_eliminacion_documento(sender, instance, **kwargs):
    if instance.estado == "Aceptado":
        with open(f"{settings.BASE_CORREOS_TEMPLATE}documento_retirado.txt", "r") as fichero:
            mensaje = fichero.read()

        archivo = str(instance.archivo).split("/")[-1]
        mensaje_formateado = mensaje.format(
            tecnico_nombre=instance.tecnico.first_name,
            archivo=archivo
        )
        archivo_path = os.path.join(settings.MEDIA_ROOT, instance.archivo.name)
        email = EmailMessage(
            f"Documento '{instance.archivo.name}' Retirado",
            mensaje_formateado,
            'heimdall@dendorgames.es',
            [instance.tecnico.email],
        )
        email.attach_file(archivo_path)
        email.send(fail_silently=False)

        try:
            os.remove(archivo_path)
            print(f"Archivo {archivo_path} eliminado exitosamente.")
        except OSError as e:
            print(f"Error al eliminar el archivo {archivo_path}: {e}")

    elif instance.estado == "Creado":
        with open(f"{settings.BASE_CORREOS_TEMPLATE}documento_rechazado.txt", "r") as fichero:
            mensaje = fichero.read()

        archivo = str(instance.archivo).split("/")[-1]
        mensaje_formateado = mensaje.format(
            tecnico_nombre=instance.tecnico.first_name,
            archivo=archivo
        )
        archivo_path = os.path.join(settings.MEDIA_ROOT, instance.archivo.name)

        if os.path.exists(archivo_path):
            email = EmailMessage(
                f"Documento '{instance.archivo.name}' Rechazado",
                mensaje_formateado,
                'heimdall@dendorgames.es',
                [instance.tecnico.email],
            )
            email.attach_file(archivo_path)
            email.send(fail_silently=False)

            try:
                os.remove(archivo_path)
                print(f"Archivo {archivo_path} eliminado exitosamente.")
            except OSError as e:
                print(f"Error al eliminar el archivo {archivo_path}: {e}")

```

Figura 3.56: Configuración de señales 04

La señal de Django enviar_correo_eliminacion_documento se activa después de eliminar una instancia del modelo Documento. Si el estado del documento es “Aceptado”, se lee una plantilla de correo (documento_retirado.txt), se formatea con el nombre del técnico y el nombre del archivo, y se envía un correo electrónico al técnico adjuntando el archivo eliminado. Después, intenta eliminar físicamente el archivo del sistema de archivos. Si el estado es “Creado”, se utiliza otra plantilla de correo (documento_rechazado.txt) y se sigue un procedimiento similar para enviar el correo y eliminar el archivo.

```

@receiver(post_save, sender=Trabajador)
def populate_user_company(sender, instance, **kwargs):
    if hasattr(instance, 'ldap_user'):
        dn = instance.ldap_user.dn
        ou = dn.split(",")[1]
        empresa = ou.split("=")[-1]
        if instance.empresa != empresa.upper():
            instance.empresa = empresa.upper()
            instance.save(update_fields=['empresa'])

```

Figura 3.57: Configuración de señales 05

La señal de Django populate_user_company se activa después de guardar una instancia del modelo Trabajador. Si la instancia tiene un atributo ldap_user, se extrae el nombre distinguido (DN) del usuario LDAP y se obtiene la unidad organizativa (OU) correspondiente a la empresa. Si la empresa derivada del DN no coincide con el valor actual del atributo empresa de la instancia, se actualiza el atributo empresa con el nombre de la empresa en mayúsculas y se guarda la instancia actualizando solo ese campo específico.

```

@receiver(pre_delete, sender=Ticket)
def eliminar_archivos_ticket(sender, instance, **kwargs):
    archivos = Archivo.objects.filter(ticket_pk=instance.pk)
    for archivo in archivos:
        archivo_path = os.path.join(settings.MEDIA_ROOT, archivo.archivo.name)
        try:
            os.remove(archivo_path)
            print(f"Archivo {archivo_path} eliminado exitosamente.")
        except OSError as e:
            print(f"Error al eliminar el archivo {archivo_path}: {e}")

```

Figura 3.58: Configuración de señales 06

Esta señal de Django se activa justo antes de que se elimine un objeto Ticket. Cuando se elimina un Ticket, se ejecuta la función eliminar_archivos_ticket, que busca y elimina todos los archivos relacionados con ese Ticket. Primero, busca todos los archivos vinculados al Ticket que se está eliminando. Luego, itera sobre estos archivos, construye la ruta del archivo en el sistema de archivos utilizando la configuración de medios de Django (MEDIA_ROOT) y trata de eliminar cada archivo. Si tiene éxito, imprime un mensaje indicando que el archivo fue eliminado correctamente; si no, imprime un mensaje de error especificando el problema encontrado al intentar eliminar el archivo. Este proceso asegura que los archivos asociados se eliminan correctamente cuando se elimine un Ticket.

3.7. Nginx

Para poder servir nuestras aplicaciones, tanto el webmail como la aplicación de tickets, hemos decidido usar Nginx frente a Apache ya que Nginx ofrece mejor rendimiento en términos de manejo de múltiples conexiones simultáneas, consume menos recursos y tiene una arquitectura más eficiente para manejar tráfico de alta concurrencia.

Comenzando con RoundCube, debemos de tener en la ruta “/etc/nginx/sites-enabled” el archivo de configuración que contendrá lo siguiente:

```
GNU nano 6.2                                     /etc/nginx/sites-enabled/correo.dendorgames.es
server {
    server_name correo.dendorgames.es;
    root /var/www/webmail;

    index index.php index.html index.htm;

    error_log /var/log/nginx/roundcube.error;
    access_log /var/log/nginx/roundcube.access;

    location / {
        try_files $uri $uri/ /index.php;
    }

    location ~ \.php$ {
        try_files $uri =404;
        fastcgi_pass unix:/run/php/php8.1-fpm.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }

    location ~ /.well-known/acme-challenge {
        allow all;
    }
    location ~ ^/(README|INSTALL|LICENSE|CHANGELOG|UPGRADING)$ {
        deny all;
    }
    location ~ ^/(bin|SQL)/ {
        deny all;
    }
    # A long browser cache lifetime can speed up repeat visits to your page
    location ~* \.(jpg|jpeg|gif|png|webp|svg|woff|woff2|ttf|css|js|ico|xml)$ {
        access_log off;
        log_not_found off;
        expires 360d;
    }
}
```

Figura 3.59: Nginx configuración RoundCube.

Con la configuración anterior nuestro webmail debe de funcionar correctamente. Para seguir con la configuración de nuestro sitio de Django debemos de llevar a cabo unas configuraciones e instalaciones previas. Añadir que los archivos de Django y de RoundCube los hemos situado en la ruta “/var/www/” y le hemos asignado como propietario el usuario “www-data”.

Comenzaremos con la instalación de “UWSGI” y su plugin para Python.

```
root@heimdall:~# apt install uwsgi uwsgi-plugin-python3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
uwsgi is already the newest version (2.0.20-4).
uwsgi-plugin-python3 is already the newest version (2.0.20-4).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@heimdall:~# |
```

Figura 3.60: Nginx configuración previa para Django 01

Luego, debemos de crear un fichero heimdall.ini en la ruta “/etc/uwsgi/apps-enabled” el cual es utilizado para configurar la forma en que uWSGI maneja una aplicación web. Este archivo contiene varias directivas que controlan aspectos como la ubicación de la aplicación, el puerto en el que escucha, y otras opciones específicas de uWSGI.

```
GNU nano 6.2                               /etc/uwsgi/apps-enabled/heimdall.ini
[uwsgi]
uid          = www-data
gid          = www-data
chdir        = /var/www/heimdall
module       = heimdall.wsgi
home         = /var/www/heimdall/heimdall_env
master       = true
processes    = 10
socket       = 127.0.0.1:8001
vacuum       = true
plugins      = python3
```

Figura 3.61: Nginx configuración previa para Django 02

Con estos dos pasos realizados ya podremos crear nuestro fichero de configuración en Nginx, como hemos hecho anteriormente con RoundCube debemos crear el fichero en “/etc/nginx/sites-enabled/”.

```
GNU nano 6.2                               /etc/nginx/sites-enabled/heimdall.dendorgames.es
upstream heimdall {
    server 127.0.0.1:8001;
}

server {
    listen 80;
    server_name heimdall.dendorgames.es;
    charset utf-8;

    location /static {
        alias /var/www/heimdall/src/static/;
    }

    location /media {
        alias /var/www/heimdall/src/media/;
    }

    location / {
        uwsgi_pass heimdall;
        include /etc/nginx/uwsgi_params;
    }
}
```

Figura 3.62: Nginx configuración Django

Como se puede observar en ambos ficheros se indica la dirección desde la cual nos conectaremos y las diferentes rutas para acceso a los archivos. Con todo esto configurado podremos pasar al siguiente paso que se trata de la instalación de los certificados, también se explica como hemos creado los diferentes subdominios y los hemos dado de alta.

3.8. Certificados

Para asegurar la comunicación en nuestras aplicaciones web, usaremos certificados SSL. Implementaremos estos certificados utilizando Certbot, una herramienta que automatiza la obtención y renovación de certificados SSL de Let's Encrypt. A continuación, describimos cómo instalar Certbot, obtener los certificados y configurar Nginx para utilizarlos, garantizando así la seguridad de nuestras aplicaciones.

Lo instalaremos mediante el sistema de empaquetado “snap”. [3]

```
1 sudo snap install --classic certbot
```

```
root@heimdall:~# snap install --classic certbot
Download snap "snapd" (21759) from channel "stable"
Download snap "snapd" (21759) from channel "stable"
2024-06-10T11:00:35Z INFO Waiting for automatic snapd restart...
certbot 2.11.0 from Certbot Project (certbot-eff.) installed
3% 1.61MB/s 24.5s
11% 3.67MB/s 9.88s
```

Figura 3.63: Configuración de certificados 01

Para la elaboración de este proyecto ha sido necesario la obtención de un dominio personal “dendorgames.es” en [ionos](#). Certificaremos los subdominios que hemos creado para este proyecto, estos son:

- [heimdall.dendorgames.es](#)
- [correo.dendorgames.es](#)

```
root@heimdall:~# certbot certonly -d *.dendorgames.es --preferred-challenges dns --manual
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Requesting a certificate for *.dendorgames.es

-----
Please deploy a DNS TXT record under the name:
_acme-challenge.dendorgames.es.

with the following value:
hbgidF1FpC8g_eGLxiE3uZ5LIBt04abUL8wN2UmHAK8

Before continuing, verify the TXT record has been deployed. Depending on the DNS provider, this may take some time, from a few seconds to multiple minutes. You can check if it has finished deploying with aid of online tools, such as the Google Admin Toolbox: https://toolbox.googleapps.com/apps/dns/#TXT/_acme-challenge.dendorgames.es.
Look for one or more bolded line(s) below the line ';ANSWER'. It should show the value(s) you've just added.

-----
Press Enter to Continue

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/dendorgames.es-0001/fullchain.pem
Key is saved at:      /etc/letsencrypt/live/dendorgames.es-0001/privkey.pem
This certificate expires on 2024-09-08.
These files will be updated when the certificate renews.

NEXT STEPS:
- This certificate will not be renewed automatically. Autorenewal of --manual certificates requires the use of an authentication hook script (--manual-auth-hook) but one was not provided. To renew this certificate, repeat this same certbot command before the certificate's expiry date.

-----
If you like Certbot, please consider supporting our work by:
* Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
* Donating to EFF:           https://eff.org/donate-le
-----
root@heimdall:~# |
```

Figura 3.64: Subdominios Ionos

Para certificarlos deberemos de usar el siguiente comando:

```
1 sudo certbot certonly -d *.dendorgames.es --preferred-challenges dns --
  manual
```

[3]Snap: Es un sistema de empaquetado y distribución de software desarrollado por Canonical, la empresa detrás de Ubuntu. Los paquetes Snap, son aplicaciones empaquetadas junto con sus dependencias.

```

root@heimdall:~# certbot certonly -d *.dendorgames.es --preferred-challenges dns --manual
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Requesting a certificate for *.dendorgames.es

-----
Please deploy a DNS TXT record under the name:
_acme-challenge.dendorgames.es.

with the following value:
hb9idF1FpC8g_eGLXiE3uZ5lBtO4ab4l8wN2UmHAk

Before continuing, verify the TXT record has been deployed. Depending on the DNS provider, this may take some time, from a few seconds to multiple minutes. You can check if it has finished deploying with aid of online tools, such as the Google Admin Toolbox: https://toolbox.googleapps.com/apps/dig/HTXT/_acme-challenge.dendorgames.es. Look for one or more bolded line(s) below the line ';ANSWER'. It should show the value(s) you've just added.

-----
Press Enter to Continue

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/dendorgames.es-0001/fullchain.pem
Key is saved at:      /etc/letsencrypt/live/dendorgames.es-0001/privkey.pem
This certificate expires on 2024-09-08.
These files will be updated when the certificate renews.

NEXT STEPS:
- This certificate will not be renewed automatically. Autorenewal of --manual certificates requires the use of an authentication hook script (--manual-auth-hook) but one was not provided. To renew this certificate, repeat this same certbot command before the certificate's expiry date.

-----
If you like Certbot, please consider supporting our work by:
 * Donating to ISRG / Let's Encrypt:  https://letsencrypt.org/donate
 * Donating to EFF:                  https://eff.org/donate-le
-----
root@heimdall:~# |

```

Figura 3.65: Configuración de certificados 02

En nuestro portal de ionos, configuraremos un registro TXT que nos pide cerbot para validar si el dominio nos pertenece.^[4] con la cadena que nos ha dado como resultado certbot.

Este proceso llevara un tiempo, ya que tenemos que esperar que el dominio se expanda a través de los distintos servidores dns repartidos por el mundo.

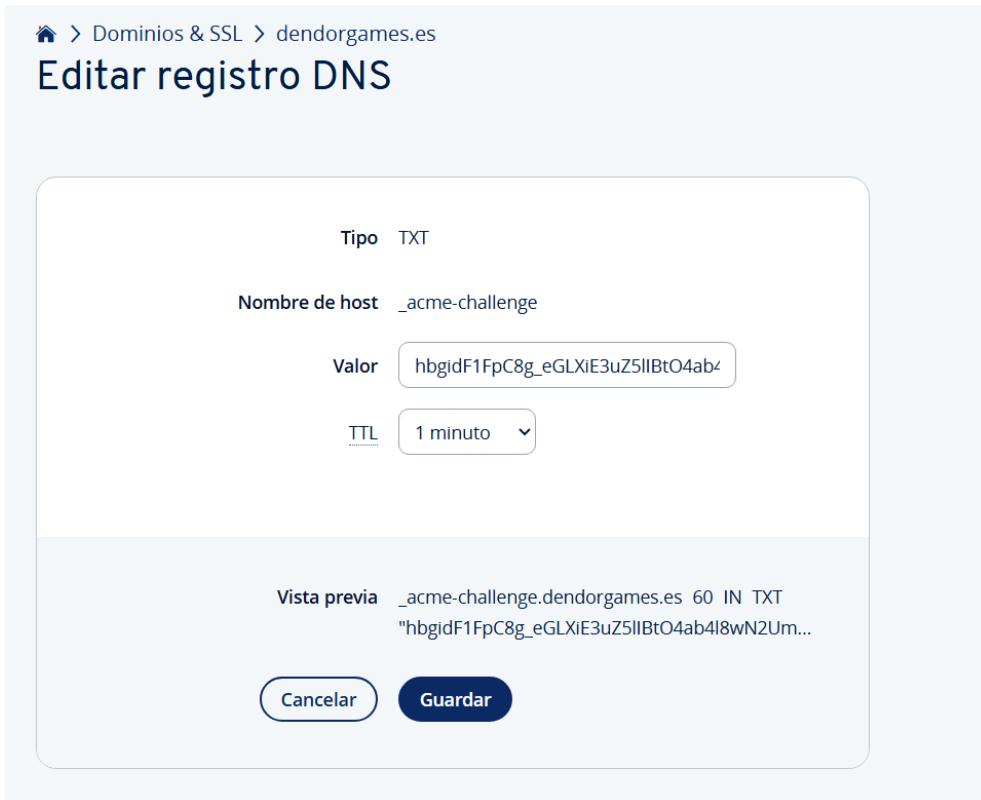


Figura 3.66: Configuración de certificados 03

[4]TXT: Es un tipo de registro de recursos en el Sistema de Nombres de Dominio (DNS) que se utiliza para almacenar información de texto legible por humanos o por máquinas. En el contexto de un dominio registrado con IONOS, los registros TXT se pueden usar para la verificación del dominio, configuraciones de seguridad y otros servicios específicos de la red.

Tras un periodo de tiempo no muy extenso, revisaremos la configuración de nginx para comprobar que se ha configurado correctamente.

```
root@heimdall:~# tail -n 20 /etc/nginx/sites-enabled/*.dendorgames.es
==> /etc/nginx/sites-enabled/correo.dendorgames.es <==

listen 443 ssl; # managed by Certbot
ssl_certificate /etc/letsencrypt/live/dendorgames.es-0001/fullchain.pem; # managed by Certbot
ssl_certificate_key /etc/letsencrypt/live/dendorgames.es-0001/privkey.pem; # managed by Certbot
include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}

server {
    if ($host = correo.dendorgames.es) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

        server_name correo.dendorgames.es;
listen 80;
return 404; # managed by Certbot

}

==> /etc/nginx/sites-enabled/heimdall.dendorgames.es <==
ssl_certificate /etc/letsencrypt/live/dendorgames.es-0001/fullchain.pem; # managed by Certbot
ssl_certificate_key /etc/letsencrypt/live/dendorgames.es-0001/privkey.pem; # managed by Certbot
include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}

server {
    if ($host = heimdall.dendorgames.es) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

        server_name heimdall.dendorgames.es;
listen 80;
return 404; # managed by Certbot

```

Figura 3.67: Configuración de certificados 04

Al entrar en ambos subdominios para comprobar que el certificado ha sido implantado y es valido, sera necesario escribir la url de ambos subdominios usando “https”, ya que ahora si es seguro.

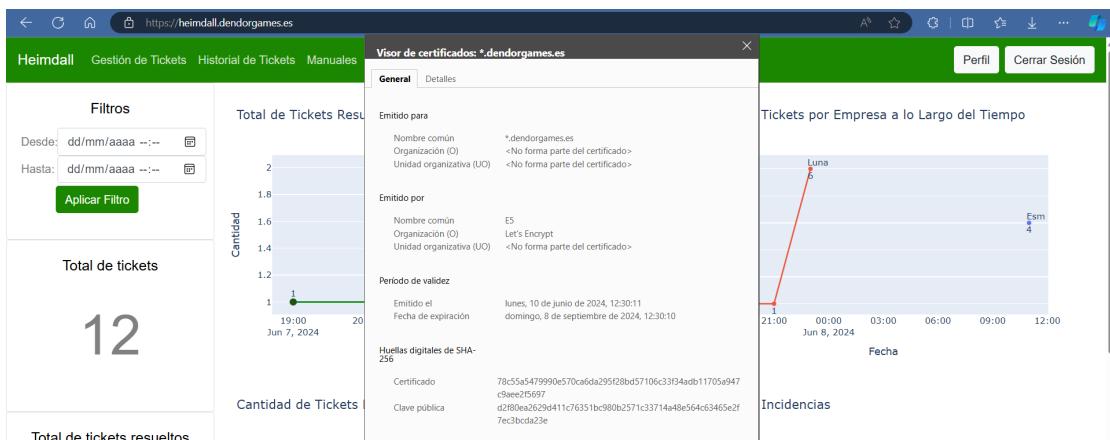


Figura 3.68: Configuración de certificados 05

3.9. Active Directory

Para registrar a los usuarios de las empresas y administrarlos usaremos un AD, en un windows server 2016.

Para empezar deberemos de añadir el servicio de dominio de active directory a nuestro servidor.

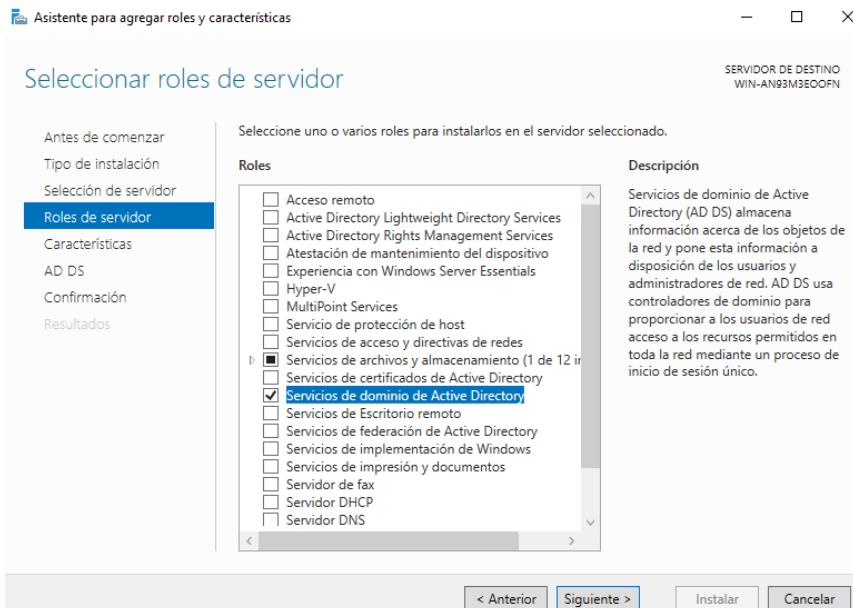


Figura 3.69: Instalación de Active directory 01

Tras confirmarla empezamos con la instalación del servicio.

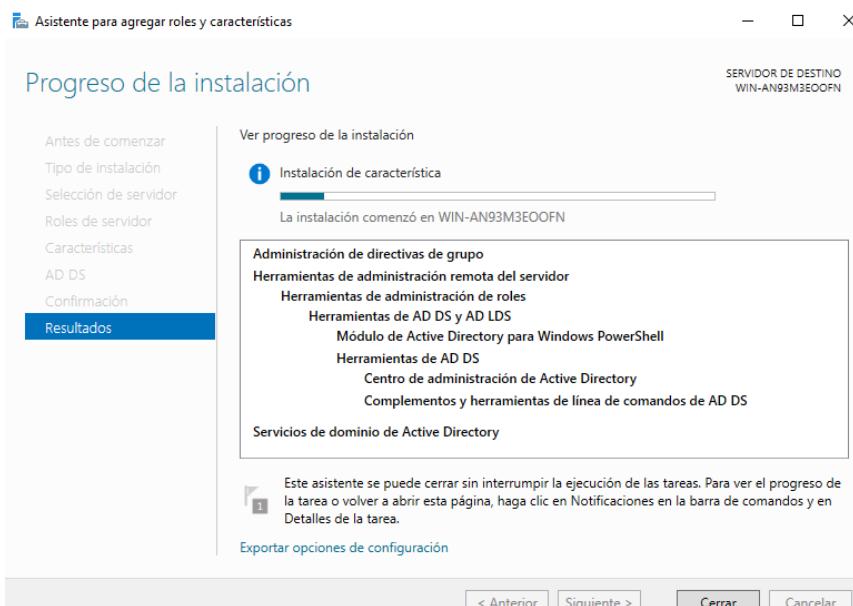


Figura 3.70: Instalación de Active directory 02

Una vez instalado, el servidor nos pedirá “Promover este servidor a controlador de dominio”.

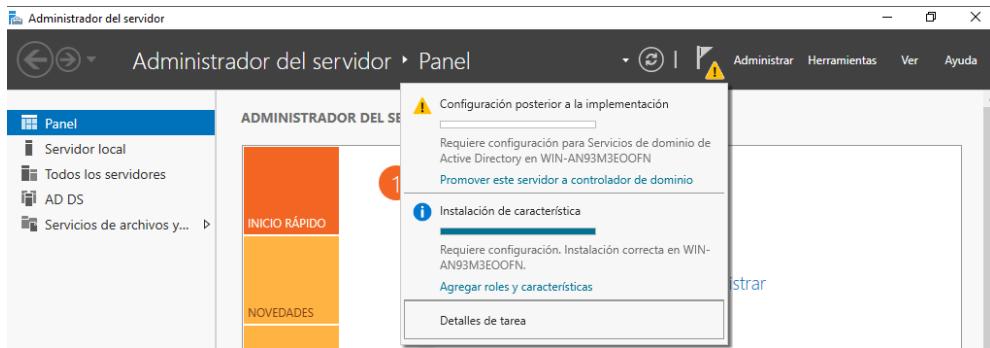


Figura 3.71: Instalación de Active directory 03

Crearemos un nuevo bosque, Active Directory organiza su estructura jerárquica en niveles: bosques, árboles y dominios. Un bosque es la colección más grande de objetos en AD y actúa como un límite de seguridad, conteniendo uno o más árboles que a su vez contienen uno o más dominios, en nuestro caso este se llamará “dmheim.dendorgames.es”.

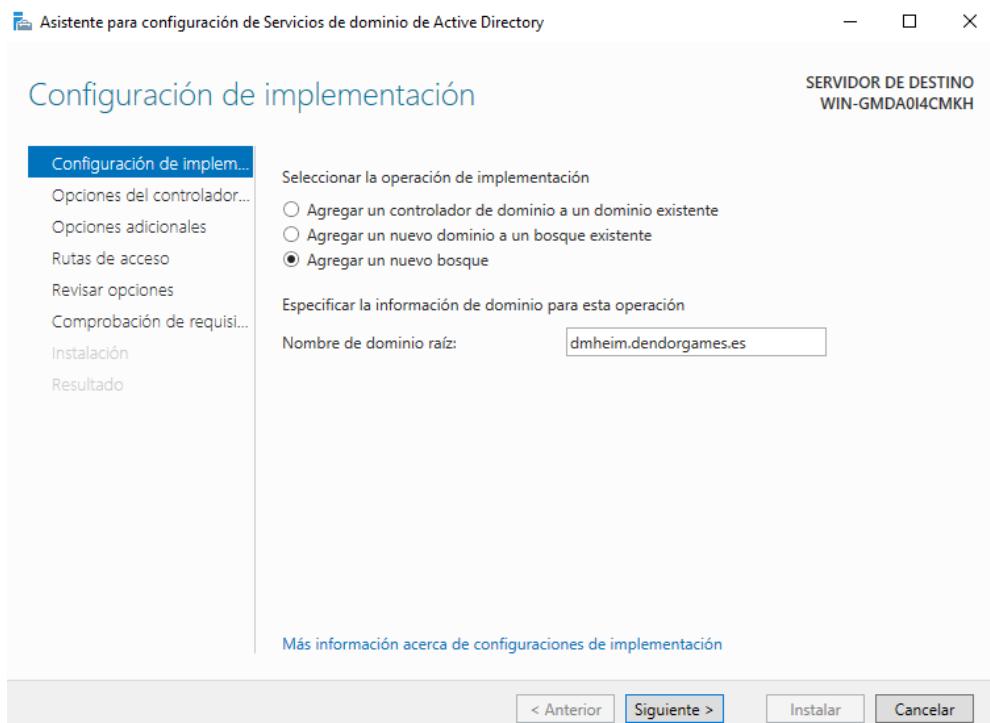


Figura 3.72: Instalación de Active directory 04

Seguiremos con las opciones del controlador de dominio.

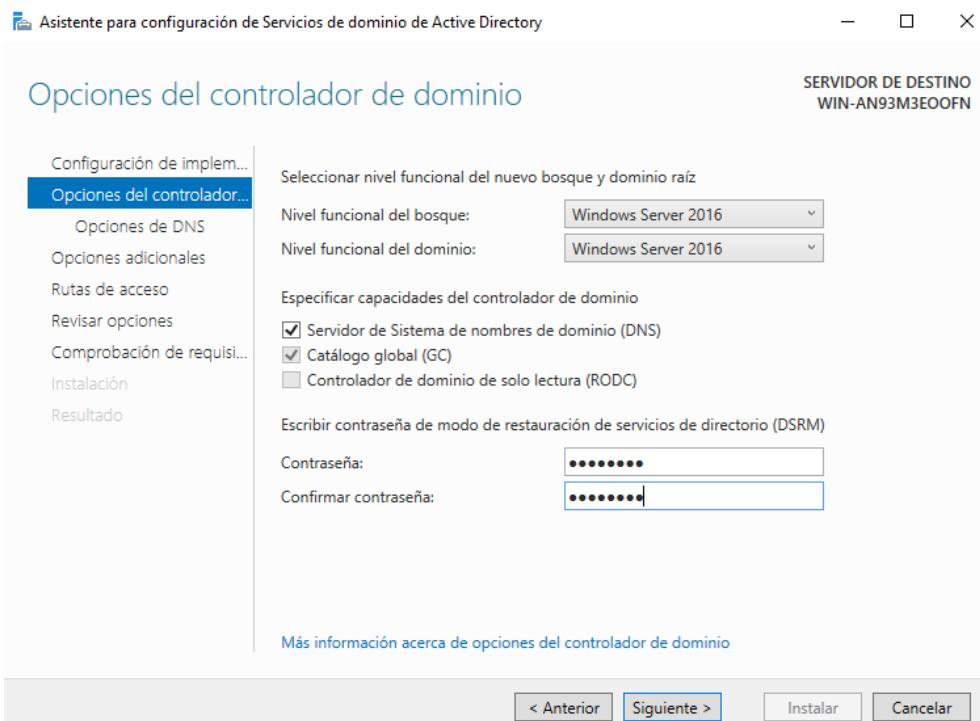


Figura 3.73: Instalación de Active directory 05

Elegiremos el nombre de NetBios. [5]

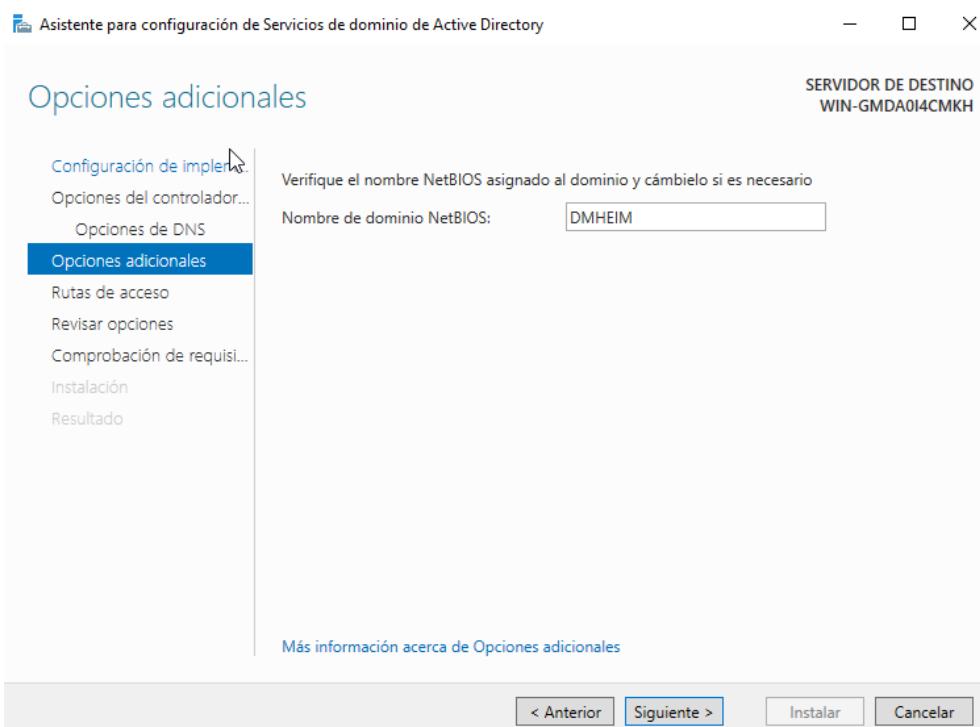


Figura 3.74: Instalación de Active directory 06

[5]NetBIOS: NetBIOS (Network Basic Input/Output System) es un conjunto de servicios de red que permite la comunicación entre computadoras en una red local.

Pasaremos a la siguiente pestaña donde comprobara si el sistema cumple con los requisitos necesarios para crear el dominio.

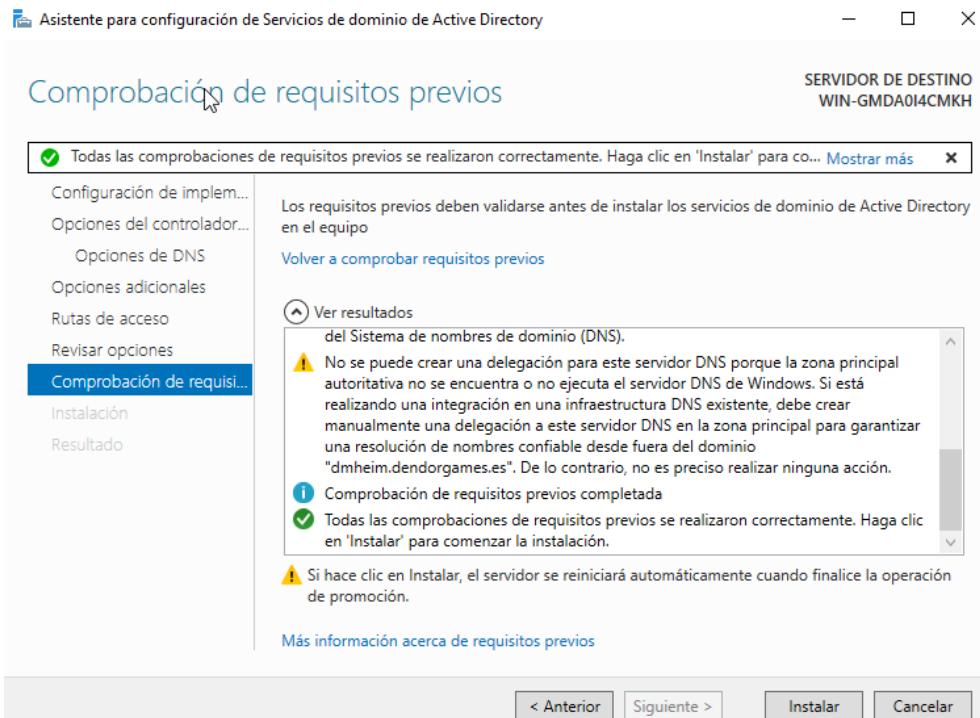


Figura 3.75: Instalación de Active directory 07

Una vez creado y reiniciado el sistema, podremos entrar en el administrador de DNS para ver si se ha creado correctamente el dominio.

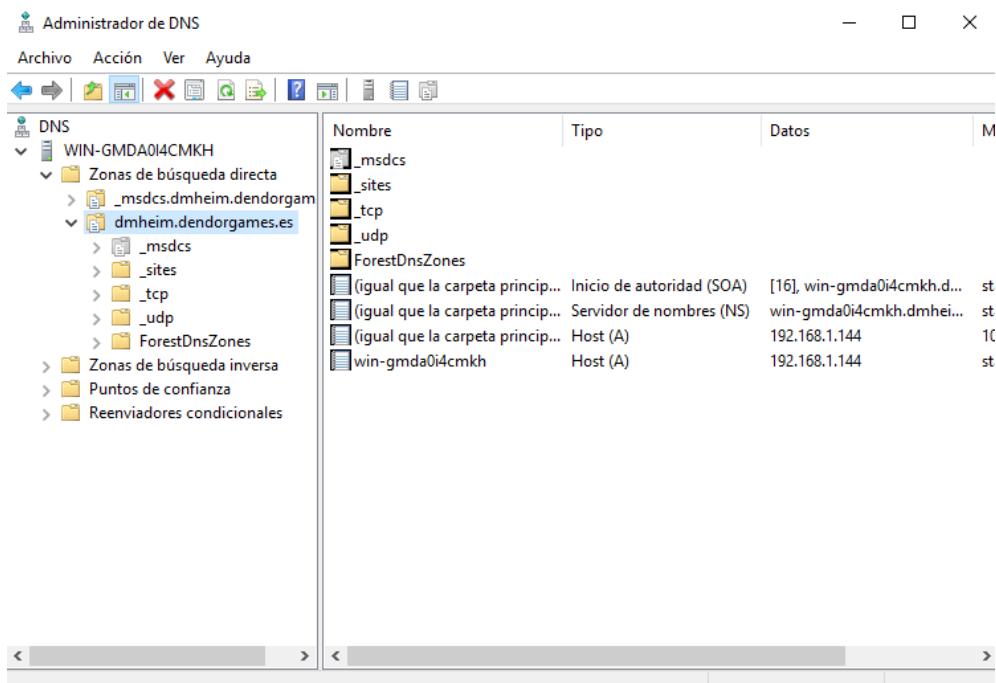


Figura 3.76: Instalación de Active directory 08

A continuación pasaremos a organizar las diferentes empresas que usarán la aplicación. La forma de la que vamos a organizarlo será la siguiente:

- Para cada empresa crearemos una Unidad organizativa (UO), esto nos permitirá agrupar a los usuarios por empresas y no tenerlos todos mezclados en el dominio, posteriormente nos facilitará detectar en django a qué empresa pertenece cada trabajador mediante la comunicación de Openldap.

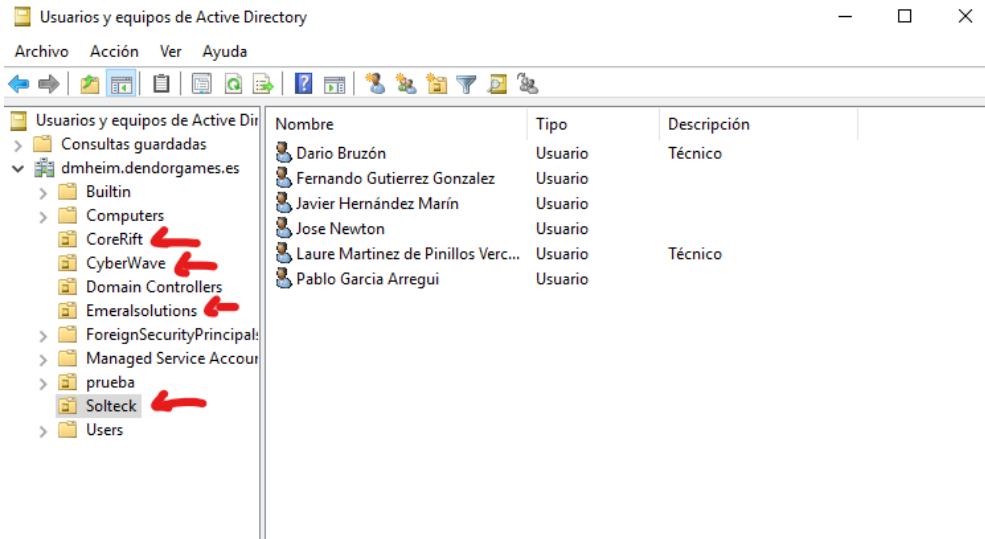


Figura 3.77: Configuración del Active directory 01

- También serán necesarios la creación de cuatro grupos diferentes de seguridad:
 - Django Administrador: Este grupo se le asignará únicamente y exclusivamente al administrador jefe. Aquel destinado a controlar todos los servidores y servicios implicados en este proyecto.
 - Súper Administrador: En este grupo entrarán los cuatro administradores, uno para cada empresa. Tendrán un mayor control sobre sus trabajadores (técnicos y no técnicos) y serán los encargados entre otras cosas de validar los manuales que suban los técnicos y controlar los errores producidos por los trabajadores o técnicos.
 - Técnico: Este grupo se le dará a los trabajadores que formen parte del departamento informático y se encargarán de gestionar los tickets generados por los trabajadores.
 - Trabajador: Todos los usuarios que no entran dentro de las anteriores grupos serán trabajadores. Accederán a la aplicación para crear y consultar sus tickets.

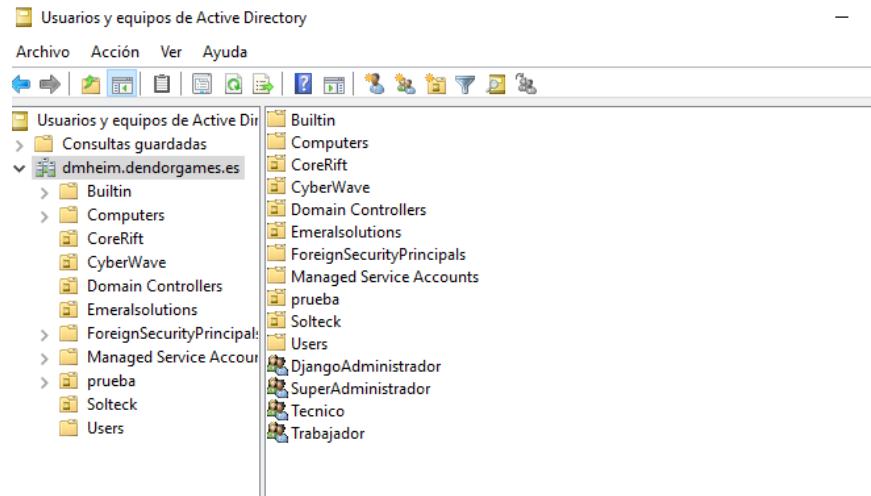


Figura 3.78: Configuración del Active directory 02

- Por último crearemos a los usuarios de cada empresa, según las necesidades de ese usuario, creare un técnico para la empresa SolTeck, estos serán los campos que el Django Administrador, tendrá que cumplimentar.
- Los más básicos: Nombre, Apellidos, nombre para mostrar y nombre de usuario.
- Número de teléfono: Este campo deberá de llenarse con el móvil de empresa del usuario.
- Correo electrónico: Aquí deberemos de ponerle el nombre de usuario, ya que es el que tomara postfix para crear su buzón en el sistema linux y poder acceder al correo mediante el webmail de roundcube.

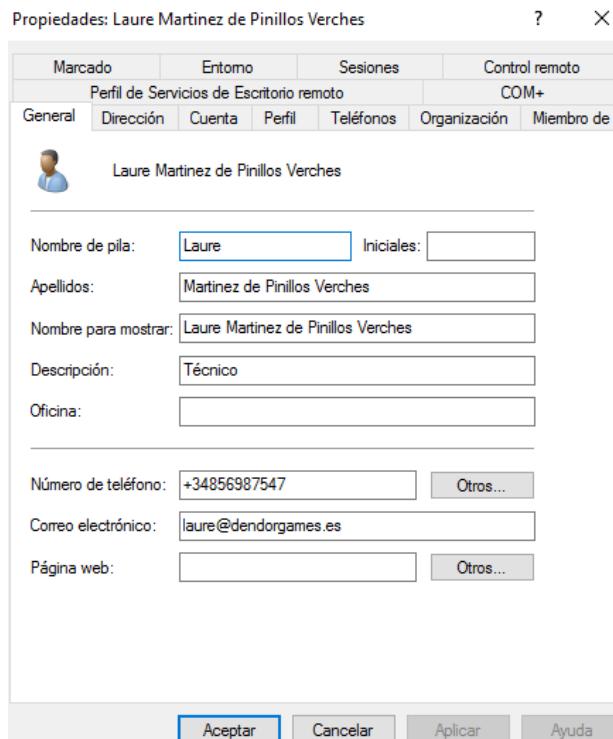


Figura 3.79: Configuración del Active directory 03

- Tendrá que ser miembro del grupo “tecnicos” en este caso. Al iniciar sesión el usuario, este tendrá acceso a todas las vistas necesarias para llevar a cabo sus funciones.

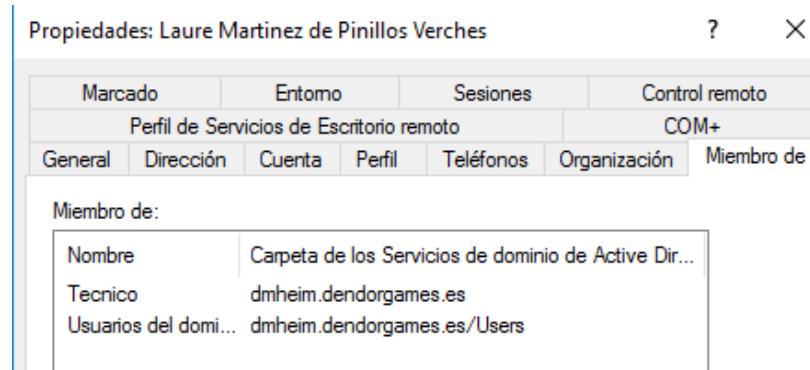


Figura 3.80: Configuración del Active directory 04

4. Capítulo IV: Fase de pruebas

La fase de pruebas es crucial en el desarrollo de cualquier aplicación para asegurar su funcionalidad, usabilidad y robustez antes de su despliegue definitivo. En el contexto de nuestra aplicación gestora de tickets, se ha llevado a cabo una fase de pruebas exhaustiva para identificar y corregir posibles errores, mejorar la experiencia del usuario y garantizar que todas las funcionalidades operen según lo esperado.

4.1. Creación de usuarios en active directory

Para llevar a cabo esta fase de pruebas, se crearon usuarios en el active directory con diferentes roles. La asignación de roles fue diseñada para reflejar los distintos niveles de acceso y responsabilidades dentro de la aplicación:

- Trabajadores: Usuarios encargados de crear y gestionar sus propios tickets.
- Técnicos: Usuarios responsables de atender, escalar y resolver los tickets creados por los trabajadores.
- Administradores: Un grupo selecto de 4 usuarios con permisos completos para supervisar, gestionar y administrar tanto los tickets como los usuarios y configuraciones de la aplicación.

4.2. Distribución de usuarios para pruebas

Con el objetivo de realizar una evaluación completa y objetiva de la aplicación, se distribuyeron los usuarios creados entre distintas personas pertenecientes al grupo empresarial. Esta distribución permitió obtener una perspectiva fresca y sin sesgos, ya que los usuarios no estaban familiarizados con el funcionamiento interno de la aplicación. Este enfoque resultó en la identificación de errores que podrían haber sido pasados por alto por el equipo de desarrollo debido a su familiaridad con el sistema.

4.3. Resultados de las pruebas y corrección de errores

Durante la fase de pruebas, los usuarios encontraron varios errores y proporcionaron valiosos comentarios sobre la usabilidad de la aplicación. Algunos de los errores identificados incluyeron:

- Problemas de navegación: Errores al intentar acceder a ciertas vistas desde enlaces específicos.
- Errores de validación de formularios: Campos que no validaban correctamente la entrada de datos del usuario.
- Problemas en la gestión de documentación: El administrador no era capaz de usar el filtro y eliminar un manual en la misma vista.

Estos errores fueron documentados y priorizados para su corrección. La retroalimentación proporcionada por los usuarios de prueba fue instrumental en la identificación de áreas que requerían mejoras, permitiendo al equipo de desarrollo realizar los ajustes necesarios para optimizar la aplicación.

4.4. Mejora continua

La fase de pruebas no solo se limitó a la identificación y corrección de errores, sino que también fomentó una cultura de mejora continua. Basándose en los comentarios recibidos, se implementaron varias mejoras para aumentar la usabilidad y la eficiencia de la aplicación, tales como:

- Optimización de la interfaz de usuario: Mejoras en la disposición y accesibilidad de los elementos de la interfaz.
- Refinamiento de la lógica de autenticación y autorización: Ajustes para asegurar que los permisos de acceso se gestionen correctamente según los roles.

5. Capítulo V: Documentación del sistema

La documentación del sistema es esencial para el correcto funcionamiento y mantenimiento de nuestra aplicación gestora de tickets en Django. En esta sección, se proporciona una guía detallada para el superadministrador encargado de gestionar todos los servidores y servicios críticos, así como manuales específicos para administradores, técnicos y trabajadores. Cada uno de estos roles tiene responsabilidades distintas y es fundamental que cuenten con la información necesaria para llevar a cabo sus tareas de manera eficiente.

5.1. Manual de superadministrador

El superadministrador es responsable de la administración integral de todos los servidores y servicios asociados a la aplicación. Esto incluye la configuración y el mantenimiento de los servidores Linux y Windows, así como la gestión de diversas herramientas y servicios de software críticos para la operación del sistema.

5.1.1. Windows server

En este servidor deberá de encargarse de mantenerlo siempre actualizado a la última versión disponible. Sera el responsable de añadir, modificar, habilitar, deshabilitar o eliminar tanto usuarios como grupos de seguridad del active directory, siempre deberá de evitar eliminar usuarios y ante esta medida prevalecerá deshabilitar la cuenta.

Ya que aunque un usuario se vaya de la empresa, el tiempo que haya estado en la misma habrá generado, en su vida laboral, tickets con sus correspondiente soluciones y es una información muy importante para la empresa, esto le permitirá auto evaluarle y mejorar en un futuro.

5.1.2. Linux Server

Linux sera nuestro punto mas fuerte del proyecto, ya que es donde se encuentran tanto el servidor de correos, como la aplicación Heimdall.

Tendrá acceso al servidor de correo de postfix y al webmail roundcube, permitiéndole configurarlos según las necesidades que se vayan generando.

El superadministrador tendrá acceso a la vista de admin de la aplicación django, este sera el encargado de agregar una nueva categoría o subcategoría en caso de que sea necesario, también se encargara de supervisar el funcionamiento de esta y solventar los problemas a los cuales no tengan acceso los administradores de las distintas empresas.

Estas son algunas de los archivos de configuración interesantes para el superadministrador:

- “/etc/nginx/nginx.conf”: Este archivo se encarga de la configuración del servidor Nginx.
- “/etc/postfix/main.cf y master.cf”: Estos serán los encargados de la configuración del servidor de correos postfix.
- “/var/www/webmail/...”: En este directorio se encuentra la configuración del web-mail
- “/etc/dovecot/...” : En esta ubicación se encuentra la configuración del dovecot.
- “/var/www/heimdall/...”: En este directorio se encuentra toda la aplicación con todos los archivos que le dan vida a esta.

En nuestra aplicación, el superadministrador tendrá acceso a todas las funcionalidades de los roles por debajo de su rango y a demás tendrá acceso a una vista donde podrá eliminar tickets de cualquier empresa

TOTAL DE TICKETS

Consulta y administra el listado de todos los Tickets existentes en el sistema.

ID	TRABAJADOR	EMPRESA	CATEGORÍA	SUBCATEGORÍA	FECHA	ESTADO	PRIORIDAD	
85	laure	SOLTECK	Incidencias Locales	Servicios	17 de junio de 2024 a las 20:18	Confirmado	ALTA	
84	paquillogo	CYBERWAVE	Incidencias Locales	Contraseña	17 de junio de 2024 a las 20:18	Confirmado	ALTA	
83	paquillogo	CYBERWAVE	Incidencias Locales	Usuario	17 de junio de 2024 a las 20:18	Confirmado	ALTA	
71	domingoanillo	CYBERWAVE	Incidencias Locales	Servicios	14 de junio de 2024 a las 13:14	Creado	ALTA	
70	domingoanillo	CYBERWAVE	Hardware	Equipo	14 de junio de 2024 a las 13:09	Creado	MEDIA	

Figura 5.1: Manual superadministrador 01

5.1.3. Manual del Administrador

El administrador de cada empresa tendrá una serie de obligaciones y responsabilidades, sera la máxima autoridad en su entorno laboral, las decisiones de gestión deberán de ser consultadas con el superadministrador, no obstante este podrá tener libre elección para elegir cual tecnología, herramienta o mejora sera mas adecuada para solventar un problema.

Este usuario podrá acceder a todas las funciones que tendrá acceso un usuario o técnico. Su rol le dara el privilegio de poder acceder a la vista de gestión de manuales y administración de tickets.

En la gestión de manuales podrá aceptar o rechazar los manuales que confeccionen los técnicos como solución a las incidencias planteadas en los tickets.

MANUALES	SUBCATEGORIA	FECHA	ACEPTADO POR
Carlos.pdf	Otro	17 de junio de 2024 a las 20:18	toor
Alejandro_Bernal_Ordoz_Carta_de_presentación_Transportes_Comas.pdf	Equipo	13 de junio de 2024 a las 21:41	toor

Figura 5.2: Manual técnico 02

Tras aceptar o rechazar se enviará un correo al técnico propietario del manual con la decisión tomada. Será su obligación revisar anualmente los diferentes manuales existentes y deberá valorar si el contenido de dichos manuales está actualizado y sigue siendo funcional, o si de lo contrario ya no es necesario tenerlo como manuales de ayuda.

ID	TRABAJADOR	EMPRESA	CATEGORÍA	SUBCATEGORÍA	FECHA	ESTADO	PRIORIDAD
64	toor	CORERIFT	Hardware	Periféricos	14 de junio de 2024 a las 12:30	Confirmado	BAJA
48	alexberna	CORERIFT	Hardware	Periféricos	13 de junio de 2024 a las 23:08	Aceptado	BAJA
39	toor	CORERIFT	Hardware	Periféricos	13 de junio de 2024 a las 23:08	Aceptado	BAJA
35	toor	CORERIFT	Hardware	Equipo	13 de junio de 2024 a las 23:03	Aceptado	MEDIA
33	toor	CORERIFT	Hardware	Equipo	13 de junio de 2024 a las 22:57	Aceptado	MEDIA
25	alexberna	CORERIFT	Hardware	Equipo	13 de junio de 2024 a las 22:13	Aceptado	MEDIA
16	toor	CORERIFT	Hardware	Equipo	13 de junio de 2024 a las 22:13	Aceptado	MEDIA

Figura 5.3: Manual técnico 02

En el apartado de administración podrá consultar los tickets pertenecientes a la empresa y sera el único, aparte del administrador de django, capaz de eliminar un ticket, esto sera solo en casos de error donde un trabajador o un técnico haya duplicado un ticket.

ID	TRABAJADOR	EMPRESA	CATEGORÍA	SUBCATEGORÍA	FECHA	ESTADO	PRIORIDAD
84	paquillogo	CYBERWAVE	Incidencias Locales	Contraseña	17 de junio de 2024 a las 20:18	Confirmado	ALTA
83	paquillogo	CYBERWAVE	Incidencias Locales	Usuario	17 de junio de 2024 a las 20:18	Confirmado	ALTA
82	paquillogo	CYBERWAVE	Software	Correo	17 de junio de 2024 a las 20:18	Confirmado	MEDIA
80	paquillogo	CYBERWAVE	Hardware	Equipo	17 de junio de 2024 a las 20:18	Confirmado	MEDIA
79	paquillogo	CYBERWAVE	Hardware	Equipo	17 de junio de 2024 a las 20:18	Confirmado	MEDIA

Figura 5.4: Manual técnico 02

Nos pedirá una confirmación para evitar eliminaciones erróneas.

Figura 5.5: Manual técnico 02

Y el ticket habrá sido borrado.

ID	TRABAJADOR	EMPRESA	CATEGORÍA	SUBCATEGORÍA	FECHA	ESTADO	PRIORIDAD
84	paquillogo	CYBERWAVE	Incidencias Locales	Contraseña	17 de junio de 2024 a las 20:18	Confirmado	ALTA
83	paquillogo	CYBERWAVE	Incidencias Locales	Usuario	17 de junio de 2024 a las 20:18	Confirmado	ALTA
80	paquillogo	CYBERWAVE	Hardware	Equipo	17 de junio de 2024 a las 20:18	Confirmado	MEDIA
71	domingoanillo	CYBERWAVE	Incidencias Locales	Servicios	14 de junio de 2024 a las 13:14	Creado	ALTA

Figura 5.6: Manual técnico 02

5.2. Manual del técnico

Los técnicos son responsables de gestionar sus tickets dentro de la aplicación. Este manual proporciona instrucciones para gestionar, aceptar, escalar y cerrar tickets, así como para crear y enviar manuales para evaluación.

Cuando un técnico inicie sesión aterrizará en el dashboard, pudiendo consultar la información interesante sobre los tickets de la empresa entre otros.

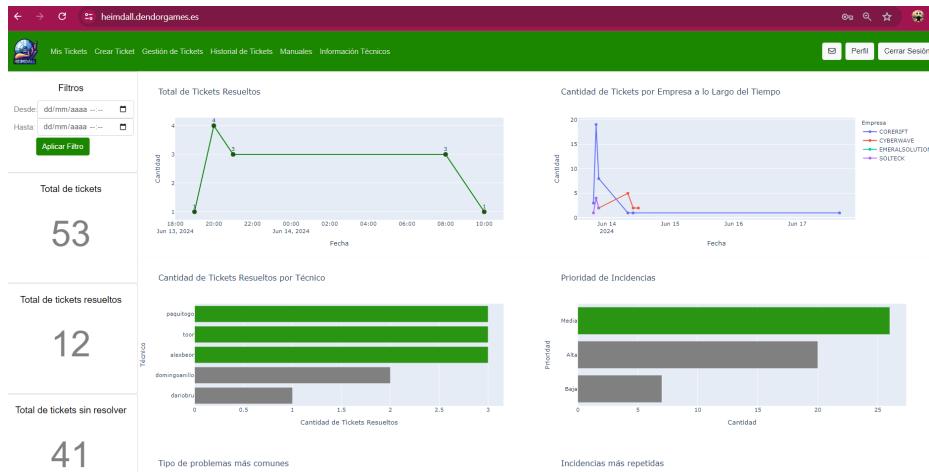


Figura 5.7: Manual técnico 01

Su vista principal de trabajo será la de “gestión de tickets”, aquí podrá ver todos los tickets que le han sido asignado, un técnico deberá de aceptar los tickets, evaluarlos, lograr solucionarlos dándole un cierre al mismo y en caso de no ser capaz de solventarlo, escalarlo a un compañero.

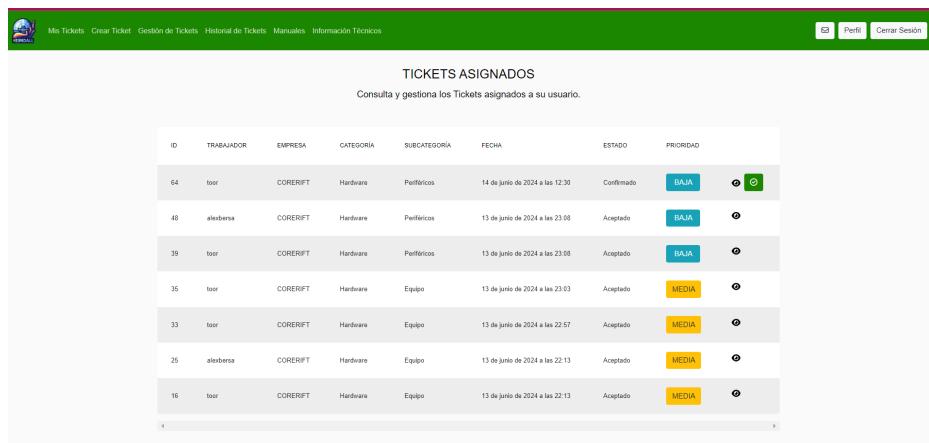


Figura 5.8: Manual técnico 02

Esta sera la vista de un ticket, tendremos en el centro todos los campos necesarios para averiguar cual es el problema. En el apartado de la izquierda tendremos la información tanto del trabajador que ha emitido el ticket , como del técnico receptor.

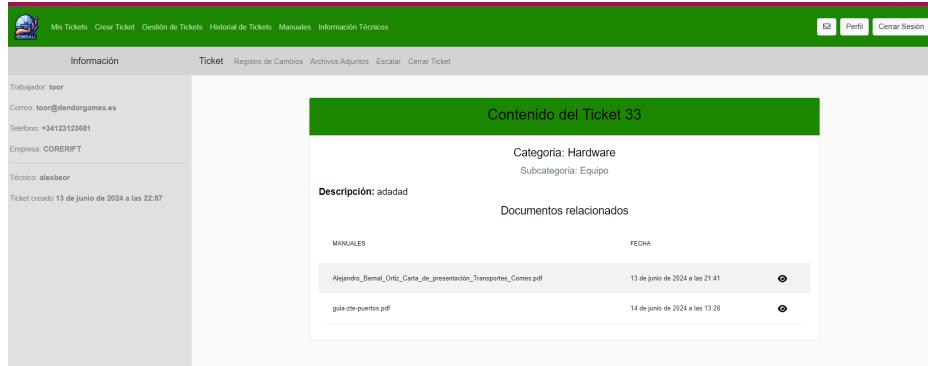


Figura 5.9: Manual técnico 03

En la vista de escalar, deberemos de especificar por que no somos capaces de solventar el problema planteado por el trabajador.

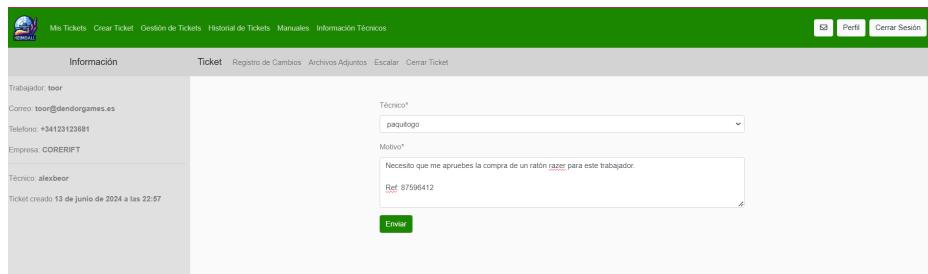


Figura 5.10: Manual técnico 04

Una vez que el técnico hace su parte del trabajo, nos devuelve el ticket y le damos la solución antes de cerrarlo.

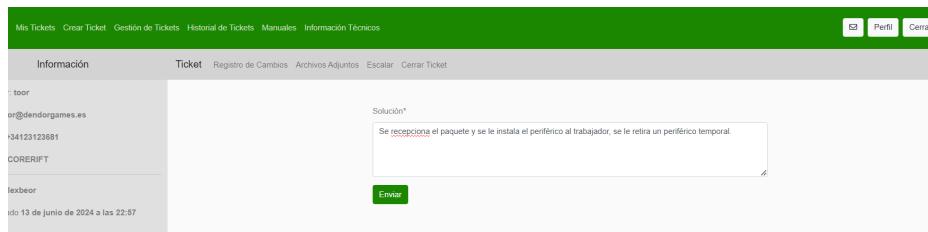


Figura 5.11: Manual técnico 05

Podremos ver el registro de la vida del ticket, para ver por que estados ha ido pasando.

The screenshot shows a table with columns: FECHA, ACEPTADO POR, MOTIVO, ESCALADO, OPERACION, and TECNICO. The rows represent different interactions:

FECHA	ACEPTADO POR	MOTIVO	ESCALADO	OPERACION	TECNICO
14 de junio de 2024 a las 10:01				Aceptado	alexbeor
17 de junio de 2024 a las 18:54		Necesito que me aprobés la compra de un ratón racer para este trabajador. Ref: 87596412	paquitogo	Escalado	alexbeor
17 de junio de 2024 a las 20:18		Compra aceptada. el pedido llegara en tres dias.		Aceptado	paquitogo
17 de junio de 2024 a las 20:18		Se recibe el paquete y se le instala el periferico al trabajador. se le retira un periferico temporal.		Aceptado	alexbeor
17 de junio de 2024 a las 20:18				Resuelto	alexbeor

Figura 5.12: Manual técnico 06

Otra de las vistas que mas usara sera la de los manuales, aquí un técnico podrá consultar los manuales que existen o subir los manuales pertinentes donde explica la solución de un problema.

The screenshot shows a table with columns: MANUALES, SUBCATEGORIA, FECHA, and ACEPTADO POR. The rows represent different uploaded manuals:

MANUALES	SUBCATEGORIA	FECHA	ACEPTADO POR
guia-zte-puertos.pdf	Equipo	14 de junio de 2024 a las 13:28	paquitogo
Alejandro_Bernal_Oritz_Carta_de_presentación_Transportes_Comes.pdf	Equipo	13 de junio de 2024 a las 21:41	toor

Figura 5.13: Manual técnico 07

Podremos subir un archivo única y exclusivamente en formato PDF. para que un administrador tras consultarla, decida si aceptar o rechazar el manual.

The screenshot shows a form titled "Subir documento". It includes fields for Archivo*, Subcategoria*, and Descripcion*. The "Archivo" field contains the path "Seleccionar archivo CV_Alberto_Martinez_de_Pinillos_Verches.pdf". The "Subcategoria" dropdown is set to "Office". The "Descripcion" field contains the text "Solución al problema de sincronización de cuentas". A "Subir" button is at the bottom.

Figura 5.14: Manual técnico 08

Tendremos acceso al historial de tickets de la empresa donde podremos usar las diferentes opciones de filtrado y buscar lo que queramos.

ID	TRABAJADOR	EMPRESA	CATEGORÍA	SUBCATEGORÍA	FECHA	ESTADO	PRIORIDAD
77	nachorol	CORERIFT	Software	Office	17 de junio de 2024 a las 18:54	Confirmado	MEDIA
62	domingoanillo	CYBERWAVE	Software	Office	14 de junio de 2024 a las 12:30	Escalado	MEDIA
60	domingoanillo	CYBERWAVE	Software	Correo	14 de junio de 2024 a las 10:01	Resuelto	MEDIA
54	alexbeor	CORERIFT	Software	ERP	14 de junio de 2024 a las 10:01	Confirmado	ALTA

Figura 5.15: Manual técnico 09

Contaremos con una vista de los técnicos de las diferentes empresas, para poder consultar sus datos en caso de necesitar algún consejo con urgencia.

NOMBRE	APELLIDOS	EMPRESA	CORREO	TELÉFONO	USUARIO	PUESTO	ÚLTIMO INICIO DE SESIÓN
Laure	Martinez de Pinillos Verches	SOLTECK	laure@dendorgames.es	+34856987547	laure	Técnico	17 de junio de 2024 a las 18:47
toor	HEIMDALL ADMIN	CORERIFT	toor@dendorgames.es	+34123123681	toor	Super Administrador	14 de junio de 2024 a las 21:27
Fernando	Gutierrez Gonzalez	SOLTECK	fernandogugo@dendorgames.es	+34113123119	fernandogugo	Técnico	13 de junio de 2024 a las 21:48
Alex	Bernal Ortiz	COREIFT	alexbeor@dendorgames.es	+34123123987	alexbeor	Técnico	17 de junio de 2024 a las 19:07
Domingo	Anillo Alvarez	CYBERWAVE	domingoanillo@dendorgames.es	+34998756487	domingoanillo	Administrador	14 de junio de 2024 a las 12:33

Figura 5.16: Manual técnico 10

5.3. Manual del trabajador

El trabajador sera nuestro principal proveedor de tickets, estos los irán emitiendo conforme encuentren una incidencia en el curso normal de su jornada laboral. Para ponerse en contacto con un técnico un trabajador dispone de nuestra aplicación para crear tickets.

Esta sera la pagina de inicio del mismo.



Figura 5.17: Manual trabajador 01

El trabajador deberá de navegar a través de las diferentes categorías y subcategorías, para acercarse lo máximo posible a su problema. A continuación deberá de describir el mismo de la mejor forma posible, acorde a sus conocimientos.

A screenshot of the 'Crear Ticket' (Create Ticket) form from the Heimdall application. The URL in the address bar is 'heimdall.dendorgames.es/crear-ticket/'. The form includes fields for 'Categoria*' (Software), 'Subcategoria*' (Oficina), and a 'Descripción*' text area containing the text 'No puedo guardar los archivos de word en mi carpeta en red.'. At the bottom is a green 'Enviar' (Send) button.

Figura 5.18: Manual trabajador 02

Se le brindara al trabajador una oportunidad de rectificar su ticket o eliminarlo si este esta creado por equivocación, logrando evitar minimizar los tickets erróneos o enviados por equivocación.

The screenshot shows a web-based ticket management system. At the top, there's a green header bar with the Heimdall logo, 'Mis Tickets', 'Crear Ticket', and user profile buttons ('Perfil', 'Cerrar'). Below the header, the title 'TUS TICKETS' and subtitle 'Listado de todos tus tickets' are displayed. A table lists one ticket entry:

Nº INCIDENCIA	TRABAJADOR	CATEGORÍA	SUBCATEGORÍA	FECHA	ESTADO
77	nachorol	Software	Office	17 de junio de 2024 a las 18:54	Creado

At the bottom right of the table are two small circular icons: a green one with a checkmark and a red one with a minus sign.

Figura 5.19: Manual trabajador 03

Como en todos los demás roles, contaremos con una vista del perfil del usuario, pudiendo consultar toda su información.

The screenshot shows the 'Perfil de Usuario' (User Profile) page. At the top, there's a green header bar with the Heimdall logo, 'Mis Tickets', 'Crear Ticket', and user profile buttons ('Perfil', 'Cerrar Sesión'). The main content area is titled 'Perfil de Usuario' and displays the following information:

Nombre de usuario	Nombre
nachorol	Nacho

Below this, there are two columns of information:

Apellido	Email
Roldan	nachorol@dendorgames.c...

Teléfono	Empresa
+34785123697	CORERIFT

On the left side of the profile card, there is a circular profile picture with the text 'CoreRift' below it.

Figura 5.20: Manual trabajador 04

Todos los usuarios tendrán acceso mediante su credenciales de inicio de sesión en la app, al webmail. Aquí recibirán los distintos avisos de sus tickets y otra información interesante para el resto de usuarios.

The screenshot shows a webmail inbox interface. On the left, there's a sidebar with navigation links: 'Entrada' (selected), 'Redactor', 'Correo', 'Contactos', and 'Configuraci...'. The main area shows an 'Entrada' folder with several emails listed. One email is selected, showing its details:

Ticket '77' Confirmado - A la Espera de Asignación

From: heimdall@dendorgames.es (Heimdall) on 2024-06-17 19:03
Subject: Ticket '77' Confirmado - A la Espera de Asignación

The email body contains the following text:

Le informamos que su ticket con el nº de incidencia '77' ha sido confirmado con éxito. Su solicitud está ahora a la espera de ser asignada a un técnico.

Nuestro equipo de soporte se pondrá en contacto con usted en breve para resolver su solicitud.

Agradecemos su paciencia y comprensión mientras procesamos su solicitud.

Si tiene alguna pregunta adicional o necesita más información, no dude en ponerse en contacto con nosotros.

Gracias por su atención.

Saludos cordiales,

@Heimdall

Figura 5.21: Manual trabajador 05

6. Capítulo VI: Conclusiones finales

El desarrollo de la aplicación de gestión de incidencias para el grupo empresarial nos ha permitido abordar de manera eficiente las necesidades específicas de las empresas involucradas. A lo largo del proyecto, se lograron varios resultados clave que han contribuido significativamente a mejorar la gestión interna.

En primer lugar, se ha logrado una mejora en la gestión de incidencias gracias a la implementación de un sistema centralizado de tickets. Este sistema permite un seguimiento más riguroso y una resolución más rápida de las incidencias, optimizando los tiempos de respuesta y la eficacia del equipo técnico.

Además, se ha desarrollado un dashboard con Plotly que proporciona a los administradores una visión clara de las estadísticas de las incidencias, facilitando la toma de decisiones informadas.

La integración eficiente de usuarios mediante Active Directory ha simplificado la gestión de credenciales y permisos, asegurando un acceso controlado y seguro a la aplicación.

La comunicación interna también ha mejorado significativamente con la implementación de un servicio de correos internos. Esto asegura que los usuarios estén informados en cada etapa del ciclo de vida de una incidencia.

La inclusión de manuales de resolución ha sido otra mejora destacada, ayudando a los técnicos a abordar problemas recurrentes de manera más eficiente y aumentando la productividad.

Aunque el proyecto ha alcanzado sus metas establecidas, existen varias áreas de mejora y expansión que podrían abordarse en el futuro. Entre estas, la ampliación de funcionalidades es clave, como la integración con otros sistemas de gestión empresarial (ERP) para una mayor automatización y control.

Desarrollar una versión móvil de la aplicación permitiría a los usuarios gestionar incidencias desde cualquier lugar y en cualquier momento, aumentando la flexibilidad y accesibilidad del sistema.

Añadir soporte multilingüe facilitaría el uso de la aplicación por parte de usuarios de diferentes regiones, aumentando su accesibilidad y adaptabilidad.

La integración de técnicas de análisis predictivo y aprendizaje automático podría anticipar problemas y optimizar la gestión de recursos.

Un sistema de inventariado para gestionar y rastrear activos y recursos de la empresa sería una valiosa adición, así como la implementación de un sistema de monitoreo de servicios y dispositivos para comprobar su estado en tiempo real y proporcionar alertas en caso de fallos.

En resumen, la aplicación ha cumplido con los objetivos planteados, proporcionando una solución integral y eficiente para la gestión de incidencias dentro del grupo empresarial. Las futuras líneas de desarrollo mejorarán la funcionalidad y usabilidad de la aplicación, garantizando su relevancia y adaptabilidad en un entorno empresarial en constante cambio.

Bibliografía

- [1] *Certbot*. URL: <https://github.com/certbot/certbot>. [Accedido: 10-06-2024].
- [2] *Django (framework)*. URL: [https://es.wikipedia.org/wiki/Django_\(framework\)](https://es.wikipedia.org/wiki/Django_(framework)). [Accedido: 10-06-2024].
- [3] *Dovecot*. URL: <https://es.wikipedia.org/wiki/Dovecot>. [Accedido: 10-06-2024].
- [4] Microsoft. *Introducción a Active Directory Domain Services*. URL: <https://learn.microsoft.com/es-es/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview>. [Accedido: 10-06-2024].
- [5] *MySQL*. URL: <https://es.wikipedia.org/wiki/MySQL>. [Accedido: 10-06-2024].
- [6] *Nginx*. URL: <https://es.wikipedia.org/wiki/Nginx>. [Accedido: 10-06-2024].
- [7] *Plotly*. URL: <https://en.wikipedia.org/wiki/Plotly>. [Accedido: 10-06-2024].
- [8] *Postfix*. URL: <https://es.wikipedia.org/wiki/Postfix>. [Accedido: 10-06-2024].
- [9] *Python*. URL: <https://es.wikipedia.org/wiki/Python>. [Accedido: 10-06-2024].
- [10] *Roundcube*. URL: <https://es.wikipedia.org/wiki/Roundcube>. [Accedido: 10-06-2024].
- [11] P. Ruiz. *Conceptos básicos en una estructura de Directorio Activo*. URL: <https://somebooks.es/3-2-conceptos-basicos-en-una-estructura-de-directorio-activo/>. [Accedido: 10-06-2024].
- [12] *Servidor HTTP Apache*. URL: https://es.wikipedia.org/wiki/Servidor_HTTP_Apache. [Accedido: 10-06-2024].

7. Anexo I: Documento de pruebas apor-tado por Carlos Montesino

Al crear el primer tinkerbell, en “Mis tickets” sale dos veces repetido, volviendo a pedir confirmación con un botón con un check en verde, al crear un segundo ticket diferente no se me pide nada.

Mis Tickets Crear Ticket Gestión de Tickets Historial de Tickets Manuales Gestión de manuales Administración Información Técnicos

TUS TICKETS
Listado de todos tus tickets

Nº INCIDENCIA	TRABAJADOR	CATEGORÍA	SUBCATEGORÍA	FECHA	ESTADO
57	paquitogo	Hardware	Equipo	14 de junio de 2024 a las 10:01	Confirmado
56	paquitogo	Incidencias Locales	Servicios	14 de junio de 2024 a las 10:01	Confirmado
55	paquitogo	Incidencias Locales	Servicios	14 de junio de 2024 a las 10:01	Confirmado

Las prioridades están “harcodeadas”?

Mis Tickets Crear Ticket Gestión de Tickets Historial de Tickets Manuales Gestión de manuales Administración Información Técnicos

HISTORIAL DE TICKETS
Busca y filtra tickets en el sistema

ID	TRABAJADOR	EMPRESA	CATEGORÍA	SUBCATEGORÍA	FECHA	ESTADO	PRIORIDAD
57	paquitogo	CYBERWAVE	Hardware	Equipo	14 de junio de 2024 a las 10:01	Confirmado	MÉDIA
56	paquitogo	CYBERWAVE	Incidencias Locales	Servicios	14 de junio de 2024 a las 10:01	Confirmado	ALTA
55	paquitogo	CYBERWAVE	Incidencias Locales	Servicios	14 de junio de 2024 a las 10:01	Confirmado	ALTA

Se me han asignado tres tickets mientras usaba la app, 2 de ellos se crearon hace más de dos horas y me acaban de aparecer.(no se si es como debe de funcionar)

The screenshot shows a web-based ticket management interface. At the top, there's a green header bar with various navigation links: 'Mis Tickets', 'Crear Ticket', 'Gestión de Tickets', 'Historial de Tickets', 'Manuales', 'Gestión de manuales', 'Administración', and 'Información Técnicos'. Below the header, the main content area has a title 'TICKETS ASIGNADOS' and a subtitle 'Consulta y gestiona los Tickets asignados a su usuario.' A table lists three assigned tickets:

ID	TRABAJADOR	EMPRESA	CATEGORÍA	SUBCATEGORÍA	FECHA	ESTADO	PRIORIDAD
62	domingoanillo	CYBERWAVE	Software	Office	14 de junio de 2024 a las 12:30	Confirmado	MEDIA
60	domingoanillo	CYBERWAVE	Software	Correo	14 de junio de 2024 a las 10:01	Confirmado	MEDIA
52	domingoanillo	CYBERWAVE	Hardware	Periféricos	14 de junio de 2024 a las 10:01	Confirmado	BAJA

Creale una página bonita para el forbidden y para el 404 que se ve muy feo así xD

The screenshot shows a browser window with the URL 'heimdall.dendorgames.es/ticket/escalar/62'. The page displays a large, bold '403 Forbidden' error message. The browser's address bar and a row of icons are visible at the top.

403 Forbidden