



Universidad
Zaragoza

Trabajo Fin de Grado

Desarrollo e implementación de un sistema de
reservas y control de aulas e integración en el
ecosistema sensoriZAR

Development and implementation of a book and
capacity control system in classrooms and
integration in sensoriZAR ecosystem

Autor

Aarón Ibáñez Espés

Directores

Raquel Trillo Lado

Ignacio Martínez Ruiz



DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D^a. Aarón Ibáñez Espés,

con nº de DNI 25203069Z en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)
Grado, (Título del Trabajo)
Desarrollo e implementación de un sistema de reservas y control de
aulas e integración en el ecosistema de sensoriZAR.

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 24 de Noviembre de 2022

Fdo: Aarón Ibáñez Espés

AGRADECIMIENTOS

En primer lugar, me gustaría agradecer los tutores de mi trabajo, Raquel Trillo Lado e Ignacio Martínez Ruiz por su labor realizada durante estos meses de trabajo. Me gustaría destacar la disponibilidad que siempre han tenido para resolver mis dudas, en cualquier momento, independientemente de la situación. Así mismo, agradecer a Alberto Mur, del equipo de sensoriZAR toda la ayuda prestada durante el trabajo que realicé con ellos, resolviendo cualquiera de mis dudas sobre la infraestructura del proyecto, explicándome como funcionaba la tecnología que utilizaban y abriéndome las puertas de su laboratorio en el I3A para poder ir a trabajar siempre que fuera necesario. Finalmente, destacar la colaboración de Sergio Ilarri por permitirme hacer uso del *cluster* Atila para el despliegue del sistema en su periodo de prueba piloto con el objetivo de poder recolectar datos y realizar un análisis posterior sobre ellos.

Por otra parte, me gustaría agradecer el apoyo recibido por parte de mi familia y amigos, mostrándose comprensivos en los momentos más difíciles y apoyándome durante el desarrollo de todo el proyecto.

Resumen

Debido a la situación sanitaria dada en los últimos años, se han presentado necesidades que previamente habían pasado desapercibidas o no habían sido consideradas. Dichas necesidades, junto con los *Objetivos de Desarrollo Sostenible* (ODS), la eficiencia energética y la construcción de edificios inteligentes, que cada vez están más presentes, abren la puerta al desarrollo de herramientas y tecnologías que contribuyan a los mismos. En este documento se describe el desarrollo e implementación de un sistema de gestión de reservas y control de aulas (denominado NeoB) y su integración con el ecosistema de sensoriZAR de monitorización inteligente de espacios de la Universidad de Zaragoza.

NeoB surge en el contexto de la pandemia generada por la CoViD (*Corona Virus Disease*) con el objetivo de cubrir una necesidad clara: facilitar al estudiantado la reserva de espacios para estudiar de forma segura y minimizar el contacto entre el personal de las salas y el estudiantado. Para ello, se desarrolla un sistema de información que crea un gemelo digital de cada una de las salas que presta ese servicio a través del cual los estudiantes son capaces de consultar todos los datos relevantes de las mismas, así como, realizar reservas para el seguimiento de las clases o estudio. Por otra parte, NeoB consta de un espacio para administradores, el cual facilita a los trabajadores la gestión de la sala, poniendo a su disposición la posibilidad de modificar el estado de los asientos o inhabilitar días de reserva debido a que la sala permanece cerrada, entre otras funcionalidades.

Además, del servicio que ofrece al estudiantado, NeoB se concibe como un sistema que permite a los administradores consultar la ocupación de la sala en tiempo real (asientos disponibles y ocupados) y que facilita la recopilación de datos sobre el uso que los estudiantes realizan de la misma. A partir de estos datos recopilados, se puede proceder a su análisis para proporcionar información de calidad para la realización de informes sobre la sala o para el apoyo a los procesos de toma de decisiones futuras basadas en datos.

Abstract

Due to the health situation in recent years, new needs have appeared. These needs, together with the sustainable development goals, the energy efficiency and the construction of intelligent buildings are becoming more important. Thus, nowadays, there exists a great chance for the development of tools and technologies that contribute to achieving these goals. This work describes the development and implementation of a reservation management and classroom control system and its integration with the sensoriZAR ecosystem, called NeoB.

NeoB is created during the pandemic in order to cover a clear need: facilitating students the reservation of spaces to study safely and minimizing the contact between the staff managing the rooms and the students. In order to offer that service, an information system has been developed. This information system provides each library with a digital twin where students are able to query all the relevant information about the different rooms and spaces, as well as make reservations. On the other hand, NeoB has also functionalities for administrators, which facilitate the management of the room. Thus, it is possible to modify the status of the seats (indicating if they are available or not), disabling reservation days due to the fact that the room is closed, etc.

In addition to the service offered to students, NeoB is conceived as a system that allows administrators to consult room occupancy in real time (available and occupied seats). Moreover, this system facilitates the recollection of data of the use of the different rooms by students. These data can be analyzed in order to provide quality information (for example, generating reports based on evidences) and in order to support future data-driven decision making.

Índice

1. Introducción	1
1.1. Objetivos	2
1.2. Metodología	3
1.3. Estructura de la memoria	3
2. Estudio del estado del arte	7
2.1. Estudio de aplicaciones de propósito similar	7
2.2. Tecnologías y <i>frameworks</i> principales utilizados para el desarrollo . . .	9
3. Análisis de requisitos	11
3.1. Requisitos del sistema	11
3.1.1. NeoB	11
3.1.2. NeoB Admin	12
3.2. Casos de uso del sistema	14
4. Diseño del sistema	15
4.1. Diseño de la base de datos	15
4.2. Diseño del <i>backend</i> y ambos <i>frontends</i>	19
4.2.1. Arquitectura del sistema	19
4.2.2. Diseño de la aplicación	20
5. Implementación y desarrollo	23
5.1. Tecnologías utilizadas	23
5.2. Decisiones de diseño durante el desarrollo	25
5.3. Automatización de la creación de códigos QR	26
5.4. Despliegue del sistema	28
5.4.1. Sistema operativo	28
5.4.2. Recursos asignados a la máquina virtual	28
5.4.3. Software utilizado para el despliegue	29

6. Validación experimental	31
6.1. Prueba piloto de la aplicación	31
6.1.1. Pegado de códigos	32
6.1.2. Formación de los bibliotecarios	32
6.1.3. Prueba piloto de uso del sistema	33
6.1.4. Evaluación de la aceptación del sistema	35
6.1.5. Análisis de los datos para el apoyo a la toma de decisiones. . . .	39
7. Análisis de riesgos y plan de mantenimiento del sistema	45
7.1. Plan de análisis y gestión de riesgos	45
7.2. Plan de mantenimiento y seguridad	48
8. Conclusiones y líneas futuras de trabajo	55
8.1. Conclusiones	55
8.2. Líneas de trabajo futuras	56
9. Bibliografía	59
Lista de Figuras	61
Lista de Tablas	63
Anexos	64
A. Mapas de navegación del sistema	67
B. Configuración completa de la máquina virtual e instrucciones de despliegue	71
B.1. Configuración de la máquina virtual	71
B.1.1. Instalación del sistema operativo	73
C. Trabajo realizado con sensoriZAR	79
C.1. Configuración de antenas Lorix One	79
C.1.1. Actualización del sistema operativo de la antena de lorix legacy a LorixOS	79
C.1.2. Dar de alta la antena en la instancia de tts privada de sensoriZAR	82
C.1.3. Configurar el forwarder de la antena	84
C.2. Montaje de las antenas en Veterinaria y San Francisco	87

D. Resultados de las encuestas	91
D.1. Encuesta realizada a los estudiantes	91
D.2. Encuesta realizada al personal de la biblioteca	94
E. Scripts de automatización para la generación de los códigos QR	97
F. Consultas e información extraída de ellas para la realización de las gráficas	101

Capítulo 1

Introducción

NeoB [1] es un sistema de reservas de plazas de estudio y control de aulas, el cual pretende facilitar el estudio a los estudiantes, garantizando una plaza en la biblioteca siempre que esta haya sido previamente reservada. Además, se persigue ayudar a los conserjes y administradores de las salas en las labores de gestión de las mismas. El sistema está separado en dos grandes espacios diferenciados por la labor de cada uno de ellos, el propio sistema NeoB, enfocado a los estudiantes, y el sistema de administración, NeoB Admin [2], enfocado a los administradores.

La idea del sistema surgió a comienzos de la pandemia, debido a las restricciones sanitarias establecidas, principalmente a las restricciones de aforo y distancia, y a la recomendación de minimizar el contacto entre las personas. Esto desembocó en la idea de NeoB como un sistema de reservas que facilitase a los estudiantes la posibilidad de comprobar el estado de su sala de estudio habitual, si esta se encontraba abierta, el aforo del que se disponía en función de las restricciones vigentes, las plazas ocupadas y las libres, así como, ofrecer la posibilidad de realizar una reserva de una plaza libre en una franja horaria preestablecida para el seguimiento de la docencia o estudio. Por ello, el sistema cuenta con un apartado denominado *protocolo covid*, en el cual se describen las recomendaciones y restricciones vigentes y se ofrece un formulario para reportar un caso positivo, el cual se hace llegar al responsable covid de la escuela, agilizando la gestión del positivo y su posterior actuación ante el mismo. Inicialmente, NeoB contaba con un espacio dentro de la misma página orientado a los administradores de las salas, pero, posteriormente, debido a que dicho apartado no estaba enfocado al mismo público que NeoB, se tomó la decisión de separarlo del sistema principal, desarrollando otro espacio llamado NeoB Admin, enfocado a el personal administrativo de las bibliotecas y salas. Dicho espacio, cuenta con un apartado de inicio de sesión, a partir del cual en función del rol asignado que tenga el administrador, puede acceder a dos paneles diferentes, al de gestión de las aulas asignadas, en caso de ser administrador

o al de gestión de personal y asignación de aulas, en caso de ejercer el rol de moderador.

Conforme se ha ido desarrollando NeoB, se ha dejado de ver únicamente como un sistema de reservas y control de aforos y se ha descubierto su potencial como sistema que pueda facilitar la digitalización de las aulas integrándolas en el ecosistema sensoriZAR [3] de monitorización inteligente de espacios, contribuyendo a garantizar las mejores condiciones posibles de docencia y estudio. Por ello, se plantea la creación de un gemelo digital de cada una de las aulas con las que se trabaja, para recopilar datos de su uso, aforo, horarios de máxima afluencia, plazas más demandadas y demás datos de interés que, posteriormente, puedan ser analizados para contribuir en dicho objetivo.

A partir de ahí y alrededor de dicha idea se ha planteado todo el desarrollo realizado a lo largo de este Trabajo Fin de Grado (TFG).

El código elaborado está disponible públicamente en el siguiente repositorio https://github.com/aaronainzon17/NeoB_TFG.git junto con su licencia.

1.1. Objetivos

El objetivo principal del proyecto es desarrollar e implantar un sistema de reservas y control de aforo, el cual permita a los estudiantes la reserva de plazas para el estudio y a los administradores gestionar las salas (las aulas comunes de la EINA) e integrar dicho sistema en el ecosistema sensoriZAR.

Para ello, se han definido una serie de problemas a abordar que todos ellos contribuyen al alcance del objetivo principal del proyecto:

- Generalizar y adaptar NeoB para que pueda emplearse en cualquier espacio de forma sencilla.
- Integrar el sistema desarrollado en el ecosistema sensoriZAR desplegando una red LoraWAN que permita la recogida de datos procedentes de sensores de ocupación.
- Implantar el sistema en la biblioteca de Hypatia de Alejandría para, posteriormente, analizar y evaluar dicho trabajo de implantación y uso por parte de los usuarios.

1.2. Metodología

La metodología de trabajo seguida durante el desarrollo del proyecto ha sido una metodología iterativa. Esta se ha considerado la mejor opción dado que durante el desarrollo pueden surgir inconvenientes, nuevas ideas o distintos enfoques para solucionar un mismo problema que pueden ser de interés para el mismo. Además de ello, se han realizado varias presentaciones del sistema al personal de dirección de la biblioteca, por lo que sus ideas y aportaciones al proyecto se han tenido en cuenta, pudiendo implementarlas con mayor eficacia gracias a la metodología elegida.

Este desarrollo de mejora continua se ha realizado basado en los ciclos de Deming o *Plan-Do-Check-Act* (PDCA). Para ello, se han definido las 4 tareas en cada una de las iteraciones de la siguiente manera.

- Durante la fase de *Plan* se establecen los objetivos y tareas a abordar en el correspondiente ciclo o fase de la espiral, considerando los potenciales riesgos.
- Durante la fase *Do* se realiza el desarrollo e implementación de las tareas definidas en el apartado anterior.
- Tras ello, en el tercer paso, *Check*, se realiza una validación del desarrollo realizado y evaluación del mismo.
- Finalmente, en la fase de actuación, *Act*, se definen nuevas tareas en función de los resultados de la evaluación.

La duración de cada ciclo o fase de espiral ha sido de 15 días, aproximadamente, y en cada fase se han realizado reuniones de seguimiento y coordinación semanales.

1.3. Estructura de la memoria

Durante el desarrollo del presente documento se detalla el proceso de desarrollo del sistema de reservas y control de aforos NeoB. A continuación, se detalla de forma resumida el contenido de cada uno de los capítulos.

Estudio del estado del arte 2. En este capítulo, se realiza el estudio de aplicaciones de propósito similar para determinar si existen otras aplicaciones que cubren unos requisitos similares a la propuesta. En función de las conclusiones extraídas de este apartado se optó por el desarrollo de un sistema a medida en lugar de adoptar uno ya existente y adaptarlo.

Análisis de requisitos 3. A lo largo de este capítulo, se detallan los requisitos que debe cumplir el sistema para satisfacer las necesidades de los usuarios. Además de ello, también se enumeran los casos de uso del sistema.

Diseño del sistema 4. En este capítulo, se detalla el diseño de los diferentes espacios de la aplicación, comenzando por el diseño de la capa de persistencia de datos. A continuación se describe la arquitectura del sistema y, finalmente, la interfaz gráfica de la aplicación.

Implementación y desarrollo 5. Durante este capítulo se describen en mayor profundidad las tecnologías utilizadas en el sistema, así como las decisiones de diseño que se tomaron durante la fase de desarrollo y, finalmente, se aborda el despliegue de la aplicación.

Validación experimental 6. En este apartado se detalla el proceso de formación del personal de la biblioteca para la utilización del sistema. Además, se detallan los resultados y modificaciones realizadas al sistema durante la prueba piloto. Para finalizar, se realiza un análisis de los datos de uso obtenidos para el apoyo a la toma de decisiones.

Análisis de riesgos y plan de mantenimiento del sistema 7 A lo largo de este capítulo, se realiza un análisis de los riesgos a los que se encuentra expuesto el sistema utilizando la metodología MAGERIT y se define un plan de mantenimiento para asegurar el correcto funcionamiento del mismo.

Conclusiones y líneas futuras de trabajo 8 En este último capítulo, se expone como se han logrado alcanzar los objetivos del proyecto, detallando el trabajo realizado para cada uno de ellos. Además, se ofrece una conclusión sobre el trabajo realizado y se enumeran posibles vías de desarrollo futuras.

En la presente documentación también se proporciona un conjunto de anexos donde se describen diferentes aspectos referenciados a lo largo de los diferentes capítulos en mayor profundidad. En concreto, este trabajo contiene los siguiente anexos:

Anexo A. En este apartado se incluyen los mapas de navegación del sistema.

Anexo B. En este anexo se enumera paso a paso cómo crear y configurar la máquina virtual para desarrollar e implantar el sistema de información desarrollado, es decir cómo realizar el despliegue del sistema.

Anexo C. En este anexo se detalla el trabajo realizado en colaboración con sensoriZAR para la configuración de varias antenas Lorix One y su posterior colocación en los campus de Veterinaria y San Francisco, contribuyendo en el despliegue de la red LoraWAN para la recogida de datos por parte de los sensores.

Anexo D. Se incluyen los resultados extraídos de las encuestas realizadas tanto a los estudiantes como al personal de la biblioteca sobre el sistema y su uso.

Anexo E. En este anexo se dispone el código de los *scripts* realizados para la automatización de la creación de las etiquetas con los códigos QR.

Anexo F. Como complemento al apartado en el que se detalla el análisis de los datos obtenidos para el apoyo a la toma de decisiones, en este anexo se detallan las consultas realizadas a la base de datos para obtener la información necesaria.

Capítulo 2

Estudio del estado del arte

Tras identificar la necesidades que se quieren cubrir con el sistema propuesto en el proyecto, se realiza el estudio del estado del arte para conocer las aplicaciones de propósito similar existentes y que necesidades cubren. En función de los resultados encontrados y de la similitud de los sistemas que se consideran similares con el que se propone, se toma la decisión final de proceder con el desarrollo del sistema, modificarlo para cubrir alguna otra necesidad que no se había tenido en cuenta previamente o seguir con su desarrollo previsto. Además, la realización del estudio del estado del arte permite ser consciente de las aplicaciones existentes y facilita el encontrar un factor diferenciador en el sistema para diferenciarse de la competencia.

Se han analizado aplicaciones de reserva de espacio de sistemas de venta de entradas de cine y de medios de transporte (aviones, autobuses, etc.), como, por ejemplo el sistema de Renfe [4], Cinesa [5] y Ryanair [6]. Sin embargo, dichos sistemas son propietarios y el coste de comprarlos y adaptarlos era elevado; es más, muchos de ellos no están disponibles para su venta. Por ello, se optó por un desarrollo propio ajustado a las necesidades específicas de la Escuela de Ingeniería y Arquitectura.

Por otro lado, existen plugins de Moodle basados en tecnología *bluetooth* que son capaces de determinar qué personas se encuentran en una determinada sala. Sin embargo, esta opción se considera más invasiva. De ahí el que se haya descartado.

2.1. Estudio de aplicaciones de propósito similar

La única aplicación que se ha encontrado con un propósito similar en la Universidad de Zaragoza ha sido reserva de aulas [7]. Esta aplicación permite la realización de solicitudes de reserva para diferentes espacios en el campus, desde aulas grandes, pequeñas o seminarios a aulas de informática. Dichas solicitudes pueden realizarse para diferentes propósitos como la docencia, realización de exámenes o alquiler del espacio.

Sin embargo, todo lo que se puede realizar desde la aplicación es la solicitud de reserva que, posteriormente, tendrá que ser aceptada por parte de los administradores vía correo electrónico. Por otro lado, en este sistema no se ofrece la posibilidad de reservar un asiento de estudio en una de las bibliotecas para garantizar el asiento. Por ello, se considera que pese a poder cubrir una necesidad similar, no cubre la necesidad principal que se desea satisfacer con el sistema propuesto.

The screenshot displays the 'Meeting Room Booking System' interface. At the top, it identifies the 'Servicio de Informática y Comunicaciones' and the 'Área de Ordenadores Personales'. A navigation bar includes links for 'Help', 'Admin', 'Ver Solicitudes', 'Report', 'Search', and a user profile 'You are 772088'. Below this, there are sections for 'Edificios' (listing various university buildings) and 'Aulas' (listing different room types like 'Aulas Grandes(pupitres)', 'Aulas pequeñas(pupitres)', etc.). The main part of the interface is a calendar for October and November 2022. Below the calendar, a specific timetable for 'martes 18 octubre 2022' is shown for the 'Rio Ebro - EINA - Agustín de Betancourt - 1-Aulas Grandes(pupitres)' room. This timetable lists various courses and their scheduled times, such as '045 Fotografía Composición y Edición de Imágenes (prob 1.2)' and '821 Fundamentos de Ingeniería de materiales (prob 1.2)'. The interface is color-coded with different background colors for different rooms or courses.

Figura 2.1: Página principal de reservadeaulas.

Además del sistema analizado anteriormente, se ha encontrado un apartado en la Web de la Universidad donde se ofrecen diferentes documentos para la solicitud de alquiler de aulas. Sin embargo, al no ser un sistema como tal y solamente ofrecer un impreso para rellenar y presentar la solicitud a través de otra plataforma como, por ejemplo, *regtel*, no se ha tenido en cuenta por considerar que no cubre ninguna necesidad de las que se quieren cubrir con NeoB.



Figura 2.2: Apartado de la pagina de la universidad donde se dispone de los impresos para la solicitud de alquiler de espacios.

2.2. Tecnologías y *frameworks* principales utilizados para el desarrollo

A lo largo del desarrollo del software que da soporte al sistema propuesto se han empleado las siguientes tecnologías y *frameworks*:

- **React JS [8]**: es una biblioteca de JavaScript de código abierto que permite la creación de interfaces de usuario de forma sencilla, declarativa y flexible. Parte del concepto de componente para la creación de interfaces de usuario más complejas. Un componente se concibe como un fragmento de código encapsulado que gestiona su propio estado y puede ser reutilizado. Además, de ello, React se encarga de gestionar de forma inteligente los componentes, actualizando solamente el mínimo fragmento de código necesario cuando los datos cambian, por lo que la fluidez y velocidad de la aplicación se ve mejorada. Con el uso de React se permite la creación de aplicaciones de pagina única o *Single Page Application* (SPA).
- **Express JS [9]**: es un *framework* web de *Node* para el desarrollo de *backend*. Express se ofrece bajo una licencia de software libre y se define como un *framework* minimalista y flexible debido a la cantidad de módulos que ofrece para su instalación y gestión de *cookies*, inicios de sesión, enrutado y muchos otros. Se tomó la decisión de utilizar Express para el desarrollo del *backend* debido a que es uno de los *framework* más populares de *Node*, con una gran comunidad y soporte. Además, este *framework* ofrece otras ventajas, como una API de enrutado que simplifica la creación de rutas, un gran rendimiento y alta escalabilidad.
- **PostgreSQL [10]**: es un gestor de código abierto de bases de datos objeto-relacionales que utiliza el lenguaje SQL y consta de muchas funcionalidades para almacenar datos y gestionar y escalar flujos de trabajo complejos. La reputación de este gestor se ha incrementado debido a la gran comunidad de desarrolladores que colaboran para ofrecer constantemente soluciones innovadoras y de alto rendimiento. Este gestor es el que se ha decidido utilizar en el sistema debido a que ofrece una serie de características y funcionalidades útiles.

En primer lugar, la instalación de este software es ilimitada y gratuita, PostgreSQL está disponible para un amplio abanico de sistemas operativos, lo que lo convierte en un sistema gestor de bases de datos multiplataforma, esto se considera una ventaja ya que el equipo donde se vaya a instalar no supondrá una restricción. Además, este gestor implementa todas las funcionalidades del

estándar SQL ISO/IEC 9075:2011 [11], por lo que facilita su integración con otros motores de bases de datos.

Otro punto a favor de este gestor es su robustez, eficiencia y estabilidad, ya que cumple con las características *Atomicity, Consistency, Isolation and Durability* (ACID). Por ello, permite que las transacciones no interfieran entre ellas y garantiza que la información de la base de datos sea siempre consistente. Esto se considera un gran punto a favor, ya que el sistema tendrá que gestionar gran cantidad de inserciones debido a que tiene que registrar todas las reservas realizadas en el sistema.

Además de cumplir con las características ACID, posee un sistema de control de alta concurrencia multiversión (MVCC), el cual permite que mientras un proceso escribe una tabla, otros puedan acceder a la misma tabla sin necesidad de verse bloqueados, y cada usuario obtiene una visión consistente.

Como último punto a favor dispone de una herramienta, pgAdmin, muy fácil e intuitiva para la administración de las bases de datos.

Además de todas las ventajas comentadas anteriormente, uno de los factores determinantes de la elección de PostgreSQL como el sistema gestor de bases de datos de la aplicación es que ya se había utilizado previamente y se estaba familiarizado con el mismo.

Capítulo 3

Análisis de requisitos

La fase de análisis y especificación de requisitos es el momento en el que se analizan las necesidades del usuario para poder definir los requisitos que debe de cumplir el sistema final. Cada uno de los requisitos define las prestaciones que debe satisfacer cada funcionalidad en el sistema. En este capítulo se especifican tanto los requisitos funcionales y no funcionales del sistema, como los casos de uso del mismo.

3.1. Requisitos del sistema

Dado que se dispone de dos sistemas con distintos requisitos (uno para usuarios de las salas -NeoB- y otro para la administración de las mismas -NeoB Admin-), se ha separado el análisis de los mismos en dos subapartados diferentes. Además, para cada uno de los dos subsistemas se analizan los requisitos funcionales y no funcionales.

3.1.1. NeoB

En el primer subapartado se realiza el análisis de requisitos de NeoB, el sistema enfocado a los estudiantes.

Requisitos funcionales

Requisito	Descripción
RF1	El sistema debe identificar a los usuarios como estudiantes de UNIZAR para permitirles hacer uso del sistema.
RF2	El sistema debe permitir consultar las salas disponibles para el estudio.
RF3	El sistema debe permitir consultar el estado de cada una de las salas para cada franja horaria de estudio.
RF4	El sistema debe permitir al estudiante realizar un máximo de una reserva de forma simultánea.
RF5	El sistema debe permitir a los usuarios modificar sus reservas.
RF5.1	El sistema debe permitir al estudiante cancelar su reserva si se ha equivocado o finalmente no va a estudiar en dicha sala.

RF5.2	El sistema debe permitir al usuario finalizar el uso de una plaza antes de que se agote el tiempo de estudio, permitiendo ser reservada por otra persona.
RF6	El sistema debe limitar las reservas permitidas a una por estudiante por cada una de las salas disponibles.
RF7	El sistema debe permitir al usuario elegir la sala de estudio o biblioteca deseada.
RF8	El sistema debe permitir al usuario elegir el asiento deseado.
RF9	El sistema debe permitir al usuario elegir el horario de estudio deseado, siempre y cuando queden asientos disponibles.
RF10	El sistema debe permitir reportar casos de COVID 19.
RF11	El sistema deberá notificar al responsable covid en caso de que se reporte un positivo en COVID 19.

Tabla 3.1: Especificación de requisitos funcionales de NeoB.

Requisitos no funcionales

Requisito	Descripción
RNF1	Para garantizar la accesibilidad a la web a todos los usuarios, el sistema debe ser adaptativo a cualquier tipo de pantalla.
RNF2	Se tendrá en cuenta la paleta de colores elegida y el diseño de interfaz realizado, para conseguir una interfaz intuitiva y sencilla de utilizar, para hacerla accesible a todos los estudiantes.
RNF3	No será necesario el inicio de sesión para realizar una reserva, solamente la identificación con el correo de UNIZAR o asociados.
RNF4	El sistema debe estar disponible para prestar servicio de lunes a sábado.
RNF5	El sistema debe recuperarse en un tiempo máximo de 30 minutos.

Tabla 3.2: Especificación de requisitos no funcionales de NeoB.

3.1.2. NeoB Admin

A continuación, se realiza el análisis de requisitos de NeoB Admin, sistema enfocado al personal de la biblioteca.

Requisitos funcionales

Requisito	Descripción
RF1	El sistema debe permitir el registro de usuarios acorde al rol que desempeña cada uno.
RF1.1	El sistema debe diferenciar entre dos tipos de usuarios, administrador y moderador, con distintos permisos de administración.
RF1.2	El rol de administrador debe tener permisos para la administración y gestión de las salas, mientras que el moderador debe tener los permisos para gestionar a los administradores.
RF2	El sistema permitirá a los moderadores crear nuevos usuarios de cualquier rol.

RF2.1	El sistema permitirá crear cuentas temporales de ambos roles.
RF2.2	Tras la creación de una cuenta, el sistema enviará un correo con la contraseña de acceso temporal al mail asociado.
RF3	El sistema permitirá el cambio de la contraseña de acceso a la cuenta.
RF4	El sistema permitirá a los moderadores eliminar usuarios de cualquier rol.
RF5	El sistema permitirá la asignación de gestión de salas por parte de los moderadores a los administradores.
RF5.1	La asignación de gestión de una sala puede ser de forma permanente o temporal, fijando una fecha de caducidad.
RF6	El sistema permitirá a un moderador dar de baja en la gestión de una sala a un administrador.
RF7	El sistema debe mostrar a los administradores un panel con todas sus salas asignadas.
RF8	El sistema debe mostrar a los moderadores un panel con todos los usuarios del sistema.
RF9	El sistema debe permitir consultar el estado de una sala a todos sus administradores asignados.
RF9.1	En el estado de la sala se debe poder consultar su aforo máximo disponible, el número de asientos reservados, el número de asientos ocupados y un mapa de la sala que proporcione una representación gráfica de su ocupación para cada uno de los tramos horarios definidos.
RF10	El sistema debe permitir a los administradores modificar el aforo disponible en la sala, así como el estado de cada uno de los asientos.
RF10.1	El sistema dispone de 5 estados para cada uno de los asientos: no disponible, libre, ocupado, escaneado y sucio.

Tabla 3.3: Especificación de requisitos funcionales de NeoB Admin.

Requisitos no funcionales

Requisito	Descripción
RNF1	El sistema debe ser sencillo e intuitivo de usar, para facilitar la labor de los conserjes y administradores e incentivar su uso.
RNF2	El registro de ambos tipos de usuarios se llevará a cabo mediante usuario y contraseña.
RNF3	La creación de nuevas cuentas de administración se realizará de forma manual por un usuario moderador.
RNF4	Solamente se podrán crear usuarios de cuentas pertenecientes a UNIZAR y asociados.
RNF5	La paleta de colores utilizada para la representación del estado de los asientos debe ser clara y explicativa.
RNF6	El sistema debe estar disponible para prestar servicio de lunes a sábado.
RNF7	El sistema debe recuperarse en un tiempo máximo de 30 minutos.

Tabla 3.4: Especificación de requisitos no funcionales de NeoB Admin.

3.2. Casos de uso del sistema

En el sistema se pueden diferenciar tres tipos de usuarios, los estudiantes, los administradores de las salas y los moderadores del sistema, teniendo diferentes funcionalidades disponibles en el sistema en función de cada tipo de usuario. Por ello, a continuación, se ha realizado la especificación de los casos de uso para cada uno de los tres tipos de usuario.

– **Estudiantes:**

- Consultar el aforo de una biblioteca en una determinada franja horaria.
- Consultar la ocupación de una biblioteca en una determinada franja horaria.
- Consultar la ocupación de la biblioteca en tiempo real.
- Realizar una reserva de un asiento para estudiar en una biblioteca en una determinada franja horaria.
- Realizar una reserva de forma instantánea al llegar a la sala escaneando el código QR.
- Cancelar una reserva previamente realizada.
- Finalizar su estancia en la biblioteca antes de que acabe su reserva para liberar el asiento.

– **Administradores:**

- Consultar la ocupación de la biblioteca en tiempo real.
- Consultar el aforo de una biblioteca en una determinada franja horaria.
- Consultar la ocupación de una biblioteca en una determinada franja horaria.
- Consultar la ocupación que ha habido en la sala en franjas horarias previas.
- Modificar el estado de los asientos.
- Modificar reservas.
- Modificar el estado de la sala.

– **Moderadores:**

- Creación de usuarios de tipo administrador y moderador tanto permanentes como temporales.
- Eliminación de cuentas tanto de tipo administrador como moderador.
- Asignación de salas de forma temporal o permanente a los administradores.
- Eliminar permiso de administración a usuarios de tipo administrador.

Capítulo 4

Diseño del sistema

Tras haber definido los requisitos del sistema en el apartado anterior, en este se va detallar el diseño del sistema propuesto para cumplir con dichos requisitos. En primer lugar, se va a comentar el modelo de persistencia de datos utilizando un esquema entidad relación (E/R) [12] de la base de datos, tras ello, se comentará el diseño del *backend* y ambos *frontends* y, finalmente, la arquitectura del sistema, apoyándonos en los diagramas necesarios para su comprensión.

4.1. Diseño de la base de datos

El diseño de la base de datos es el proceso por el cual se lleva a cabo la organización de los datos según un modelo definido. En el diseño se tienen en cuenta cuales son los datos que se quieren almacenar y como se relacionan entre ellos. Por ello, una vez se han especificado los requisitos del sistema, se deciden cuales son los datos necesarios para que el sistema funcione correctamente y cuales son los datos que se quieren almacenar sobre las reservas, salas y usuarios y como se relacionan entre sí para proceder con el diseño de un modelo conceptual para la organización de los mismos llamado modelo entidad relación (E/R).

A continuación, se comentan las decisiones de diseño tomadas para el esquema y organización de los datos que se ha decidido utilizar en el sistema.

- Los tres tipos de usuarios del sistema se han modelado utilizando la herencia entre entidades. Esta relación se ha definido de participación total, es decir, todo usuario tiene que pertenecer a alguno de los tres roles disponibles y disjunta, para no dar lugar a redundancia de cuentas de usuario en el sistema. Cabe destacar que para definición de los usuarios de tipo *ADMINISTRADOR* y *MODERADOR* se ha tomado la decisión de modelarlos haciendo uso de la misma entidad ya que ambos se consideran usuarios administradores pese a su distinta labor.

- De igual manera que en el punto anterior, se ha definido la herencia de la entidad aula. En ella se ha definido un tipo de herencia de participación total y disjunta, ya que todo aula debe ser de algún tipo definido en el sistema y cada aula puede ser de solamente un tipo. Esta definición permite la incorporación sencilla en un futuro de nuevos tipos de aulas, lo que facilita el mantenimiento y la extensión del sistema.
- Se ha definido la entidad *Asiento* como entidad débil de *Aula* ya que la clave primaria de la entidad asiento depende de la entidad *Aula*. En caso de no existir aulas, sería imposible que hubiese asientos.

Además de las decisiones de diseño tomadas para el modelado del esquema E/R, también se ha tenido en cuenta la privacidad del usuario. Para ello, se ha decidido almacenar en la base de datos la mínima información necesaria sobre los mismos. Por parte de los estudiantes, solamente se almacena el NIP de cada uno para poder tener un identificador en las reservas. Por parte de los usuarios administradores, se almacenará el usuario y la contraseña cifrada para poder hacer *log in*, además del tipo de usuario administrador del que se trata.

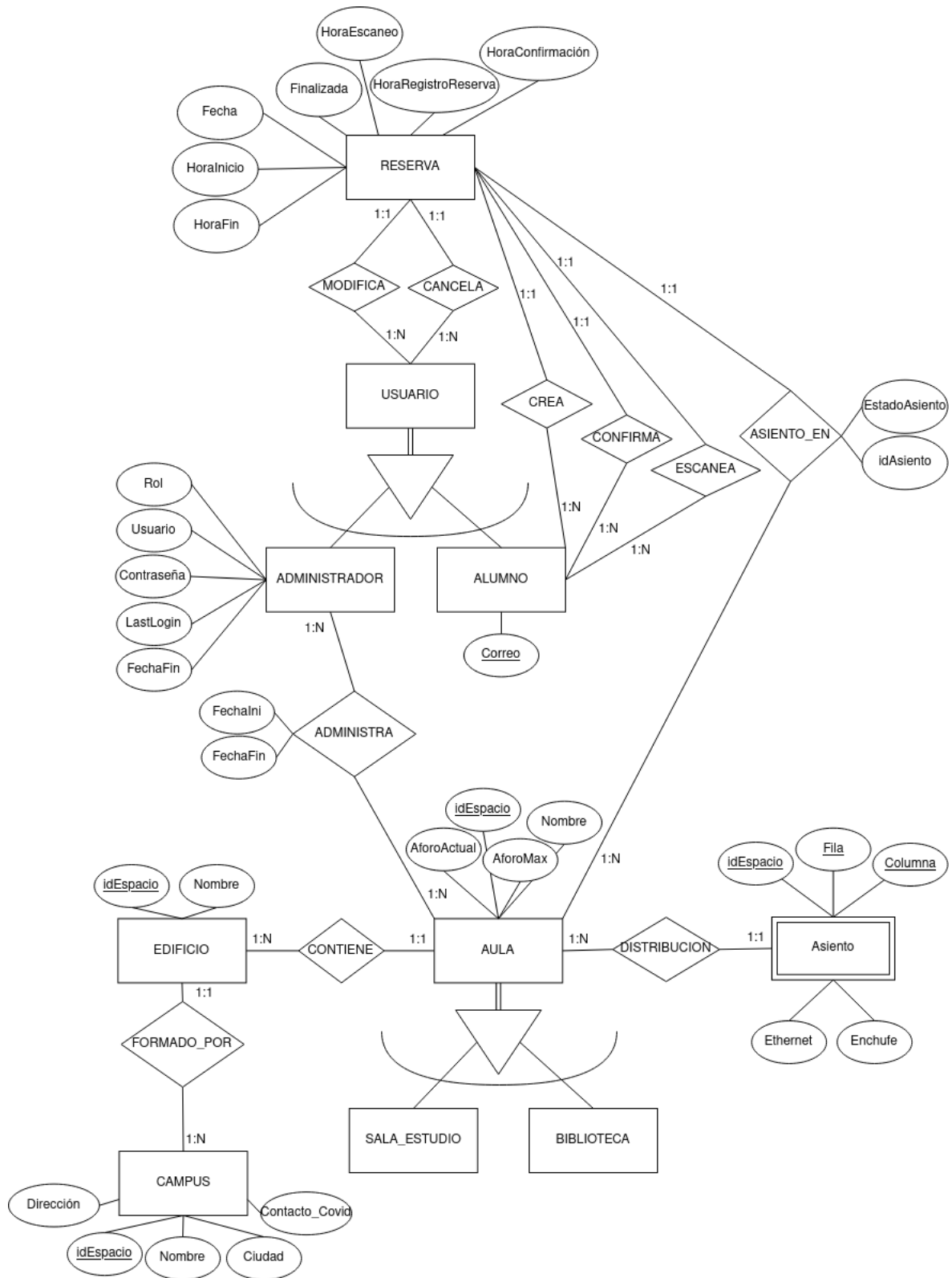


Figura 4.1: Esquema E/R.

Tras el diseño del modelo E/R, se traduce ese modelo en un modelo lógico que pueda servir de utilidad para la creación de la base de datos, en este caso se ha decidido utilizar el modelo relacional.

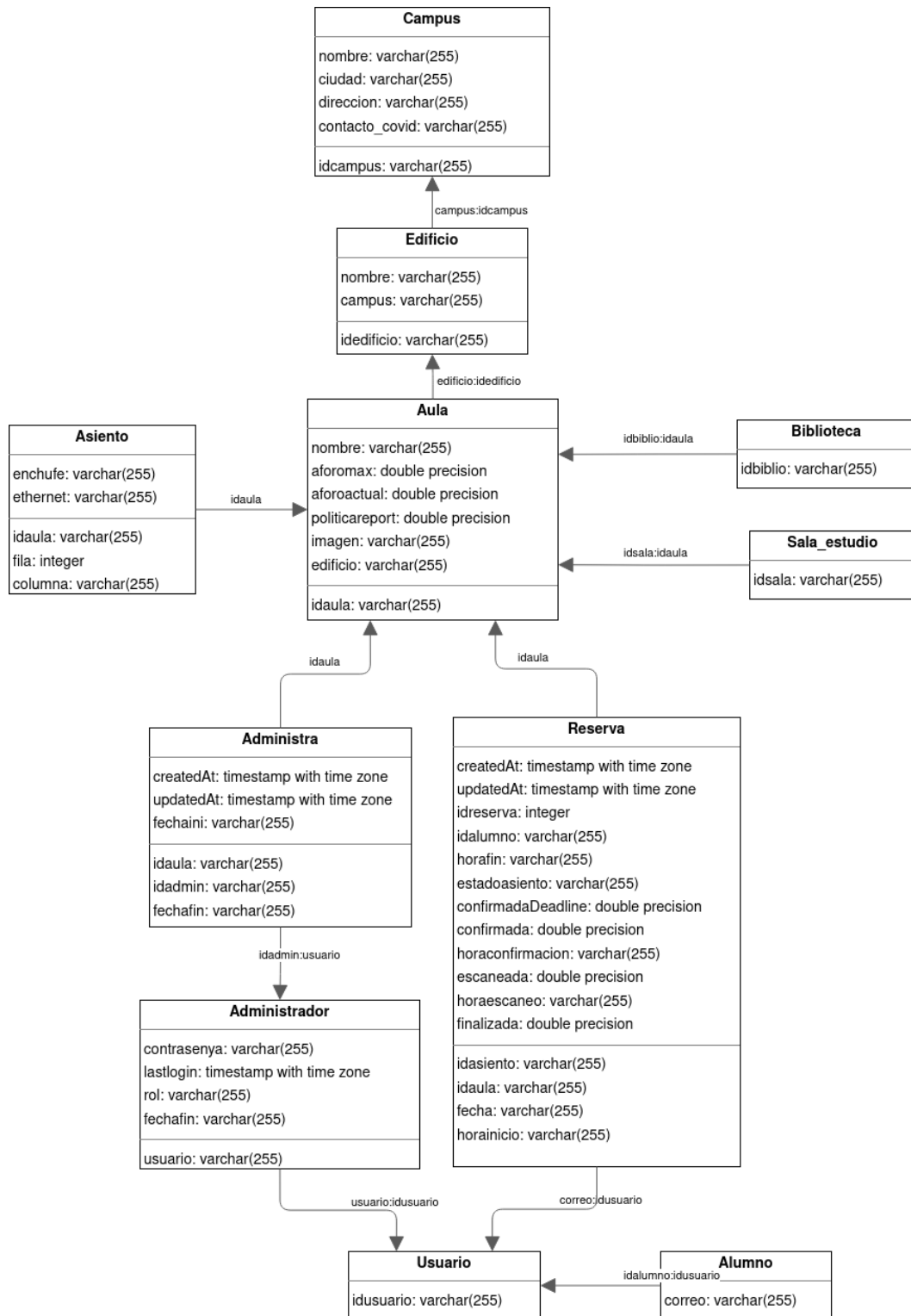


Figura 4.2: Modelo relacional.

4.2. Diseño del *backend* y ambos *frontends*

La fase de diseño del sistema es crucial a la hora del desarrollo, si se realiza un buen diseño de cada una de las capas del sistema se consigue que la aplicación funcione de la forma esperada. Realizando un buen diseño en la base de datos se consigue que los datos queden perfectamente estructurados y organizados, lo que facilita el tratamiento de los mismos por el *backend* de la aplicación, minimizando así tiempos de carga, reduciendo el número de consultas necesarias para cada operación y asegurando que se cumplen los requisitos no funcionales que se han especificado en el apartado anterior.

De la misma forma ocurre con el *frontend*, a la hora de realizar el diseño, es clave que sea un diseño sencillo, intuitivo y que invite a su uso ya que es el usuario el que decide si usar la aplicación o no y esta tiene que invitar a ello.

Por ello, a continuación, se cuenta el diseño de las capas lógica y visual del sistema:

4.2.1. Arquitectura del sistema

El sistema consta de tres componentes principales encargados de la lógica de la aplicación y una base de datos que conforma la capa de persistencia de datos de la misma.

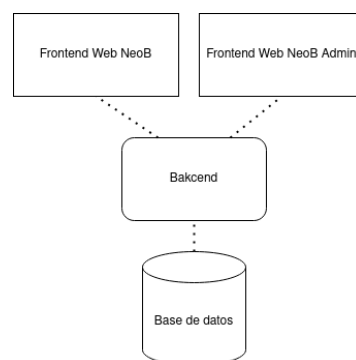


Figura 4.3: Arquitectura del sistema.

- **Backend:** Es el componente principal del sistema, este se encarga de procesar las peticiones que llegan desde ambos *frontends* y comunicarse con la base de datos para almacenar nuevas reservas o recuperar datos sobre los aforos ocupación y reservas en las aulas.
- **Frontend de uso general:** Esta es la parte visual para los usuarios estudiantes. Desde este *frontend* se ofrece la posibilidad de consultar las salas disponibles en

el sistema, su ocupación y realizar una reserva, entre otros.

- **Frontend de administración:** Esta es la parte visual enfocada a los administradores. Desde este espacio se facilitan tareas enfocadas a la administración de la sala, como la consulta de la ocupación, modificación de los aforos, eliminación de reservas, entre otras.
- **Base de datos:** En la base de datos se almacenan los datos necesarios para el correcto funcionamiento de la aplicación, salas disponibles, asientos, aforos, etc. Además de ello, se almacenan las reservas realizadas y demás información útil.

El *backend* posee un diseño en 3 capas siguiendo el patrón Modelo, Vista, Controlador (MVC). El uso de este patrón facilita la separación del código en distintas responsabilidades manteniendo distintas capas que se encargan de hacer una tarea muy concreta, lo que ofrece muchos beneficios a la hora del mantenimiento, reutilización de código y separación de conceptos en el sistema.

- **Modelo:** Esta capa es la responsable de la gestión de los datos. En ella se encuentran los mecanismos de comunicación con la base de datos.
- **Vista:** Es la capa encargada de recibir los datos del modelo y devolvérselos al usuario.
- **Controlador:** Recibe las peticiones de entrada y hace consultas al modelo cuando se requiere de información almacenada en la base de datos. Esta capa sirve de enlace entre las vistas y los modelos.

4.2.2. Diseño de la aplicación

La primera impresión de un usuario sobre la aplicación es la más importante, en función de esa impresión, el usuario tomará la decisión de utilizar el sistema o por el contrario, si la primera experiencia no ha sido satisfactoria, podría tomar la decisión de rechazarlo y no utilizarlo. Por ello, el diseño de la capa de visualización del sistema es un apartado fundamental en el que se trabaja para que la experiencia de usuario sea lo mejor posible.

En cuanto al diseño de la interfaz de usuario, es relevante mantener una consistencia en el diseño para que el usuario se familiarice con la aplicación y no haya diferentes diseños en cada uno de los apartados de la página. La prioridad es conseguir una interfaz atractiva y fácil de usar, y que además, se diferencie de otras páginas similares, tratando de realizar una interfaz minimalista, sencilla e intuitiva.

Para los usuarios es mas sencillo familiarizarse con un nuevo sistema si éste es similar a otros que han utilizado previamente o recuerdan a ellos. Por ello, la paleta de colores elegida para el sistema trata de ser muy similar a la paleta de colores de la Universidad de Zaragoza (UNIZAR) [13]. De esta manera los usuarios serán capaces de relacionar este servicio con la Universidad más fácilmente.

Además, se ha trabajado en la interfaz para que esta sea sencilla, utilizando un lenguaje fácilmente comprensible en el texto presente en el sistema, minimizando la cantidad de botones sobre los que el usuario debe hacer *click* para realizar cualquier consulta y por lo tanto priorizando la navegación por el sistema con el mínimo número de *clicks* necesarios.

Para que el usuario se sienta cómodo durante la experiencia con el sistema, se ha incluido durante el proceso de reserva un botón de “Anterior”, con el que permite volver al paso anterior para consultar los datos introducidos, corregirlos o incluso deshacer todo el proceso de reserva, dando al usuario la sensación de control sobre el proceso que está realizando y generando más confianza en el mismo.

Finalmente, dado que se asume que el usuario no solamente va a acceder a la aplicación desde el ordenador, se ha realizado un diseño *responsive*, es decir, la interfaz del sistema se adapta al tamaño y orientación de la pantalla en la que se muestra, para que toda la información se muestre de forma correcta. Este punto se ha considerado vital en el diseño del sistema, ya que pese a ser una página web, se prevé que la mayoría de los usuarios accedan desde sus dispositivos móviles ya que con ellos se escanea el código QR.

A continuación, se incluyen los mapas de navegación resumidos realizados para cada uno de los espacios. En el anexo A se pueden consultar los mapas de navegación completos del sistema.

Mapa de navegación de NeoB

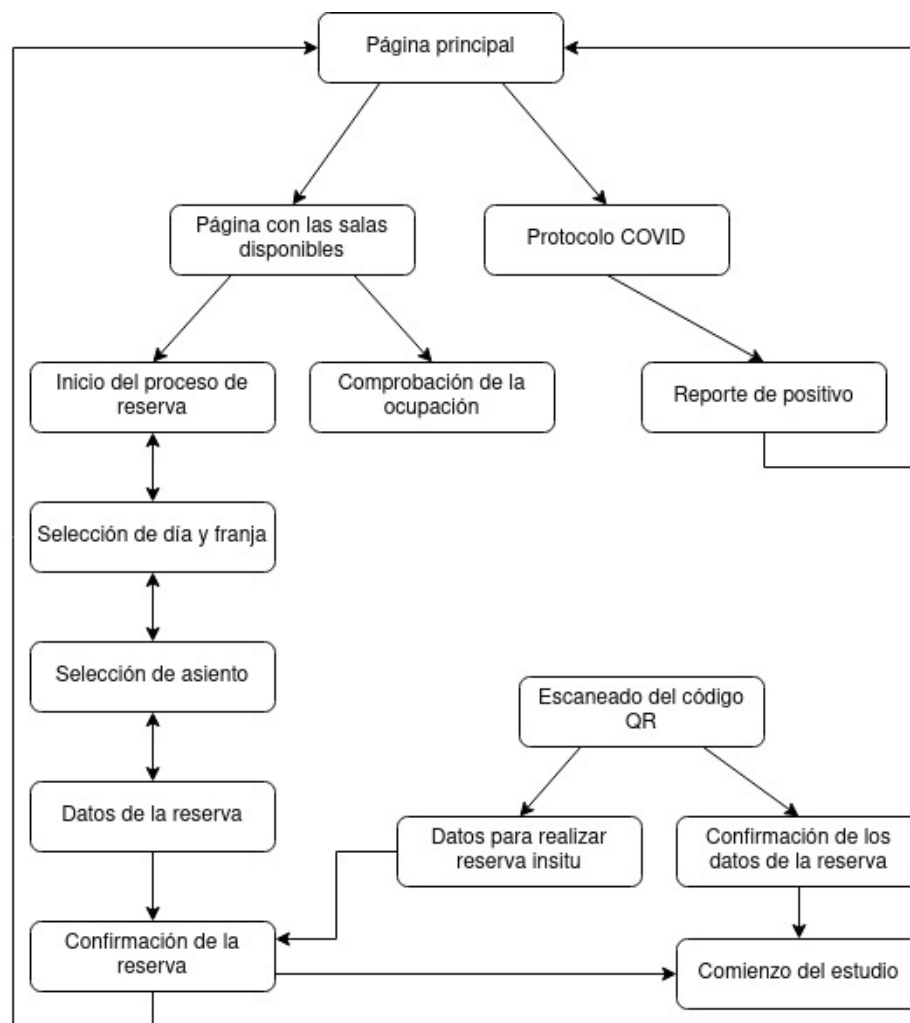


Figura 4.4: Mapa de navegación de NeoB.

Mapa de navegación de NeoB Admin

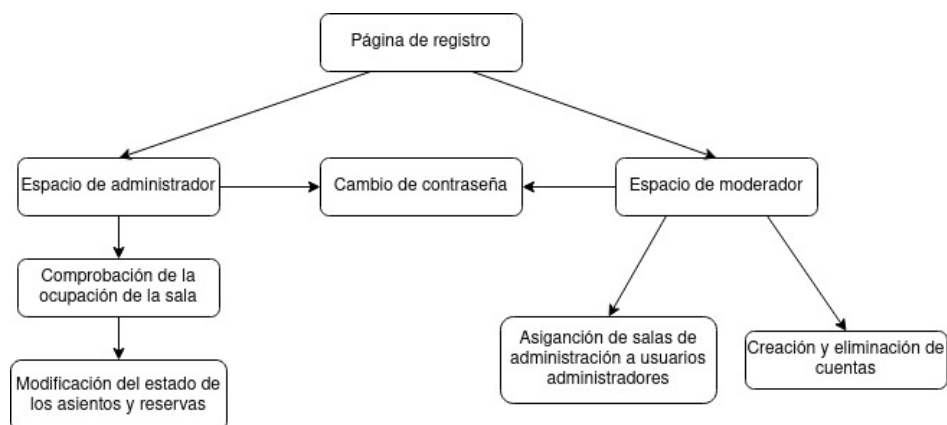


Figura 4.5: Mapa de navegación de NeoB Admin.

Capítulo 5

Implementación y desarrollo

Una vez se ha realizado la especificación de requisitos y finalizado con el diseño del sistema, se pasa a la fase de implementación. Este es el momento en el que se programa el código necesario para crear la aplicación de forma que se adapte a las decisiones de diseño que se han tomado con anterioridad y el sistema responda de la manera que se ha especificado en análisis de requisitos, cumpliendo así las expectativas puestas en él.

Durante el desarrollo del sistema, se han realizado comprobaciones periódicas al finalizar el código de cada uno de los módulos para comprobar su correcto funcionamiento.

- En el caso del *frontend*, se utiliza una herramienta que permite ver los cambios en vivo en la aplicación mientras se desarrolla llamada **Nodemon** [14], lo que facilita el trabajo a los desarrolladores evitando tener que estar recompilando y lanzando cada vez la página.
- Por otra parte, en el *backend*, para comprobar que las llamadas a cada una de las funciones devolvían los resultados esperados, se utiliza la herramienta **Postman** [15], ésta permite lanzar peticiones a un servicio y muestra la respuesta formateada para su fácil comprensión. Esto resultó de gran utilidad para ver el formato de las respuestas y posteriormente tratarlas en el *frontend* en función a su contenido.

5.1. Tecnologías utilizadas

En el estudio del estado del arte se habló sobre las tecnologías utilizadas para el sistema gestor de base de datos, la implementación del *frontend* y el *backend*. No obstante, además de estas tecnologías núcleo del sistema, se han utilizado diferentes

módulos para el desarrollo del mismo por considerarlas relevantes y de gran utilidad. Por ello, se estima oportuno enumerarlas y detallar su uso y utilidad en el sistema.

- **Material UI [16]:** es una librería de *React* de código abierto desarrollada por Google, la cual pone a disposición del desarrollador un conjunto de componentes visuales para su utilización en el sistema. Esta librería ha resultado de utilidad para incorporar elementos como cuadros de texto o desplegables, entre otros, de forma rápida, con un estilo predefinido y fácilmente adaptable al del sistema.
- **Sequelize [17]:** es un ORM de Node JS que permite traducir las tablas de una base de datos relacional en objetos para facilitar el trabajo de desarrollo. Esta herramienta resulta de gran utilidad, ya que a pesar de ofrecer la posibilidad de realizar consultas en SQL, dispone de un conjunto de operaciones predefinidas que facilitan la consulta sobre la base de datos, evitando así tener que redactar la consulta completa.
- **Nodemailer [18]:** es un módulo de código abierto de Node JS que facilita al desarrollador el envío de correos electrónicos automáticos desde el lado del servidor. Se ha seleccionado este módulo para el envío de correos, ya que su uso es sencillo y fácilmente configurable para utilizarlo con correos del dominio de unizar.
- **JSON Web token [19]:** Para la comunicación entre el *frontend* y el *backend* se utiliza un *token* de autenticación para evitar peticiones al *backend* desde terceros no deseados. Para ello se utiliza en módulo *JSON Web Tokens (JWT)* de Node JS. Los JWT son un estándar abierto que permite el intercambio de información de forma segura entre dos partes como un objeto JSON. Además de ello, también se utilizan para restringir los permisos de llamada a determinadas funciones del *backend* en función del rol del usuario que la realiza. De esta manera se consigue que solamente los administradores puedan hacer peticiones a funciones de administración.
- **Bcrypt:** este modulo de Node JS se utiliza simplemente para mantener la privacidad de los usuarios encriptando sus contraseñas y para su posterior verificación a la hora de acceder al sistema (*log in*).
- **Axios [20]:** Es una librería de Node JS que ofrece un cliente HTTP para hacer llamadas a un determinado *endpoint*.

5.2. Decisiones de diseño durante el desarrollo

En este apartado se detallan las decisiones que se han tomado durante el desarrollo del sistema debido a diferentes problemas que se han presentado a lo largo del mismo o a nuevas necesidades que han surgido durante este.

Al comienzo del desarrollo del sistema se concebía NeoB como una sola aplicación que ofrecía todas las funcionalidades que se exigen en los requisitos. Sin embargo, conforme avanzó su desarrollo se comenzaron a contemplar dos espacios claramente diferenciados, con funcionalidades muy diferentes y enfocados a dos tipos de usuarios distintos. Por ello, finalmente, se tomó la decisión de separar el *frontend* en dos aplicaciones diferentes, una enfocada a los usuarios de tipo estudiante, NeoB, en el que ellos pudiesen realizar la actividad contemplada para estos usuarios y otra enfocada a los usuarios de tipo administrador, NeoB Admin, donde se ofrecían las funcionalidades enfocadas a los administradores.

Dado que el *backend* del sistema tiene un dominio al que se realizan las peticiones, se tomo la decisión de utilizar JWTs para limitar las peticiones al *backend* a solamente los usuarios autorizados. Para ello, cuando un usuario de tipo administrador inicia sesión en la aplicación, se genera un token con validez de ocho horas para que el administrador pueda trabajar con el sistema durante su jornada sin problema, realizando todas las operaciones con el *backend* necesarias. De la misma forma ocurre con los usuarios estudiantes, cuando estos hacen una petición para comprobar el estado de una sala o para almacenar una reserva, estos poseen un token que les permite comunicarse con el *backend*. A continuación, se incluyen los diagramas de secuencias para la generación de cada uno de los tokens:

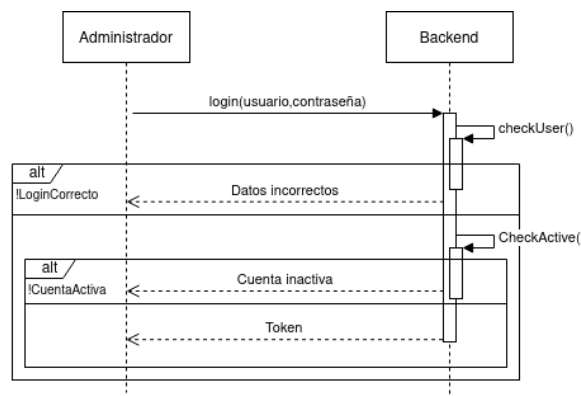


Figura 5.1: Diagrama de secuencias de generación de un token de sesión.

Como durante la mayor parte del año el aforo de la biblioteca no está completo y quedan asientos libres en la sala, se tomó la decisión de, además de poder hacer la reserva de la forma convencional a través de la página web, se pudiese reservar in-situ. Para ello, al escanear el código QR del asiento en el que un usuario se va a sentar el sistema pregunta si tienes reserva o no. En caso de no tener reserva, este comprueba si el asiento está previamente reservado por otra persona y en caso de estar libre, solicita el correo electrónico del usuario para realizar la reserva y confirmación del asiento, quedando el sitio ocupado por dicho usuario.

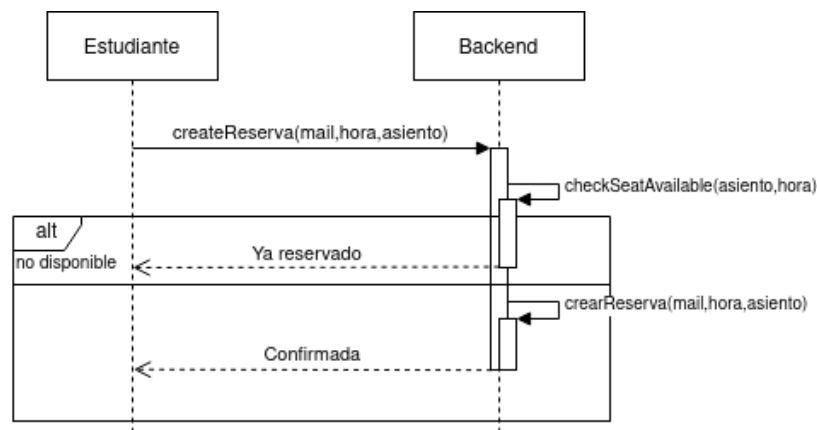


Figura 5.2: Diagrama de secuencias para la reserva in situ.

Finalmente, se tomó la decisión de implementar la creación de cuentas y asignaciones de salas a administradores temporales. Para ello, se incluye la siguiente condición a la hora de crear la cuenta: se debe indicar si el administrador es permanente o temporal, debiendo establecer una fecha de caducidad de la cuenta en este último caso.

5.3. Automatización de la creación de códigos QR

Partiendo del mapa de la sala que se quiere implementar en el sistema y de la numeración que se ha asignado a cada uno de los asientos, se deben crear los códigos QR que identifican a cada uno de ellos. Para ello, se han creado una serie de *scripts* en Python que facilitan la tarea. El código completo de los *scripts* se encuentra disponible en el Anexo E. No obstante, a continuación, se comenta brevemente:

- **generarNumeracion.py:** Es el primer *script* necesario para la creación de los códigos QR. Este, a partir de 3 parámetros de entrada, el nombre de la biblioteca, el número de columnas y el número de filas, genera un fichero de texto con la numeración de cada uno de los asientos. Cabe destacar que una vez generado el

fichero, puede que no todas las filas tengan el mismo número de asientos debido a una distribución irregular de las mesas en la sala. Por ello, esto se debe tener en cuenta a la hora de tratar el fichero. Si sucede este caso se pueden eliminar manualmente los asientos sobrantes o ser consciente de que en el fichero hay numeraciones que no corresponden con ningún asiento en la sala.

- **qr_generator.py:** Una vez se tiene la lista de asientos generada con el *script* anterior, se procede con la creación de los códigos QR que corresponden a cada uno de ellos. Para ello, el *script* recibe como primer parámetro el fichero con los asientos y a partir de la URL base de la sala, se genera la imagen con el código QR correspondiente a cada asiento, almacenando todas las imágenes en una carpeta con el nombre de la sala a la que corresponden para facilitar su posterior tratado.
- **creación_etiquetas.py:** Finalmente, tras tener todas las imágenes de los códigos QR para cada uno de los asientos, se procede a crear la plantilla de impresión. Para ello, el *script* recibe como parámetro la carpeta donde se encuentran las imágenes de los QRs y el nombre de la sala con la que se está trabajando. A partir de ahí, se utiliza una plantilla de $\text{\LaTeX}$, previamente diseñada, que se rellena con los datos de cada uno de los asientos para generar las etiquetas y el documento final con todas las etiquetas de la sala listas para su impresión.

El resultado final de las etiquetas que identifican cada uno de los asientos se puede ver en la figura 5.3.



Figura 5.3: Ejemplo de etiqueta finalizada.

5.4. Despliegue del sistema

El proceso de despliegue del sistema se ha llevado a cabo en el servidor Atila. Este servidor es un servidor de máquinas virtuales, por lo que para ello ha sido necesaria la configuración de una de ellas donde ejecutar la aplicación. Los pasos detallados de la creación, configuración y despliegue de la máquina virtual se detallan en el Anexo B.

5.4.1. Sistema operativo

El sistema operativo elegido para correr en la máquina virtual para el despliegue es Ubuntu Server 22.04 LTS [21]. Los motivos por los que se han elegido este sistema son los siguientes:

- El sistema Ubuntu Server carece de interfaz gráfica ya que está enfocado a su uso en servidores. Esta característica hace que sea un sistema ligero, que consume pocos recursos y ofrece funcionalidad completa. Además de ello, el sistema está enfocado al despliegue de aplicaciones y en su proceso de instalación ofrece la posibilidad de instalar diferentes herramientas de utilidad para el despliegue de sistemas.
- Se ha elegido la versión Long Term Support (LTS) del sistema, ya que esta ofrece actualizaciones de seguridad y mantenimiento a más largo plazo que la versión normal.
- El sistema ofrece actualizaciones de seguridad hasta abril de 2027.
- El sistema ofrece mantenimiento hasta abril de 2032.

5.4.2. Recursos asignados a la máquina virtual

Tras evaluar los recursos necesarios para que el sistema operativo elegido funcionase correctamente y tuviese la capacidad suficiente para alojar el sistema, se tomó la decisión de asignar a la máquina virtual recursos mostrados a continuación:

Recurso	Cantidad utilizada
Nº núcleos del procesador	1
Memoria RAM	2 GB
Almacenamiento interno	20 GB

Tabla 5.1: Recursos de la máquina virtual utilizada para el despliegue.

5.4.3. Software utilizado para el despliegue

Para el despliegue se ha decidido utilizar Docker [22] y Docker-compose [23].

Docker es una plataforma de código abierto que permite desarrollar y desplegar aplicaciones de forma aislada en un contenedor, sin necesidad de instalar todas las dependencias de la aplicación en el sistema donde ésta se ejecute. Docker cuenta con dos principales características las imágenes y los contenedores.

Las imágenes de Docker son un fichero, de solo lectura, con instrucciones para la creación de contenedores. Una imagen se puede crear partiendo de cero o a partir de alguna imagen existente, añadiendo las características que se consideren necesarias en cada situación. Para la creación de una imagen se utilizan los ficheros *Dockerfile*, en los que cada una de sus instrucciones forma una capa, así, en el caso de modificar alguna de ellas, solamente se cambiará dicha capa a la hora de recompilar, característica por la que es una tecnología tan ligera y rápida. Un contenedor se crea a partir de una imagen. En un contenedor se empaqueta el código y sus dependencias de forma totalmente aislada, de forma que todo se ejecuta dentro del mismo. Esto hace que la aplicación se ejecute correctamente independientemente del sistema operativo en el que se esté corriendo el contenedor o las dependencias instaladas en el mismo. Además de ello, los contenedores se caracterizan por su ligereza, ya que comparten el *kernel* del sistema operativo con el *host*, y su seguridad.

Docker Compose es una herramienta de Docker que permite definir y ejecutar aplicaciones multicontenedores. Estas aplicaciones se definen en un fichero de despliegue llamado *docker-compose.yml* y se escriben en YAML. Con la utilización de esta herramienta, se automatiza el proceso de despliegue del sistema. Además, Docker Compose cuenta con varias características por las que se ha elegido para su uso:

- Automatiza la creación de las redes de comunicación entre contenedores.
- Permite almacenar los datos de los contenedores, manteniéndolos en caso de que estos se reinicien o eliminen, mediante la utilización de volúmenes.
- Solamente vuelve a crear los contenedores que han cambiado a la hora del despliegue.

Capítulo 6

Validación experimental

Tras el diseño e implementación del sistema, una vez éste se ha desplegado y es accesible para los usuarios, se comienza con el periodo de validación experimental. Durante este periodo se pretende que, partiendo de NeoB como prototipo funcional, se pueda comprobar su viabilidad, aceptación por parte de los usuarios y trabajadores y que es una plataforma de utilidad para los mismos. En el Anexo C se incluye el trabajo realizado en colaboración con sensoriZAR en el despliegue de la red LoraWAN que dé cobertura a la Universidad.

6.1. Prueba piloto de la aplicación

Para la realización de las pruebas de validación se ha trabajado en coordinación con la dirección de la biblioteca de Hypatia de Alejandría y se ha elegido la sala de lectura como aula en la que se van a realizar las pruebas. Para ello, es necesario un mapa de la sala, con la distribución de la misma, ubicación de mesas, sillas, tomas de corriente y demás puntos de interés en la biblioteca para poder crear su gemelo digital. Una vez se dispone del mapa de la sala, se asigna una numeración a cada uno de los asientos, comenzando del fondo hasta el inicio y de izquierda a derecha, las filas se identifican con números y las columnas con letras, como se muestra a continuación.

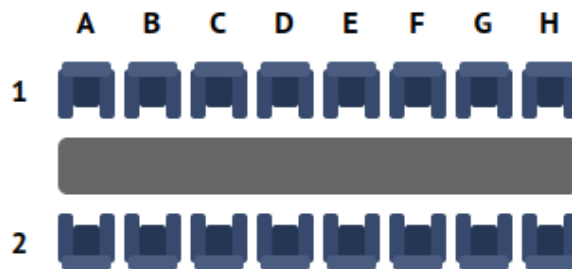


Figura 6.1: Ejemplo de numeración de asientos.

Tras enumerar todos los asientos de la sala en el gemelo digital, se debe hacer lo

mismo en la biblioteca física. Para ello, se utilizan códigos QR encargados de vincular el asiento físico con el que corresponde en el gemelo digital.

6.1.1. Pegado de códigos

Tras generar las etiquetas de cada una de las plazas de la biblioteca, se procede con el pegado de las mismas, para ello se toman una serie de decisiones de pegado. Estas se detallan en unas pequeñas instrucciones que se entregan junto a las etiquetas para su correcto pegado.

PASO 1: Cada etiqueta consta del código QR correspondiente al asiento, el número de asiento y la sala a la que corresponde. Consultar el mapa de la sala de estudio y colocar los códigos QR en la disposición de las mismas. Las salas se encuentran numeradas de forma incremental empezando por el fondo de la misma, es decir, el asiento de más al fondo a la izquierda será el 1A. Las etiquetas deben ser colocadas en la parte inferior izquierda de cada plaza tal cual se indica en la figura 6.2, su colocación debe ser en vertical.

PASO 2: Siguiendo este mapa y atendiendo al código de asiento se procede a pegar todas las etiquetas. (< Número indicando la fila> <Letra indicando la columna>) No se debe tener en cuenta el color/estado de los asientos, todas las etiquetas deberán quedar fijadas en la posición del asiento correspondiente. Tampoco se deberá tener en cuenta que las sillas no estén dispuestas en la sala físicamente.

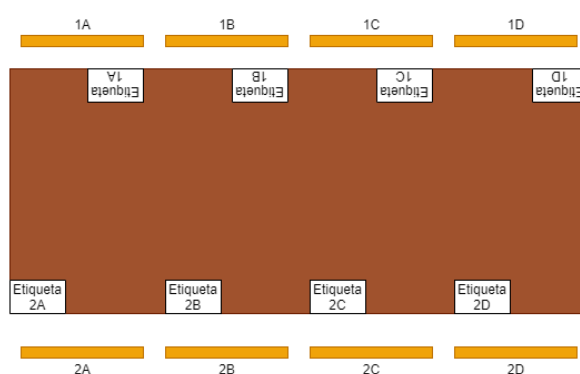


Figura 6.2: Croquis de disposición de pegado de etiquetas.

6.1.2. Formación de los bibliotecarios

Finalmente, antes de comenzar con las pruebas de validación del sistema se realiza una formación al personal de la biblioteca, para que conozcan el sistema, su funcionamiento y todas sus capacidades.

En la formación se comienza presentando el sistema. Se habla sobre sus objetivos y como se va a utilizar e implantar en la sala. Tras esto se presentan ambos sistemas, NeoB, el sistema enfocado a usuarios y NeoB Admin, el enfocado al personal de administración y se realiza una demostración de todas las funcionalidades del mismo.

Con esta formación se pretende introducir al personal en el uso de NeoB, para que el día en el que comiencen las pruebas ellos tengan los conocimientos necesarios para poder prestar ayuda a los estudiantes que lo necesiten en caso de que sea necesario. Además, también se pretende que sepan como funciona el espacio de administradores y trabajar con él para dar solución a posibles incidencias con las reservas.

6.1.3. Prueba piloto de uso del sistema

El día 15 de Noviembre de 2022, se comenzó con la prueba piloto del sistema en la sala de lectura de la biblioteca de Hypatia de Alejandría. Para ello, antes de la apertura de la biblioteca se procedió a la creación de la cuenta de los administradores de la sala y a una breve explicación de las funcionalidades del sistema para refrescar los conocimientos impartidos en la formación del personal.

Una vez abierta la biblioteca, se comenzó a explicar a los usuarios que no conocían el sistema de que se trataba y cuales eran sus objetivos y que a partir de ahora deberían de usarlo para poder mantener un registro del uso de la biblioteca.

Con el uso del sistema se encontraron ciertos detalles que no funcionaban correctamente y otros que se decidieron cambiar para facilitar su uso.

Inicialmente, en el sistema se ofrecían franjas de estudio de una hora, obligando al estudiante a realizar tantas reservas como horas quisiera estudiar. Esto se consideró muy incomodo, por ello, tras evaluar las posibles franjas que mejor se adecuaban al estudio de la mayoría, se tomó la decisión de sustituir las franjas de estudio de una hora por dos turnos de estudio en todo el día, horario de mañana, de 8:15 a 14:00 y horario de tarde, de 14:00 a 20:30. Además, al modificar las franjas de estudio y reducirlas a solamente dos, se modificó ligeramente el proceso de finalización del estudio, ya que al ser franjas tan amplias, cada usuario debe indicar cuando libera su sitio para que éste pueda ser utilizado por otra persona. Para ello, a la hora de realizar la reserva, ya sea in situ o a través de la aplicación web, el sistema envía de forma automática un correo electrónico a través del cual se puede finalizar el estudio

y liberar la plaza al salir.



Figura 6.3: Modificación de las franjas de estudio

También se modificó ligeramente la interfaz gráfica de la aplicación de NeoB, enfocada a los estudiantes, con el propósito de eliminar elementos obsoletos y facilitar la navegación del usuario por el sistema. Para ello, en la pantalla principal se eliminó el botón de “Protocolo COVID” ya que esta funcionalidad pese a estar desarrollada, actualmente no se considera útil en el sistema. Al eliminar el botón de la vista de los usuarios y trasladarlo a un lugar más secundario, menos visible, se evitan confusiones por parte de los estudiantes y se facilita la navegación por el sistema.

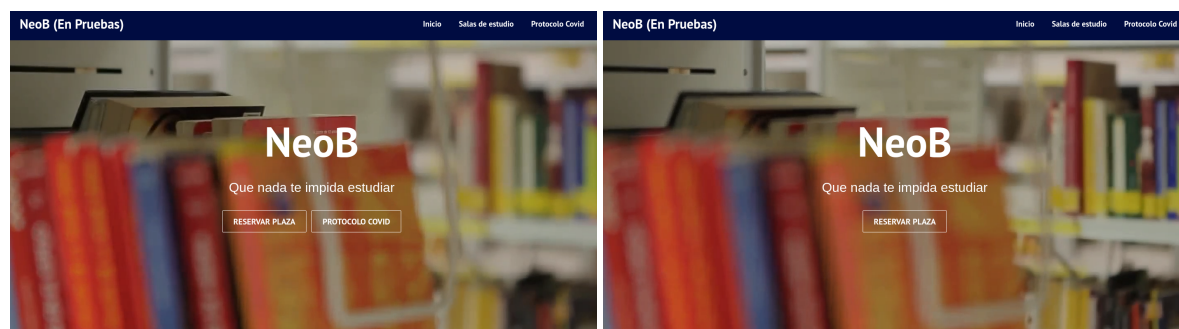


Figura 6.4: Cambio de la interfaz gráfica de la aplicación eliminando el botón de “Protocolo COVID”

Además, como último cambio en la interfaz gráfica de la aplicación, se eliminó la página de salas, a la que se podía acceder desde los botones de “Reservar Plaza” y “Salas de estudio”, en la cual se mostraban todas las salas implementadas en el sistema. En su lugar, dado que solamente se está trabajando con la sala de lectura de la biblioteca de Hypatia de Alejandria, al hacer *click* sobre cualquiera de los dos botones previamente mencionados, da comienzo el proceso de reserva de una plaza en la biblioteca.

6.1.4. Evaluación de la aceptación del sistema

Para evaluar la aceptación y el uso del sistema por parte de los estudiantes, durante los primeros días de la prueba piloto, se han estado realizando diferentes recuentos de gente presente en la sala y gente registrada en el sistema a diferentes horas, tanto en el turno de mañana como en el de tarde. Partiendo de estos recuentos realizados se ha creado el siguiente gráfico en el que se muestran los resultados.

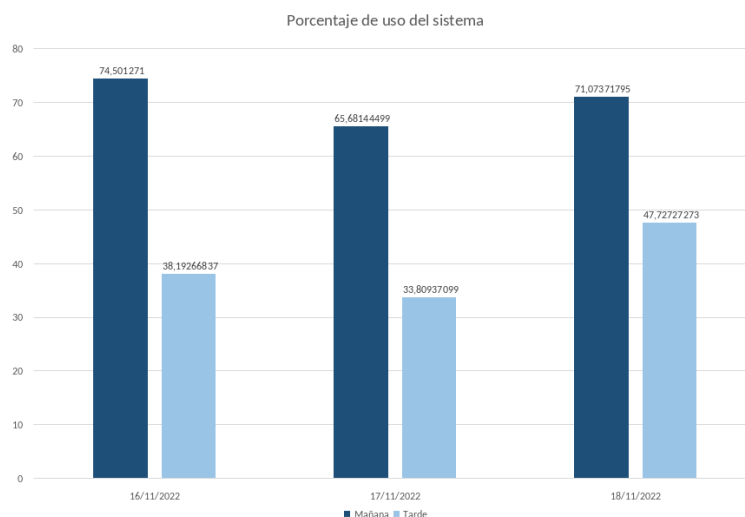


Figura 6.5: En el gráfico se muestra el porcentaje medio de uso del sistema.

Observando el gráfico se pueden sacar varias conclusiones. La primera y más clara de ellas es que el porcentaje de aceptación por los usuarios que utilizan la biblioteca en el turno de mañanas es mucho mayor que el de los usuarios de tardes, más concretamente, el porcentaje medio de uso de todas las muestras recogidas en el turno de mañana es del 70,41 %. Sin embargo, el del turno de tarde, es de 39,91 %, siendo un 30,5 % inferior la utilización del sistema por parte de los estudiantes que asisten por la tarde.

Además, se puede apreciar como el uso de la aplicación durante los días medidos se ha mantenido más o menos constante, viendo un porcentaje de aceptación bastante aceptable en el turno de mañanas y un poco más mediocre en el de tardes.

Por último, observando los datos de la tabla 6.1 donde aparecen todas las muestras tomadas, se puede observar como el porcentaje de utilización de la sala en la muestra del día 18/11/2022 a las 12 es del 102,56 %. Esto es debido a que los usuarios, a medida que va avanzando la mañana, realizan sus reservas de asientos para el estudio. Sin embargo, a la hora de salir, no finalizan la reserva, en consecuencia en el sistema sigue apareciendo como que ese asiento está ocupado, dando lugar a casos como este.

De estos datos se concluye que, dado que la labor de formación del personal ha sido mayor con el equipo del turno de mañana, la aceptación por parte de los usuarios que estudian en dicho turno es superior, ya que el personal está más preparado para la consulta de dudas y solución de problemas, por lo que se debe realizar la misma labor con el personal de tardes para incrementar el uso del sistema en ese turno. Así mismo, se observa que la gente hace uso de la aplicación registrando su llegada a la biblioteca; sin embargo, no registra su salida de ella, lo que puede dar lugar datos incorrectos entre el sistema y la realidad. Esto puede ser debido a un enfoque incorrecto de la campaña publicitaria.

Día	Hora	Ocupación Real	Ocupación en NeoB	% Uso
16/11/2022	9:50:00	52	35	67,31
16/11/2022	11:14:00	64	47	73,44
16/11/2022	12:30:00	58	48	82,76
16/11/2022	16:30:00	69	27	39,13
16/11/2022	19:40:00	102	38	37,25
17/11/2022	10:00:00	58	32	55,17
17/11/2022	13:00:00	63	48	76,19
17/11/2022	16:30:00	41	11	26,83
17/11/2022	19:30:00	76	31	40,79
18/11/2022	10:00:00	48	19	39,58
18/11/2022	12:00:00	39	40	102,56
18/11/2022	16:00:00	22	10	45,45
18/11/2022	19:45:00	38	19	50

Tabla 6.1: Muestra de ocupación tomada de forma presencial en la sala durante los primeros días de prueba.

Encuestas de satisfacción

Para conocer la opinión de los usuarios sobre la aplicación de forma más detallada se han realizado dos encuestas que se han enviado a todos los estudiantes y personal de la biblioteca que han hecho uso de cada uno de los espacios de NeoB durante la prueba piloto. Ambas encuestas cuentan con 6 sencillas preguntas las cuales permiten obtener una visión general de como ven los usuarios el sistema. En el Anexo D se incluyen los resultados obtenidos para cada una de las preguntas.

Encuesta de evaluación para los estudiantes

Observando las respuestas, se puede ver como la mayoría de los estudiantes han llegado a conocer el sistema al llegar a la biblioteca, solamente una mínima parte

conocía de su existencia antes de llegar a la sala de lectura ya fuese por que lo habían visto en redes sociales o porque les habían hablado sobre NeoB.

Por otra parte, a la pregunta de que si les ha parecido NeoB un sistema sencillo de utilizar, en la que se evalúa desde 1, muy difícil de utilizar a 5, muy sencillo de utilizar, la gran mayoría de los usuarios han contestado a la pregunta con un valor superior a 3, por lo que se asume que las labores de diseño en las que se priorizaba la navegación con pocos *clicks*, un sistema intuitivo y *responsive*, adaptable a cualquier pantalla y orientación, se han realizado con éxito, dando lugar una aplicación agradable para el uso de los estudiantes.

De la misma forma que la pregunta anterior se ha pedido a los estudiantes que evalúen la utilidad del servicio que ofrece NeoB. En este caso, los resultados obtenidos siguen siendo favorables, la mayoría de los usuarios han puntuado la utilidad del sistema por encima de 3; sin embargo, hay un pequeño grupo de personas, que no consideran que el servicio ofrecido sea útil.

Finalmente, se pregunta si el usuarios había tenido problemas en la biblioteca con los sitios antes de NeoB y en caso de haberlos tenido, si considera que el sistema puede contribuir en resolverlos. En este caso, dos tercios de los usuarios (66,7%) no han tenido nunca problemas con el sitio en las bibliotecas; sin embargo, el otro 33,3% de los usuarios sí los habían tenido. El principal problema presente entre los afectados era la falta de sitio en la sala cuando llegaban a estudiar, por lo que ellos consideran que NeoB les puede solucionar el problema al facilitar la reserva del sitio desde casa.

Encuesta de evaluación para el personal de la biblioteca

Con la realización de esta encuesta se pretende conocer la satisfacción y opinión del personal de la biblioteca sobre la utilidad e implantación del sistema.

Observando las respuestas, se puede ver como los administradores han evaluado la utilidad del servicio ofrecido por el sistema por encima de 3, por lo que de ello se concluye que el sistema resulta útil entre el personal para las labores que esta concebido, de gestión y control de aforo.

Por otra parte, en cuanto a la facilidad de uso, ocurre como en la pregunta anterior, las respuestas son favorables, por lo que se concluye que la labor de diseño del

sistema se llevó a cabo con éxito. La fácil adaptación del personal al uso del sistema también se debe a la formación recibida en el periodo previo a su despliegue en la que se introdujo al personal al sistema y se le explicaron todas sus funcionalidades. Por ello, la utilidad de la formación recibida ha sido evaluada con una puntuación de 4 sobre 5.

Finalmente, se ha querido conocer la experiencia del personal con los conflictos en la sala y su opinión sobre si creen que el sistema podría solucionarlos. Con las respuestas obtenidas se puede ver que la sala de lectura de la biblioteca de Hypatia de Alejandría no es una sala en la que se generen demasiados conflictos, ya que la mayoría del personal no ha encontrado nunca ninguno; sin embargo, los administradores que sí se han encontrado con ellos consideran que sí que podría servir de ayuda para resolverlos.

Datos sobre la utilización del sistema

Tras la primera semana de prueba piloto del sistema se han obtenido los siguientes datos sobre la utilización del sistema.

- Un total de 267 estudiantes han hecho uso del sistema para reservar plaza ya sea insitu o a través de la aplicación web.
- En el sistema se han registrado un total de 421 reservas realizadas. Quedando distribuidas como se muestra a continuación.

Fecha	Nº Reservas
16/11/2022	119
15/11/2022	117
17/11/2022	95
18/11/2022	86
21/11/2022	1
14/11/2022	1
22/11/2022	1
11/11/2022	1

Tabla 6.2: Total de reservas realizadas cada día.

- De las 421 reservas realizadas durante este periodo, se puede notar una clara preferencia por parte de los usuarios a realizar las reservas insitu sobre reservar la plaza que quieren con anterioridad. Mas concretamente, 412 de las 421 reservas totales realizadas durante este periodo han sido realizadas insitu, siendo 9 las realizadas con anterioridad.

6.1.5. Análisis de los datos para el apoyo a la toma de decisiones.

Como bien se ha comentado con anterioridad, el objetivo del sistema no es solamente garantizar una plaza en la biblioteca a los estudiantes y permitir la consulta de la ocupación en tiempo real de la misma. Otro de los objetivos de NeoB es proveer de datos precisos sobre el uso de la sala gracias a la información que se obtiene a partir de las reservas realizadas a través del sistema. Partiendo de estos datos como fuente en crudo de información, se pretenden organizar de tal manera que pueda ayudar a la hora de creación de informes sobre el uso de la sala, ocupación de la misma o como apoyo para la toma de decisiones.

Para ello, se han definido una serie de plantillas para mostrar los datos de forma ordenada y comprensible, de forma que sean claros y fáciles de entender y puedan resultar de utilidad en el apoyo a la toma de decisiones.

Pese a que todas las plantillas acaban aportando información sobre el uso de la sala, se ha querido diferenciarlas en dos grandes grupos, las que aportan información sobre la ocupación de la sala y las que aportan información sobre el uso que hacen los usuarios de ella.

Ocupación de la sala

Las plantillas definidas a continuación, aportan información de valor a la hora de comprobar la ocupación de la sala en términos relativos a gente en la misma.

Comparación del uso de la sala en las diferentes franjas

Para ello se ha diseñado una plantilla simple en la que se pueda diferenciar fácilmente las diferentes franjas horarias. Esta plantilla se representa con un gráfico circular y los datos que se pueden extraer de ella son valiosos para determinar las horas punta de estudio en la sala.



Figura 6.6: Porcentaje de reservas sobre el total en cada una de las franjas.

Del gráfico anterior se puede observar como la mayoría de las reservas se han realizado por la mañana, lo que permite deducir que la ocupación de la sala es mayor en esta franja. Cabe destacar que, en este caso, al solamente haber dos franjas de estudio tan amplias, es ligeramente más difícil determinar la hora punta de estudiantes en la sala con exactitud; sin embargo, como se ha comentado con anterioridad, las franjas son fácilmente modificables y adaptables a los horarios de cada sala y periodos de estudio, por lo que a partir de estos datos pueden servir de utilidad para determinar las franjas con mayor exactitud.

Ocupación diaria de la sala

Con esta plantilla se pretende representar el uso de la sala durante cada uno de los días de la semana. Para ello, se representa en el eje vertical del gráfico la ocupación de la sala (en porcentaje) y en el eje horizontal cada uno de los días de la semana que la sala permanece abierta.



Figura 6.7: Porcentaje de reservas diario sobre el total semanal.

Dado que la fase de validación experimental comenzó el día 15/11/2022, solamente

se dispone de datos a partir de ese día. Pese a ello ya se pueden extraer ciertas conclusiones sobre el uso semanal de los usuarios en la sala. En el gráfico se puede observar como el porcentaje de uso es mucho mayor en los días de martes y miércoles frente al resto. Esta información puede resultar útil para para la posterior toma de decisiones como, por ejemplo, aumentar la ventilación esos dos días para garantizar un ambiente adecuado de estudio cuando la sala esta muy llena, entre otras.

Ocupación semanal

De la misma forma que en la plantilla anterior, se representa la ocupación de la sala por semanas. La dimensión de esta plantilla puede ser del número de semanas que se quiera; sin embargo, se recomienda representar la ocupación por semanas mensualmente, para ver la evolución de la ocupación de la sala durante el mes.

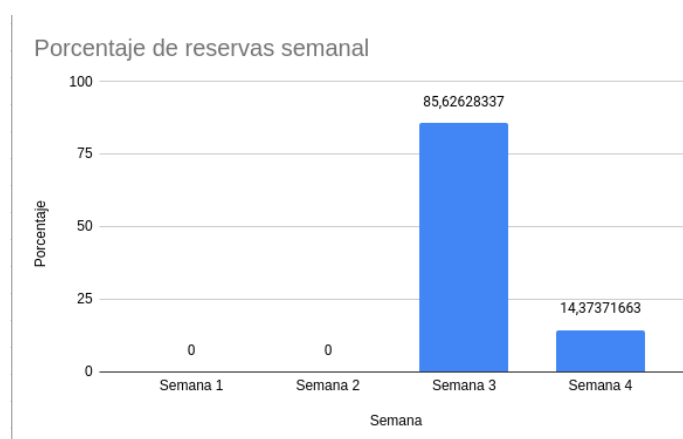


Figura 6.8: Porcentaje de reservas semanal sobre el total mensual.

La figura 6.8 representa la información que se quiere aportar con esta plantilla. Dado que los datos recopilados solamente cubren dos de las cuatro semanas del mes de noviembre, las dos primeras quedan sin información. Sin embargo, la conclusión que se puede extraer de este gráfico es la evolución de la ocupación a lo largo del mes.

Ocupación mensual

Al igual que en la ocupación semanal, se pretende representar la ocupación total de la sala para cada uno de los meses que se incluyan en la plantilla. Para que los datos de la plantilla sean de valor se recomienda rellenarla con datos anuales, lo que permitiría evaluar los meses de mas uso de la biblioteca y sacar conclusiones a partir de ello.

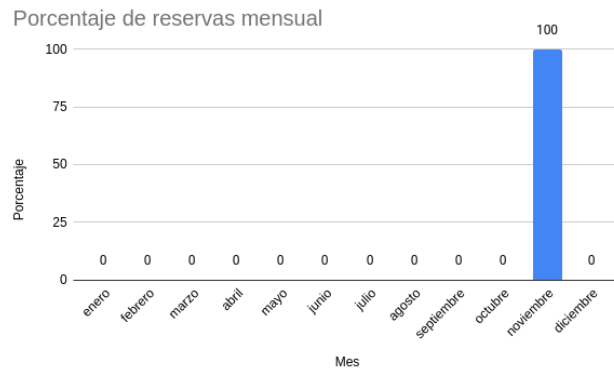


Figura 6.9: Porcentaje de reservas mensual sobre el total anual.

De forma análoga al caso de la plantilla anterior, la recopilación de datos solamente ha extraído información del mes de noviembre. Por ello, se puede observar en el gráfico que la información no es válida; sin embargo, una vez se disponga de datos anuales completos, con esta plantilla se podrá evaluar la ocupación de la sala a lo largo del año, estimando así los momentos de más afluencia de gente y pudiendo tomar decisiones en torno a ellos.

Uso de la sala

Estas plantillas aportan información sobre el comportamiento de los usuarios en la sala, es decir, con ellas se puede evaluar cuales son las zonas más utilizadas por los estudiantes y tomar decisiones en torno a ellas.

Asientos más utilizados

El objetivo de esta plantilla es mostrar cuales son los asientos más utilizados en la biblioteca. Para ello se muestra en el eje vertical en número de reservas realizadas a un asiento y en el horizontal, cada uno de los asientos.



Figura 6.10: Asientos más utilizados en la sala.

Observando los datos mostrados por la gráfica obtenida, se puede concluir que los asientos con acceso a toma de corriente, los asientos de las filas 1,2 y 21 y 22 además de los que se encuentran ubicados en la pared derecha de la sala, es decir, asientos asignados a la letra X, son los asientos que mayor número de reservas reciben. Así mismo, también se puede comprobar como los puestos denominados con la letra A son bastante frecuentes, ya que se encuentran al lado de la ventana. Estos datos pueden servir de utilidad para la toma de decisiones a la hora de la electrificación de la sala o de su ventilación.

Capítulo 7

Análisis de riesgos y plan de mantenimiento del sistema

El análisis de riesgos es el estudio de los posibles riesgos y amenazas a los que está expuesto el sistema durante su vida útil. Para la el análisis de riesgos se utiliza la Metodología de Análisis y Gestión de Riesgos orientada a Sistemas de información y Comunicación (MAGERIT). MAGERIT permite analizar y gestionar de forma sistemática, objetiva y contrastable los riesgos a los que está expuesta una organización. Los objetivos principales de esta metodología son concienciar de la existencia de riesgos y de la necesidad de gestionarlos, definir un proceso sistemático para estimar la magnitud de los riesgos a que está expuesta una organización y describir y planificar el tratamiento de los riesgos para mantenerlos bajo control.

7.1. Plan de análisis y gestión de riesgos

En el plan de análisis y gestión de riesgos realizado se comienza por la designación de los activos más importantes para la entidad; tras ello, se procede con la identificación de las amenazas a las que se encuentran expuestas los activos previamente mencionados y el impacto que puede tener en caso de materializarse una de ellas y, finalmente, se proponen medidas de salvaguarda para tratar de minimizar las posibilidades de que estas se materialicen, así como un plan de contingencia que llevar a cabo en caso de que se acabe materializando [24].

Activos

Un activo se define como un elemento del sistema de información que soporta la misión de organización. Existen dos tipos de activos, los esenciales, tales como información y servicios y los secundarios, como datos, equipos informáticos, redes de comunicación, etc. A continuación, se realiza un análisis de los activos más relevantes

del sistema.

Id	Nombre	Descripcion	Tipo	Ubicación	Crítico
S_At	Servidor Atila	Servidor donde se encuentra el sistema desplegado	Servidor físico	CPD Edificio Betancourt	Sí
R_Btn	Red	Infraestructura de red del CPD	Infraestructura de red	CPD Edificio Betancourt	Sí
O_Adm	Ordenadores	Ordenadores de los administradores	Hardware	Recepcion de la sala	No
O_Dev	Ordenador	Ordenador de desarrollo del sistema	Hardware		No
Dat	Datos	Datos almacenados del uso del sistema	Datos	CPD Edificio Betancourt	Sí

Tabla 7.1: Activos de la entidad.

Amenazas

Una amenaza se define como un evento que le puede ocurrir a cualquiera de los activos causando perjuicio a la organización. En este apartado se enumeran las amenazas más relevantes a las que se encuentran expuestos los activos. A la hora de identificar las amenazas se utiliza el catálogo de amenazas que incluye la metodología MAGERIT v3 [25].

- **N.1-I.1:** Incendio en el centro de procesamiento de datos.
- **N.2-I.2:** Inundación en el centro de procesamiento de datos.
- **I.7:** Condiciones inadecuadas de temperatura o humedad.
- **I.5:** Avería de carácter físico o lógico del servidor donde se aloja el sistema.
- **E.1:** Errores de los usuarios en el uso del software.
- **E.2:** Errores del administrador.
- **E.24:** Caída del sistema por agotamiento de recursos.

Evaluación de los riesgos

Una vez llegados a este punto se dispone de los activos más importantes y de las amenazas a las que estos están expuestos. Disponiendo de esta información se puede proceder al cálculo del riesgo. Para ello, para par <activo, amenaza> se calcula

la probabilidad de que ésta se materialice y el impacto que tendría sobre los activos. Para finalizar se calcula el riesgo siendo este igual al impacto por la probabilidad de que la amenaza se materialice. El cálculo de riesgo se puede realizar usando tanto criterios cuantitativos como cualitativos.

Cálculo de la probabilidad de materialización

Cualitativo	Cuantitativo	Descripción
Bajo	1	La amenaza se materializa como mucho una vez al año.
Medio	2	La amenaza se materializa como mucho una vez al mes.
Alta	3	La amenaza se materializa como mucho una vez a la semana.

Tabla 7.2: Tabla para el cálculo de la probabilidad de que una amenaza se materialice.

Cálculo de la probabilidad del impacto

Cualitativo	Cuantitativo	Descripción
Bajo	1	El daño derivado de la materialización de la amenaza no tiene consecuencias relevantes para la organización.
Medio	2	El daño derivado de la materialización de la amenaza tiene consecuencias reseñables para la organización.
Alta	3	El daño derivado de la materialización de la amenaza tiene consecuencias graves para la organización.

Tabla 7.3: Tabla para el cálculo del impacto de una amenaza.

Para el cálculo del riesgo depende del método de evaluación utilizado, cualitativo o cuantitativo, se deberá de seguir un método u otro. En caso de haber evaluado la probabilidad y el impacto con valores cuantitativos, se deberá calcular el riesgo como se ha comentado previamente

$$RIESGO = IMPACTO * PROBABILIDAD$$

y en caso de haberse evaluado con valores cualitativos el riesgo se rige por la siguiente tabla.

		Impacto		
		Bajo	Medio	Alto
Probabilidad	Bajo	Muy bajo	Bajo	Medio
	Medio	Bajo	Medio	Alto
	Alto	Medio	Alto	Muy alto

Tabla 7.4: Tabla para el cálculo del riesgo.

En base a los criterios de evaluación establecidos en las tablas superiores, el riesgo de cada una de las amenazas descritas se muestra en la siguiente tabla.

Amenaza	Probabilidad	Impacto	Riesgo
N.1/I.1	Baja	Alto	Medio
N.2/I.2	Baja	Medio	Bajo
I.7	Baja	Medio	Bajo
I.5	Media	Medio	Medio
E.1	Baja	Medio	Bajo
E.2	Baja	Alto	Medio
E.24	Baja	Alto	Medio

Tabla 7.5: Calculo del riesgo que supone cada amenaza.

Tratamiento del riesgo

Una vez definido el riesgo que supone cada una de las amenazas, se deberán tratar aquellas que se superen un umbral de riesgo definido. En este caso, al haber utilizado la evaluación de riesgos cualitativa, se han decidido tratar las amenazas que superasen el nivel de riesgo “Medio”.

- **N.1/I.1:** Dado que la probabilidad de que se suceda un incendio en la sala de servidores es muy baja, se ha decidido asumir el riesgo, ya que el coste de realizar una protección ignífuga es inasumible.
- **I.5:** La medida que se ha decidido adoptar para evitar que esta amenaza se materialice es el establecimiento de vigilancia en el sistema para poder detectar con la mayor rapidez posible cualquier avería y actuar en consecuencia a ella.
- **E.2:** Para la prevención de este riesgo se ha diseñado el software para tratar de eliminar todos los errores posibles del usuario para que haga lo que haga nunca afecte a la disponibilidad del sistema.
- **E.24:** Para evitar que esta amenaza se materialice, se ha definido un plan de mantenimiento del sistema en el que se establecen los pasos a seguir en casos como que se llene la cuota de disco o el agotamiento de recursos.

7.2. Plan de mantenimiento y seguridad

Con el desarrollo de un plan de mantenimiento se pretenden reducir las averías e incidencias imprevistas en el sistema definiendo una serie de tareas de mantenimiento y cuidados de las infraestructura que minimicen dichas incidencias. Pretendiendo

reducir al máximo el mantenimiento reactivo, es decir, reparar el problema una vez a surgido y tratando de realizar un mantenimiento constante.

Objetivos del mantenimiento

Los objetivos que se han establecido con el presente plan de mantenimiento son los siguientes:

- Ofrecer una disponibilidad del sistema del 99 %.
- Minimizar los fallos que se encuentra el usuario final durante su uso.
- Fácil mantenimiento.
- Garantizar los recursos computacionales suficientes para el correcto funcionamiento del sistema.

Elementos de mantenimiento

En esta sección se detallan los elementos a revisar y mantener en buenas condiciones para el correcto funcionamiento del sistema.

- Estado del servidor.
- Estado de la sala de servidores, ventilación, limpieza, temperatura, etc.
- Versiones del sistema.
- Versiones del sistema operativo del servidor.

Prevención de pérdida de datos

Para la prevención de pérdida de los datos almacenados por la aplicación se planifica la creación de copias de seguridad de la base de datos de forma periódica. Ya que la base de datos almacena los datos de uso de la sala diariamente, se considera oportuno hacer *backups* incrementales diarios para así garantizar que en caso de error la pérdida de datos es mínima. Dichas copias de seguridad, se deben almacenar en un servidor distinto a la máquina donde se encuentra la aplicación.

Mantenimiento reactivo

Cuando se detecte que el sistema ha dejado de funcionar, o funciona de forma anómala se deberá tratar el problema con la mayor rapidez posible, para restaurar el servicio cuanto antes.

En caso de que el sistema se haya caído y haya dejado de ofrecer el servicio, se revisarán los registros de cada uno de los contenedores de la aplicación, para tratar de determinar donde se ha producido el error y tratar de actuar en consecuencia a ello. Una vez se haya detectado el fallo, se documenta el fallo encontrado y como se ha tratado de solucionar para facilitar su solución en caso de que volviese a aparecer y se reinicia el servicio manualmente.

Licencia

El código se ha distribuido bajo la licencia GNU GPLv3. Esta es una licencia de código abierto y de libre distribución. A continuación, se enumeran las principales características de la misma.

- La licencia permite que el código publicado pueda ser utilizado con fines comerciales.
- El material bajo esta licencia es de libre distribución.
- Al tratarse de una licencia de código abierto este puede ser modificado por los contribuyentes.
- La licencia permite la modificación y uso del material bajo la misma para el uso privado.

Además de las características que se acaban de mencionar, para poder hacer uso del código publicado bajo esa licencia hay que cumplir una serie de condiciones.

- El código debe estar disponible cuando el material es distribuido.
- Las modificaciones del código deben ser publicadas bajo la misma licencia.
- Los cambios realizados al material deben estar correctamente documentados.
- Se debe incluir una copia de la licencia y de los derechos de autor.

En la siguiente tabla se muestran resumidas las principales características, Condiciones y Limitaciones de la licencia.

Permisos	Condiciones	Limitaciones
Uso comercial	Código libre	Responsabilidad
Distribución	Avisos de licencia y copyright	Garantía
Modificación	El uso en la red es distribución	
Uso de patente	Misma licencia	
Uso privado	Documentación de las modificaciones de código	

Tabla 7.6: Principales características de la licencia.

Requisitos hardware del sistema

En este apartado, se detallan los requisitos hardware necesarios para el alojamiento y correcto funcionamiento del sistema durante distintos periodos de tiempo.

Para el correcto funcionamiento del mismo no es necesario una gran potencia computacional, por lo que sería suficiente con un procesador de un núcleo y 2GB de memoria RAM, ya que el consumo del sistema operativo del servidor es mínimo debido a que carece de interfaz gráfica. Sin embargo, el espacio de disco se debe estimar en función del crecimiento estimado de la base de datos. Sabiendo esto y contando con el esquema relacional de la base, se puede aproximar el tamaño de la misma para garantizar que el sistema no se queda sin espacio de almacenamiento, lo que causaría que este dejase de funcionar y la pérdida de recogida de datos hasta su solución.

La tabla en la que se almacenan la mayoría de los datos de la actividad del sistema es la tabla “reserva”, ya que en ella se almacenan todas las reservas realizadas y sus modificaciones. Por ello, junto con los datos de números de usuarios recogidos durante la prueba piloto, se va a estimar el espacio de disco necesario para garantizar el correcto funcionamiento del sistema durante diferentes periodos de tiempo.

Los datos obtenidos sobre la utilización del sistema se pueden extraer de la base de datos con las siguientes consultas.

Para conocer el número de reservas totales:

```
SELECT COUNT(IDRESERVA) FROM RESERVA;
```

```
neob=# SELECT COUNT(IDRESERVA) FROM RESERVA;
count
-----
    422
(1 row)
```

Figura 7.1: Numero total de reservas realizadas durante el periodo de prueba piloto.

Además, utilizando la consulta

```
SELECT PG_SIZE_PRETTY( PG_DATABASE_SIZE('NEOB')) AS "DATABASE SIZE";
```

se obtiene el tamaño de la base de datos. Por ello se ha calculado el tamaño de la base de datos vacía, es decir, sin reservas realizadas, solamente con los datos con los que se inicializa la misma, siendo de $9137kB$ y el de la base de datos actual, con un total de 422 reservas realizadas, siendo de $9345kB$. Dado que la única diferencia presente entre ambas bases de datos son las reservas realizadas, se puede calcular de forma aproximada conste en espacio de una reserva

$$\frac{9345 - 9137}{422} = 0,49$$

, aproximadamente medio kilobyte.

Una vez conocido el espacio que ocupa una reserva, se proceda a estimar la cantidad de usuarios diarios que hacen uso de la aplicación. Para ello se van a hacer uso de los datos obtenidos durante la valoración experimental, sobre la aceptación de los usuarios del sistema. De ellos se extrae que el 55,6 % de los usuarios presentes en la biblioteca hacen uso del sistema.

Dado que las 422 reservas totales que hay registradas en el sistema son en un periodo de 4 días, se estiman aproximadamente unas 106 reservas diarias si la utilización del sistema se mantuviese en un 55,6 %. Sin embargo, se espera que esta se incremente conforme el sistema se vaya utilizando y los usuarios vean su utilidad. Por ello, se ha decidido redondear a unas 200 reservas diarias, dando lugar a un total de $200 * 0,5 = 100$ kB de espacio diario.

Tiempo	Espacio necesario para la base de datos
1 mes	3MB
6 meses	18MB
1 año	36MB
5 años	180MB

Tabla 7.7: Espacio estimado ocupado por la tabla reserva.

Por ello, teniendo en cuenta el espacio requerido para la instalación del sistema, el despliegue de los contenedores y el crecimiento de la base de datos estimado, se concluye

que con 15 Gb. de espacio de almacenamiento es suficiente para que el sistema funcione correctamente, sin falta de almacenamiento, durante al menos 5 años.

Capítulo 8

Conclusiones y líneas futuras de trabajo

En este apartado final, se realiza una conclusión del trabajo realizado a lo largo de estos meses, comprobando que se han alcanzado todos los objetivos propuestos para el desarrollo de este Trabajo Final de Grado (TFG). También se detallan las futuras líneas de trabajo posibles en el sistema en caso de continuar con su desarrollo.

8.1. Conclusiones

El objetivo principal del proyecto es desarrollar e implantar un sistema de reservas y control de aforo en aulas comunes, basándose en el sistema NeoB, e integrar dicho sistema en el ecosistema sensoriZAR. Para ello, se ha tratado de generalizar y adaptar NeoB para que pueda emplearse en cualquier espacio de forma sencilla, integrar el sistema desarrollado en el ecosistema sensoriZAR desplegando una red LoraWAN que permita la recogida de datos procedentes de sensores de ocupación y, finalmente, la implantación del sistema y análisis, y la evaluación de dicha implantación en la Biblioteca Hypatia de Alejandría.

Teniendo en cuenta el trabajo realizado durante estos meses y los resultados obtenidos, se determina como finalizado el trabajo propuesto para el marco de este proyecto, habiendo conseguido alcanzar el objetivo principal del proyecto, desarrollar un sistema que permite a los estudiantes comprobar la ocupación de la sala en tiempo real, los asientos ocupados, disponibles y no disponibles en la biblioteca, el aforo de la misma y realizar una reserva de una plaza de estudio. Además, incluye funcionalidades adicionales como, por ejemplo, la gestión de los asientos de la sala y la cancelación y modificación de reservas, enfocadas a los administradores.

Por otra parte, para completar el objetivo propuesto de integrar el sistema

desarrollado en el ecosistema sensoriZAR desplegando una red LoraWAN que permita la recogida de datos procedentes de sensores de ocupación, se considera que el trabajo realizado con sensoriZAR contribuye de una manera directa en la ampliación de su red a los campus de Veterinaria y San Francisco, lo que facilita la recogida de datos de los sensores ubicados al alcance de las nuevas antenas colocadas y ampliar su infraestructura.

En cuanto a implantar el sistema en la biblioteca de Hypatia de Alejandría para, posteriormente, analizar y evaluar dicho trabajo de implantación y uso por parte de los usuarios; se considera que la labor realizada en la biblioteca desde la formación del personal y tareas de difusión hasta la prueba piloto del sistema han sido de utilidad para recoger los primeros datos de uso sobre la biblioteca y poder evaluar el potencial del sistema, ya no solo como sistema de control de aforos, sino como sistema que puede ofrecer datos de calidad para la redacción de informes y el apoyo a la toma de decisiones futuras, habiendo alcanzado el objetivo propuesto con ello.

Finalmente, con el trabajo realizado, se puede concluir que los objetivos del proyecto se han completado satisfactoriamente, resultando de NeoB un sistema completamente funcional e implantado en la biblioteca de Hypatia de Alejandría.

8.2. Líneas de trabajo futuras

Pese a haber finalizado con el desarrollo del TFG y quedar el sistema implantado en una de las salas de la biblioteca de Hypatia de Alejandría, esto no quiere decir que el desarrollo y ampliación del sistema se tenga que abandonar, ya que existen varias ramas de desarrollo por las que todavía se puede seguir trabajando para hacer de NeoB un sistema más completo y útil.

- **Añadir más salas al sistema.** Dada la cantidad de salas de estudio y bibliotecas que hay en la Universidad de Zaragoza y la utilidad demostrada del sistema durante su prueba piloto en la biblioteca de Hypatia de Alejandría, se podría continuar el desarrollo del sistema para ampliar su uso en las bibliotecas y salas de estudio de la EINA y, posteriormente, finalizar abarcando todas las bibliotecas de la Universidad. Esta línea de trabajo es muy prometedora, ya que el sistema es fácilmente configurable en cuanto a franjas de estudio y días de reserva, por lo que se podrían ajustar los parámetros de la forma que mejor se adecuen a cada sala.

- Otra de las líneas de desarrollo posibles es la integración del sistema en los monitores de sensoriZAR, permitiendo comprobar el aforo y ocupación de la sala con precisión además de todos los datos que ya se muestran.
- **Integrar NeoB con el sistema de SIGEUZ**, ya que desde NeoB se hace uso de los identificadores con los que se nombran cada uno de los espacios en el sistema de sigueuz, se podría implementar un panel en el que además de mostrar la información del edificio o espacio, si éste está disponible en NeoB, se mostrase su ocupación.
- **Mostrar la ocupación de la sala en monitores.** Actualmente, para consultar la ocupación de la sala cada usuario debe de acceder al sistema desde un dispositivo móvil u ordenador; sin embargo, se podría adaptar el sistema para que mostrase la ocupación de la sala en tiempo real en un monitor a la entrada de la misma. Esto facilitaría al estudiante conocer los sitios libres antes de entrar, evitando molestias en días de mucha ocupación.
- **Depurar la generación de informes.** Dado que se han propuesto varias plantillas para la representación de los datos recogidos por el sistema, sería interesante continuar con el análisis de los datos recogidos e información obtenida de ellos, planteando nuevas plantillas útiles para el apoyo a la toma de decisiones.

Capítulo 9

Bibliografía

- [1] NeoB, **URL:** <http://neob.unizar.es:3398/> (accedido el: 20/11/2022).
- [2] N. Admin, **URL:** <http://neobadmin.unizar.es:3399/> (accedido el: 20/11/2022).
- [3] Sensorizar, **URL:** <https://eina.unizar.es/sensorizar> (accedido el: 20/11/2022).
- [4] Renfe, **URL:** <https://www.renfe.com/es/es> (accedido el: 20/11/2022).
- [5] Cinesa, **URL:** <https://www.cinesa.es/> (accedido el: 20/11/2022).
- [6] Ryanair, **URL:** <https://www.ryanair.com/es/es> (accedido el: 20/11/2022).
- [7] R. aulas, reserva de aulas: **URL:** <http://reservadeaulas.unizar.es/> (accedido el: 20/11/2022).
- [8] React, *Una biblioteca de JavaScript para construir interfaces de usuario*. **URL:** <https://es.reactjs.org/> (accedido el: 20/11/2022).
- [9] Express, *Infraestructura web rápida, minimalista y flexible para Node.js*. **URL:** <https://expressjs.com/es/> (accedido el: 20/11/2022).
- [10] PostgreSQL, *The World's Most Advanced Open Source Relational Database*. **URL:** <https://www.postgresql.org/> (accedido el: 20/11/2022).
- [11] I. 9075-11:2011, *Information and Definition Schemas (SQL/Schemata)*. **URL:** <https://www.iso.org/standard/53685.html> (accedido el: 20/11/2022).
- [12] P. P. Chen, "The entity-relationship model: Toward a unified view of data," *ACM Trans. on Database System*, vol. 1, no. 1, pp. 1–36, 1976.
- [13] U. de Zaragoza, **URL:** <https://www.unizar.es/> (accedido el: 20/11/2022).

- [14] Nodemon, *nodemon reload, automatically*. URL: <https://nodemon.io/> (accedido el: 20/11/2022).
- [15] Postman, URL: <https://www.postman.com> (accedido el: 20/11/2022).
- [16] MaterialUI, *Sequelize is a modern TypeScript and Node.js ORM*. URL: <https://mui.com/> (accedido el: 20/11/2022).
- [17] Sequelize, *Move faster with intuitive React UI tools*. URL: <https://sequelize.org/> (accedido el: 20/11/2022).
- [18] Nodemailer, *Allow easy as cake email sending*. URL: <https://nodemailer.com/about/> (accedido el: 20/11/2022).
- [19] JWT, *JSON Web Tokens are an open, industry standard RFC 7519 method for representing claims securely between two parties*. URL: <https://jwt.io/> (accedido el: 20/11/2022).
- [20] Axios, *Cliente HTTP basado en Promesas para el navegador y node.js*. URL: <https://axios-http.com/es/> (accedido el: 20/11/2022).
- [21] Ubuntu, *Ubuntu Server 22.04 LTS is a version of the Ubuntu operating system designed and engineered as a backbone for the internet*. URL: <https://ubuntu.com/download/server> (accedido el: 20/11/2022).
- [22] Docker, *Develop faster. Run anywhere*. URL: <https://www.docker.com/> (accedido el: 20/11/2022).
- [23] D. Compose, URL: <https://docs.docker.com/compose/> (accedido el: 20/11/2022).
- [24] INCIBE, *Análisis y gestión de riesgos*, URL: <https://www.incibe.es/protege-tu-empresa/blog/analisis-riesgos-pasos-sencillo> (accedido el: 20/11/2022).
- [25] —, *Catálogo de amenazas de la metodología MAGERIT v3*, URL: https://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Magerit.html (accedido el: 20/11/2022).
- [26] Node, *Node.js is an open-source, cross-platform JavaScript runtime environment*. URL: <https://nodejs.org/en/> (accedido el: 20/11/2022).

Lista de Figuras

2.1. Página principal de reservadeaulas.	8
2.2. Apartado de la pagina de la universidad donde se dispone de los impresos para la solicitud de alquiler de espacios.	8
4.1. Esquema E/R.	17
4.2. Modelo relacional.	18
4.3. Arquitectura del sistema.	19
4.4. Mapa de navegación de NeoB.	22
4.5. Mapa de navegación de NeoB Admin.	22
5.1. Diagrama de secuencias de generación de un token de sesión.	25
5.2. Diagrama de secuencias para la reserva insitu.	26
5.3. Ejemplo de etiqueta finalizada.	27
6.1. Ejemplo de numeración de asientos.	31
6.2. Croquis de disposición de pegado de etiquetas.	32
6.3. Modificación de las franjas de estudio	34
6.4. Cambio de la interfaz gráfica de la aplicación eliminando el botón de “Protocolo COVID”	34
6.5. En el gráfico se muestra el porcentaje medio de uso del sistema.	35
6.6. Porcentaje de reservas sobre el total en cada una de las franjas.	40
6.7. Porcentaje de reservas diario sobre el total semanal.	40
6.8. Porcentaje de reservas semanal sobre el total mensual.	41
6.9. Porcentaje de reservas mensual sobre el total anual.	42
6.10. Asientos más utilizados en la sala.	42
7.1. Numero total de reservas realizadas durante el periodo de prueba piloto.	52
A.1. Diagrama de navegación de NeoB.	68
A.2. Diagrama de navegación de NeoB Admin.	69
B.1. Selección del sistema.	71

B.2. Asignación de memoria RAM y espacio de almacenamiento.	72
B.3. Redireccionamiento de puertos.	72
B.4. Selección de imagen del sistema operativo.	73
B.5. Selección de idioma.	73
B.6. Actualización del instalador y selección del idioma de teclado.	74
B.7. Selección de la versión a instalar.	74
B.8. Configuración de red y repositorio de actualizaciones.	75
B.9. Selección de particionado para la instalación	75
B.10. Resumen del uso de disco y advertencia de borrado.	76
B.11. Datos del perfil.	76
B.12. Instalación del servidor SSH y docker.	77
C.1. Configuración de la interfaz de red para acceder en local a la antena. .	80
C.2. Quick setup de la antena.	81
C.3. Cambio de nombre de la antena.	82
C.4. Configuración de la antena en tts.	84
C.5. Configuración de la LORA basic station.	85
C.6. Certificado para comunicación.	86
C.7. Creación de una API Key.	86
C.8. Permisos necesarios para el token.	87
C.9. Configuración de la antena de veterinaria.	88
C.10. Configuración de la antena de interfaculatdes.	89
C.11. Configuración del NTP.	90
D.1. Resultados de la primera pregunta.	91
D.2. Resultados de la segunda pregunta.	92
D.3. Resultados de la tercera pregunta.	92
D.4. Resultados de la cuarta pregunta.	92
D.5. Resultados de la primera pregunta.	94
D.6. Resultados de la segunda pregunta.	94
D.7. Resultados de la tercera pregunta.	95
D.8. Resultados de la cuarta pregunta.	95

Lista de Tablas

3.1. Especificación de requisitos funcionales de NeoB.	12
3.2. Especificación de requisitos no funcionales de NeoB.	12
3.3. Especificación de requisitos funcionales de NeoB Admin.	13
3.4. Especificación de requisitos no funcionales de NeoB Admin.	13
5.1. Recursos de la máquina virtual utilizada para el despliegue.	28
6.1. Muestra de ocupación tomada de forma presencial en la sala durante los primeros días de prueba.	36
6.2. Total de reservas realizadas cada día.	38
7.1. Activos de la entidad.	46
7.2. Tabla para el cálculo de la probabilidad de que una amenaza se materialice.	47
7.3. Tabla para el cálculo del impacto de una amenaza.	47
7.4. Tabla para el cálculo del riesgo.	47
7.5. Calculo del riesgo que supone cada amenaza.	48
7.6. Principales características de la licencia.	51
7.7. Espacio estimado ocupado por la tabla reserva.	52
D.1. Respuestas a la quinta pregunta	93
D.2. Respuestas a la sexta pregunta	93
D.3. Respuesta a la quinta pregunta	95
D.4. Respuestas a la sexta pregunta	96
F.1. Plazas mas utilizadas en la biblioteca	103

Anexos

Anexos A

Mapas de navegación del sistema

En las dos siguientes páginas se muestran los mapas de navegación del sistema. El primero de ellos corresponde con el mapa de navegación de NeoB, espacio enfocado a los estudiantes y el segundo de ellos, a NeoB Admin, espacio para administradores.

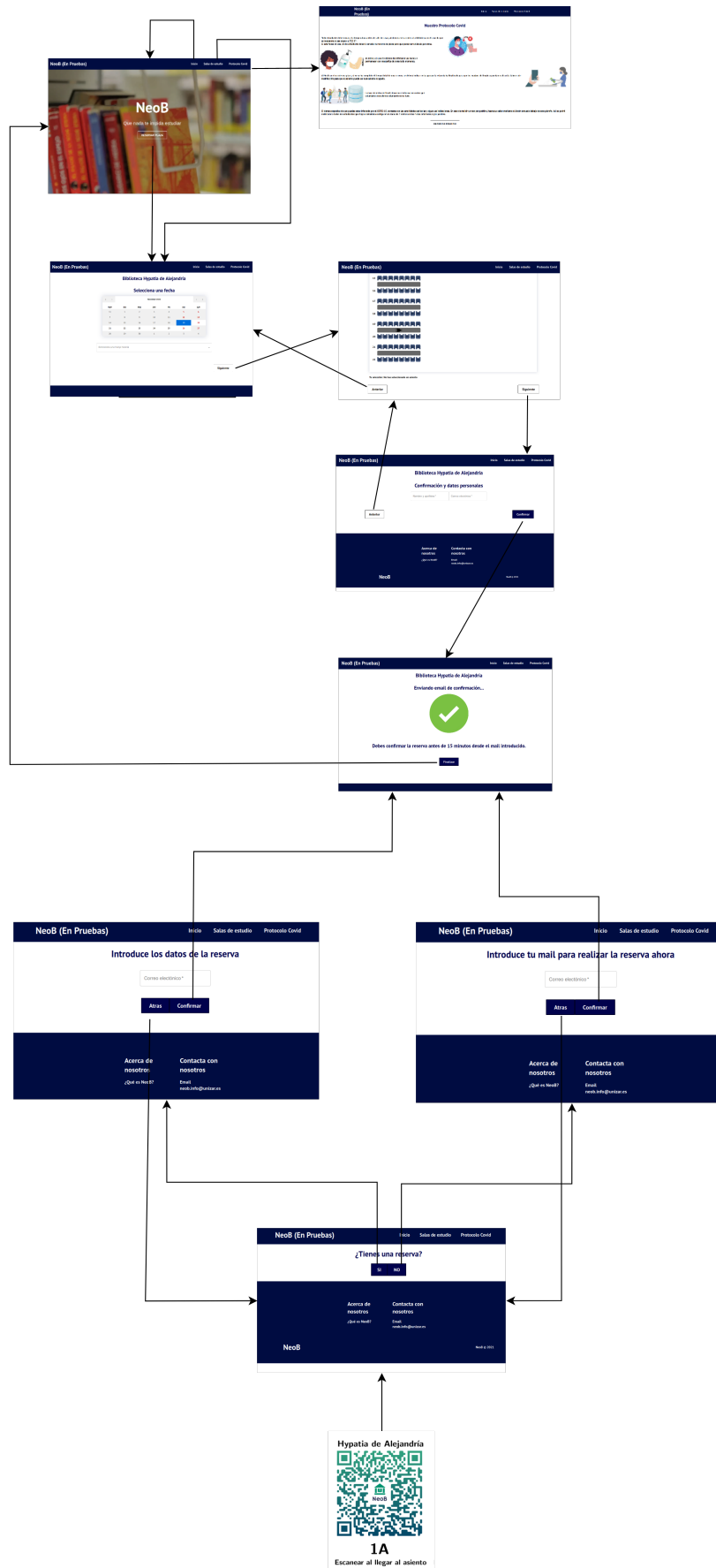


Figura A.1: Diagrama de navegación de NeoB.

Anexos B

Configuración completa de la máquina virtual e instrucciones de despliegue

Para la configuración de la maquina virtual se utiliza la aplicación Oracle Virtual Box, en ella se realiza una primera configuración de la máquina virtual, que posteriormente se exportará en un archivo `.ova` para su despliegue en el servidor final. La máquina configurada en local también sirve como máquina de pruebas para la fase de desarrollo, antes de desplegarse en la maquina en producción.

B.1. Configuración de la máquina virtual

Para la instalación del sistema primero se debe crear la instancia de la maquina virtual. Para ello se crea una nueva maquina virtual en VirtualBox pulsando sobre el boton *new*. Como primer paso se debe seleccionar el tipo de sistema que se va a correr en la maquina y los bits del mismo. En este caso se selecciona Ubuntu 64 bits.

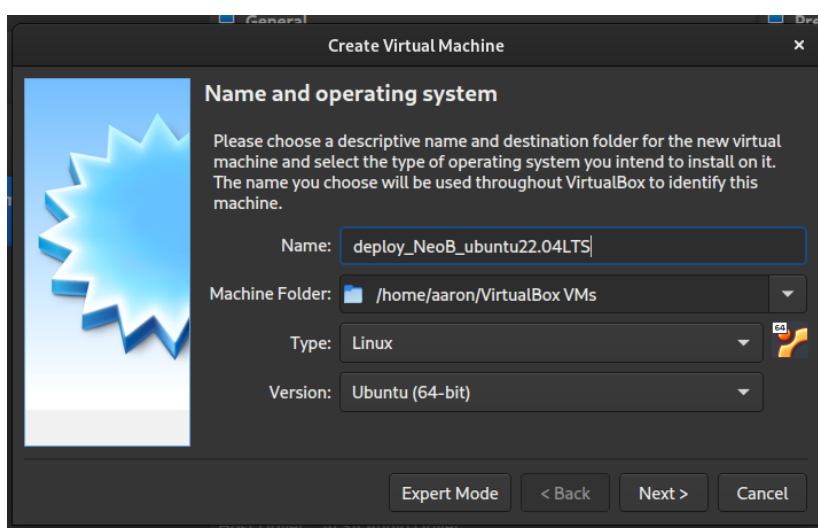


Figura B.1: Selección del sistema.

Tras esto se procede con la asignación de recursos de la máquina virtual. Se le asignan 2Gb (2048 Mb) de memoria RAM y 20Gb de espacio de disco duro. Para la configuración del espacio de disco, se debe seleccionar, *Create a virtual hard disk now*, *VDI (VirtualBox Disk Image)* y *Dynamically Allocated*. Con esta configuración de disco duro, el sistema tiene un almacenamiento máximo de 20Gb, sin embargo, el espacio se reserva dinámicamente, es decir, el disco va aumentando su espacio conforme este se va llenando.

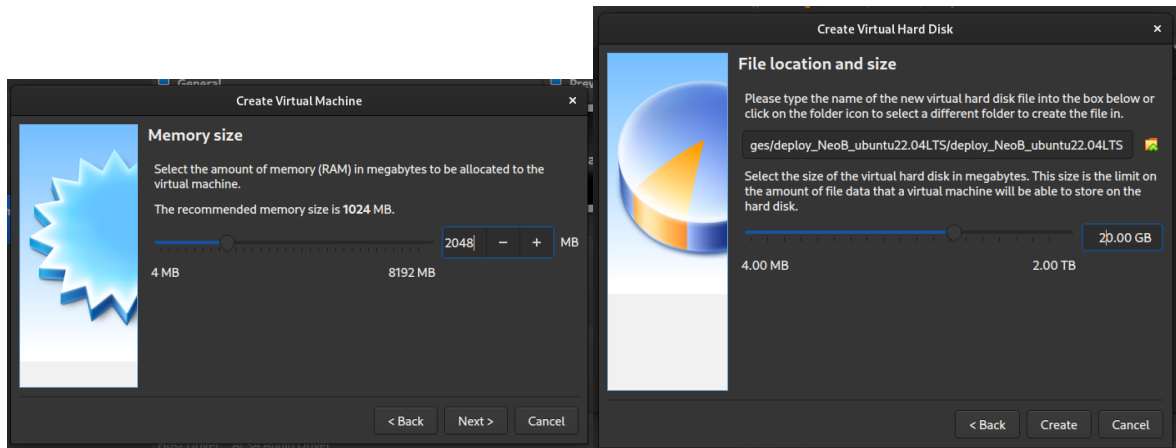


Figura B.2: Asignación de memoria RAM y espacio de almacenamiento.

Finalmente, se configura la redirección de puertos de la máquina virtual al exterior. Para ello con la maquina apagada, se accede a el menu *Settings* → *Network* → *Advanced* → *Port Forwarding* y se definen las redirecciones deseadas. En este caso se van a definir las 4 redirecciones mostradas en la imagen.

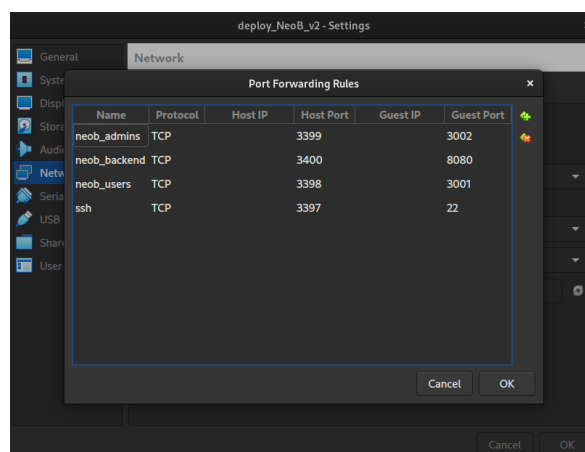


Figura B.3: Redireccionamiento de puertos.

B.1.1. Instalación del sistema operativo

Una vez finalizada la configuración de la máquina virtual, se procede con la instalación del sistema operativo. Para ello se inicia la máquina virtual que se acaba de crear y se carga la imagen de Ubuntu Server 22.04 LTS previamente descargada. Al iniciarse el sistema aparece una pantalla donde se debe seleccionar la opción *Try or install Ubuntu server*, con la que se iniciará el instalador del sistema.

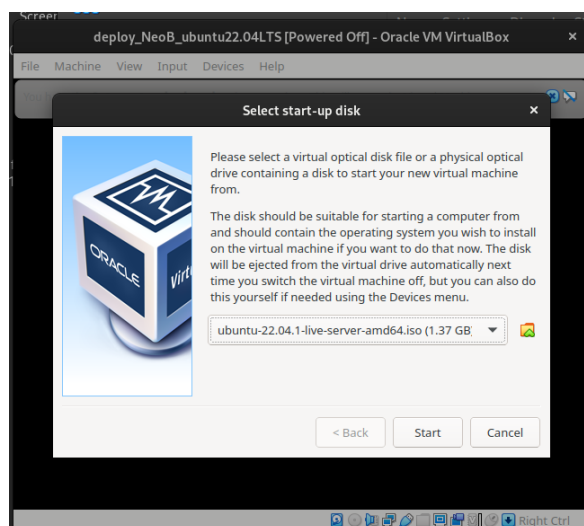


Figura B.4: Selección de imagen del sistema operativo.

Como primer paso se pide seleccionar el idioma del sistema, en este caso se ha elegido Inglés *english*.

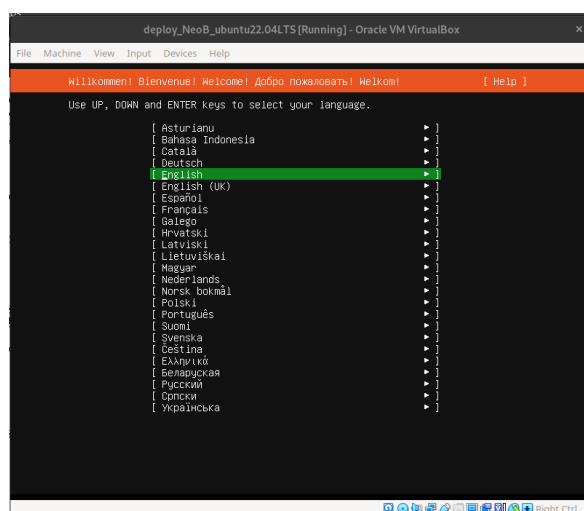


Figura B.5: Selección de idioma.

Tras esto, aparece un aviso que informa de que existe una nueva versión del instalador y si se quiere actualizar, para ello se selecciona *Update to the new installer*. En caso de que esta ventana no aparezca significa que se está utilizando la versión

más actualizada del instalador, por lo que este paso se puede ignorar. Una vez con el instalador mas reciente, se configura la distribución de teclado utilizada, en mi caso personal, utilizo la distribución de teclado en ingles, por lo que selecciono *English US*.

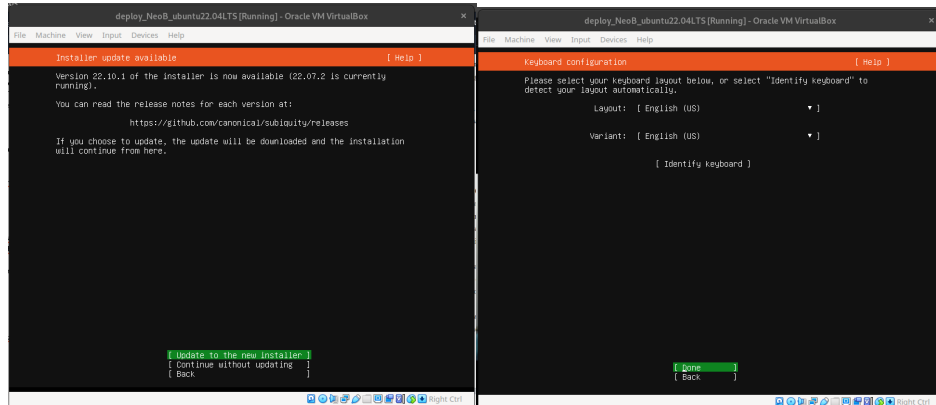


Figura B.6: Actualización del instalador y selección del idioma de teclado.

En el siguiente paso, se debe seleccionar el tipo de instalación que se desea para el sistema y si se quieren instalar controladores de terceros. Las dos versiones disponibles son:

- **Ubuntu Server:** la instalación por defecto, con los paquetes necesarios para una gestión mas cómoda.
- **Ubuntu Server minimized:** el cual viene con un conjunto de paquetes reducido, esta preparado para el bajo consumo de recursos en entornos donde no tienen que acceder los humanos.

Como se requiere de acceso a la máquina para hacer el despliegue se selecciona la versión por defecto y no se considera necesaria la instalación de controladores de terceros.

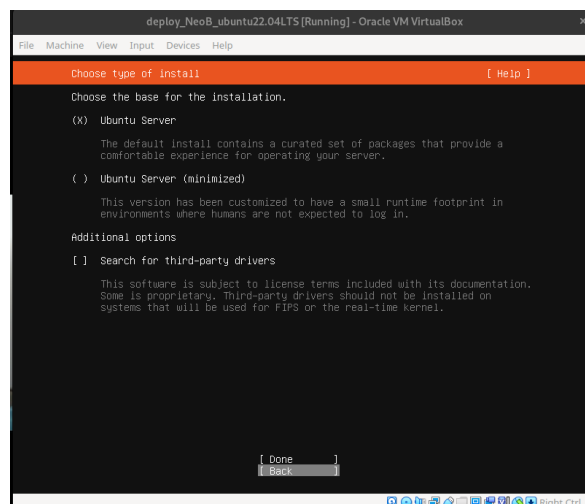


Figura B.7: Selección de la versión a instalar.

En los dos pasos siguientes se realiza la configuración de red y del repositorio de actualizaciones del sistema. Estos se dejan por defecto ya que el instalador los detecta automáticamente.

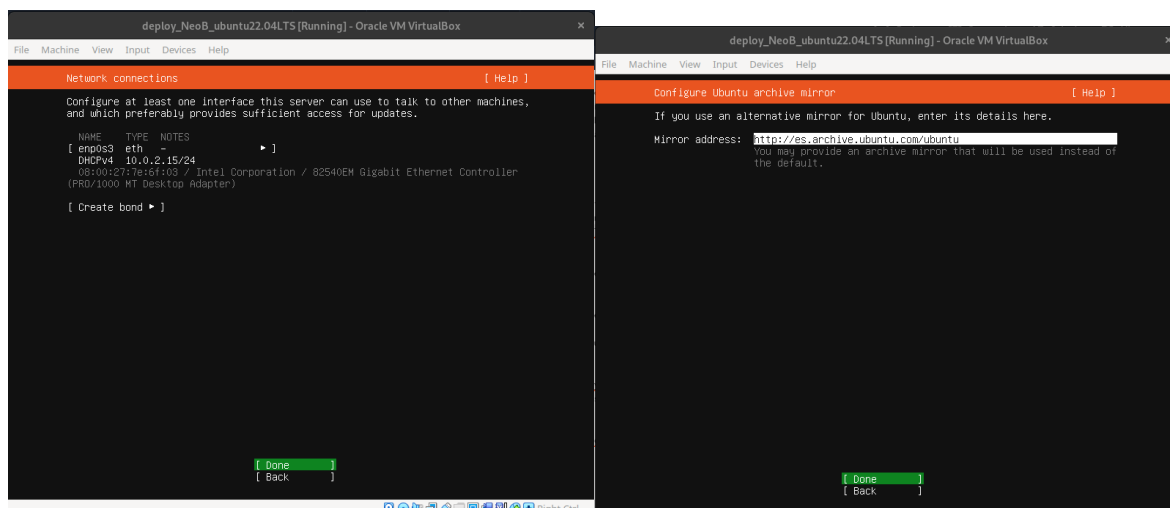


Figura B.8: Configuración de red y repositorio de actualizaciones.

Como últimos pasos de la configuración se selecciona el uso de disco en la instalación, como es el disco de la maquina virtual, se selecciona *Use the entire disk*. Tras esto aparece un resumen de las particiones que se van a realizar y un aviso sobre que se borrarán todos los archivos del disco que se va a utilizar, el que deberemos confirmar dando sobre *continue*.

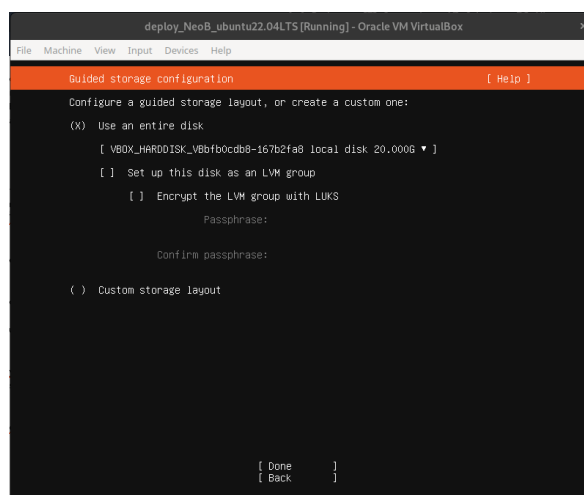


Figura B.9: Selección de particionado para la instalación

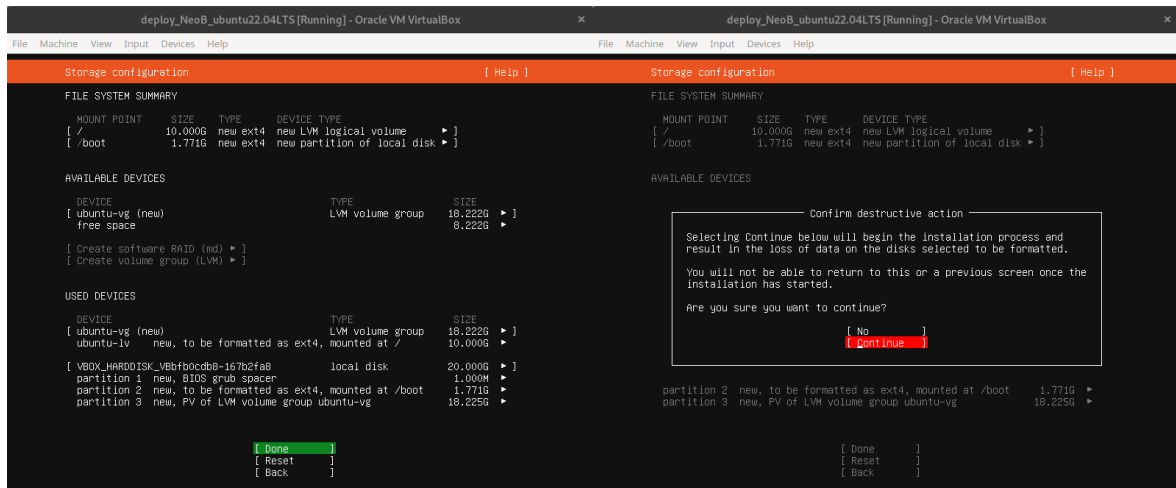


Figura B.10: Resumen del uso de disco y advertencia de borrado.

A continuación se realiza la configuración del perfil, donde se introducen los datos que posteriormente serán utilizados para hacer *log in* en la maquina ya sea mediante acceso local o remoto.

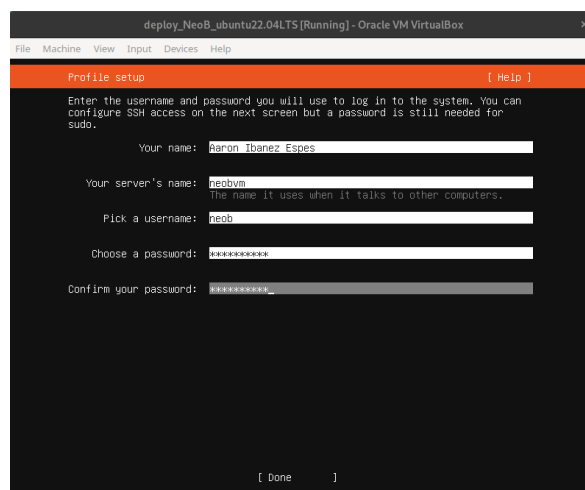


Figura B.11: Datos del perfil.

Finalmente, se instala *OpenSSH server*, para poder acceder a la maquina mediante ssh y se selecciona docker para instalar desde snap. Tras esto se instala el sistema y una vez finalizada la instalación haciendo click sobre *reboot now* se reinicia y ya se ha completado la instalación.

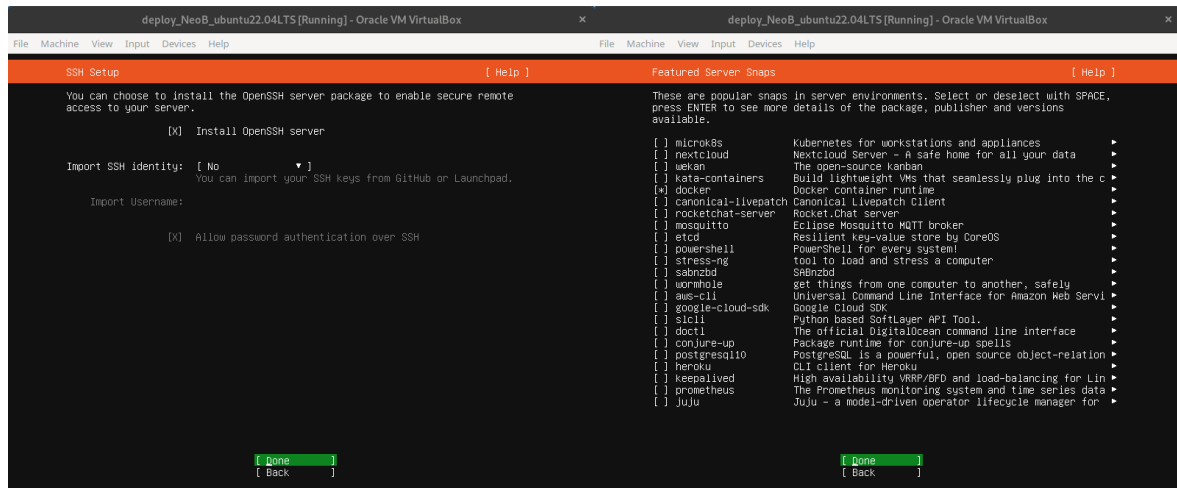


Figura B.12: Instalación del servidor SSH y docker.

Una vez finalizada la instalación, se comprueba que en el sistema se cuenta con docker y docker compose instalados y se procede con la clonación del repositorio a la máquina virtual para su posterior despliegue.

```
git clone https://github.com/aaronainzon17/NeoB_TFG.git
```

Una vez clonado el repositorio se accede al mismo

```
cd NeoB
```

y se despliega haciendo uso de docker compose

```
docker compose up -d
```


Anexos C

Trabajo realizado con sensoriZAR

SensoriZAR conforma un ecosistema para diseñar y rehabilitar edificios inteligentes (digital twins) que, como primer paso, permite monitorizar en tiempo real la información de calidad ambiental, eficiencia energética, ocupación, etc. de los principales espacios de los diversos campus de la Universidad de Zaragoza.

C.1. Configuración de antenas Lorix One

Para la configuración de las antenas Lorix One en el servidor tts (The things stack) de sensoriZAR, fue necesario actualizar el sistema operativo de las antenas, darlas de alta en la instancia de tts y configurar el forwarder de la antena para que envíe los paquetes a ese servidor.

C.1.1. Actualización del sistema operativo de la antena de lorix legacy a LorixOS

Inicialmente se presentaron varios problemas en la configuración de la antena Lorix One. No se podía acceder a ella mediante usb y además, al estar la configuración de la ip estática en la dirección 192.168.1.50, y la red del lugar donde se trató de configurar tenía el formato 192.168.100.1 no permitía su acceso por ssh.

Para solucionar dicho problema se conecta la antena directamente al equipo por el cable ethernet y se configura una interfaz en la que la red tiene el formato deseado, lo que permite su acceso a la antena por ssh.

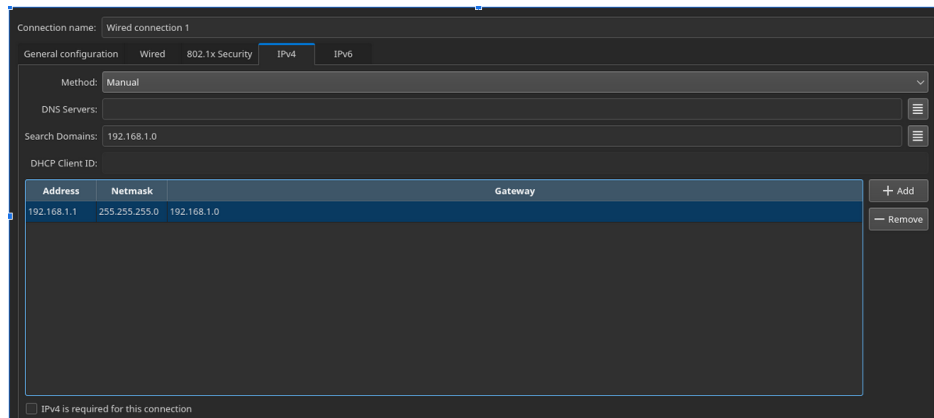


Figura C.1: Configuración de la interfaz de red para acceder en local a la antena.

Tras ello se procede con los pasos del manual para actualizar el sw de la antena del legacy a LorixOs, el cual presenta ciertas mejoras e incorpora un panel web desde el que configurar la antena más fácilmente.

Como primer paso se realiza la copia de seguridad de los ficheros indicados en la guía utilizando el comando,

```
tar -cvf backup.tar /opt/lorix/clouds/packet-forwarder/global_conf.json/
/opt/lorix/clouds/ttn/global_conf.json /opt/lorix/clouds/mp-pkt-fwd/
global_conf.json/etc/network/interfaces
```

y se copian al ordenador con

```
scp admin@192.168.1.50:~/backup.tar .
```

Para la actualización se utiliza la Lorix programming tool, ya que es el método recomendado. Para ello se descarga la herramienta dependiendo del sistema operativo, y la imagen del firmware en el formato *.wimg*, formato recomendado para la herramienta.

En el caso de nuestra antena, ya que es el modelo de 512MB se descarga la imagen Lorix One 512. Tras descargar todo eso se procede con el proceso de actualización siguiendo los siguientes pasos.

1. Conecta la antena al ordenador.
2. Entrar en modo programación. Para ello hay que hacer dos clicks cortos sobre el botón de reset y mantener un tercero por lo menos tres segundos hasta que este comience a parpadear.

3. Una vez la antena haya entrado en modo programación, la herramienta *Lorix programming tool* la detecta automáticamente.
4. Se selecciona la imagen que se quiere instalar, la descargada previamente y se comienza con la actualización.
5. Una vez esta haya finalizado, se puede reiniciar la antena satisfactoriamente y desconectar del cable usb.

Además de ello, tras la actualización, la configuración de red por defecto de la antena deja de ser de ip estática y pasa a ser dhcp, por lo que ahora se le asigna una ip automáticamente al conectarla a la red.

Tras conectar la antena se accede al panel web y se realiza el quick setup recomendado. El acceso al panel web se puede hacer mediante la ip asignada o a través de `https://lorix-one-XXXXXX.local` siendo XXXXXX los 6 últimos dígitos de la mac de la antena.

En él se mantiene la contraseña por defecto, y se le configuran los parámetros de región y ganancia de la manera que se muestra en la imagen inferior.

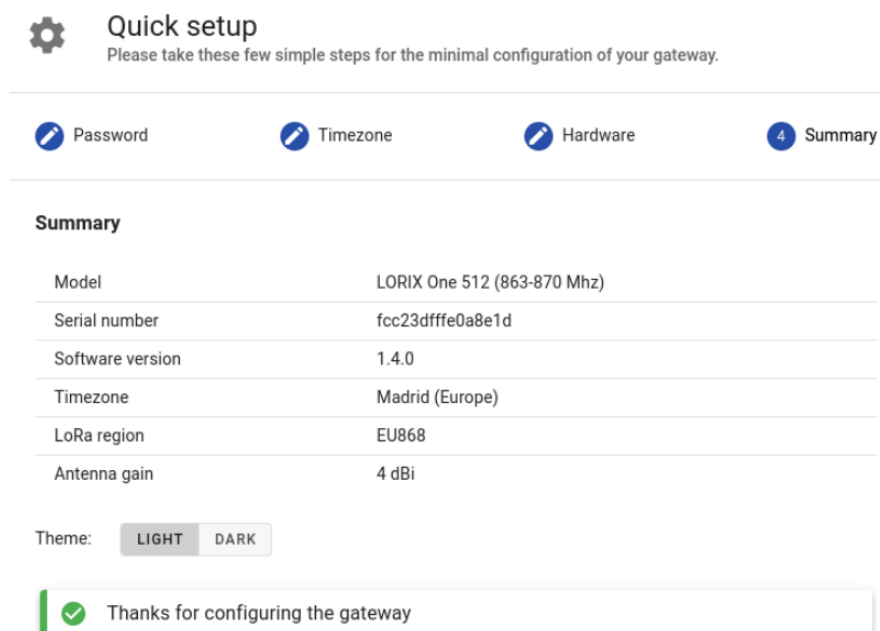


Figura C.2: Quick setup de la antena.

Además de ello se ha modificado el hostname de *lorix-one-0a8e1d* a *lorix-one-03-pruebas* desde el apartado Network → Overview → Hostname

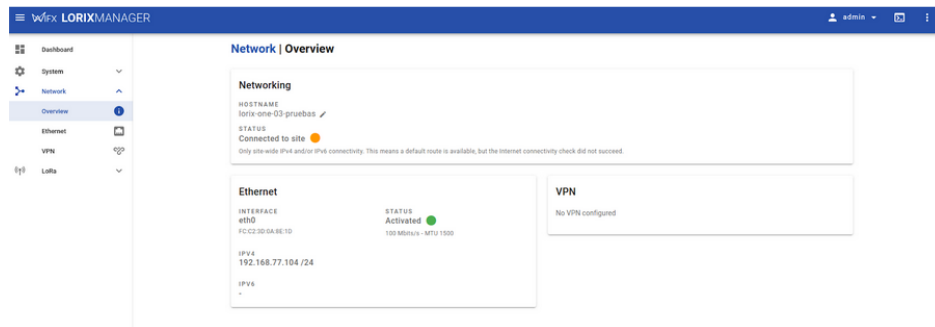


Figura C.3: Cambio de nombre de la antena.

C.1.2. Dar de alta la antena en la instancia de tts privada de sensoriZAR

Para configurar un nuevo gateway en tts se accede a gateways → add gateway. Los campos a rellenar son:

- **Gateway ID:** con un nombre único que identifique el gateway.
- **Gateway EU:** se completa con el serial del gateway.
- El apartado **require authenticated connection** debe estar activado.
- El resto de apartados se dejan por defecto.

Finalmente el panel de configuración del gateway tiene que quedar como se muestra en la siguiente figura.

Basic settings

Collapse

General settings, gateway updates and metadata

Gateway ID ⓘ *

lorix-one-03-pruebas

Gateway EUI ⓘ

FC C2 3D FF FE 0A 8E 1I

Gateway name ⓘ

lorix-one-03-aaron

Gateway description ⓘ

Description for my new gateway

Optional gateway description; can also be used to save notes about the gateway

Gateway Server address

sensorizar.unizar.es

The address of the Gateway Server to connect to

Require authenticated connection ⓘ

☒ Enabled

Controls whether this gateway may only connect if it uses an authenticated Basic Station or MQTT connection

LoRa Basics Station LNS Authentication Key

The Authentication Key for Lora Basics Station LNS connections. This field is ignored for other gateways.

Gateway status ⓘ

☒ Make location public

The status of this gateway may be visible to other users

Gateway location ⓘ

☒ **Make location public**
When set to public, the gateway location may be visible to other users of the network

Attributes ⓘ

[+ Add attributes](#)
Attributes can be used to set arbitrary information about the entity, to be used by scripts, or simply for your own organization

Automatic updates

☐ **Enabled**
Gateway can be updated automatically

Channel

Channel for gateway automatic updates

Packet Broker ⓘ

☐ **Disabled**
Disable forwarding uplink messages received from this gateway to the Packet Broker

[Save changes](#) [Delete gateway](#)

Figura C.4: Configuración de la antena en tts.

C.1.3. Configurar el forwarder de la antena

Para configurar el forwarder se realiza desde Lora → Forwarder, e crea una Lora Basic Station y se configura el Lora Network Server con los siguientes parámetros.

- **Address:** sensorizar.unizar.es
- **Port:** 8887.
- Seguridad TLS activada, es necesario cargar el certificado obtenido de la página.
- **Client Authentication:** Use Token, para ello se crea un token desde tts.

LoRaWAN Network Server

Server contacted by the Basic Station to forward LoRaWAN packets.

Address: sensorizar.unizar.es Port: 8887

☒ Secured TLS connection

Server authentication

☐ Use local root certificates

☒ Use certificate file

This is the recommended setup for on-premise servers

Certificate details (bundle)

GEANT OV RSA CA 4

Expires in 4044 days

< 1 out of 2 >

Client authentication

☐ Use certificate and key

☒ Use token

Authentication token

Figura C.5: Configuración de la LORA basic station.

Para obtener el certificado necesario para el *forwarder* se accede al tts de sensoriZAR y se realiza lo siguiente.

- Se hace click sobre el icono del candado.
- Connection secure.
- More information.
- View certificates.
- Se descarga la cadena de certificados del sitio.

- Una vez descargada la cadena se elimina el certificado de sensorizar.unizar.es ya que no es necesario y caduca en 1 año, se mantiene el resto de la cadena de certificados.

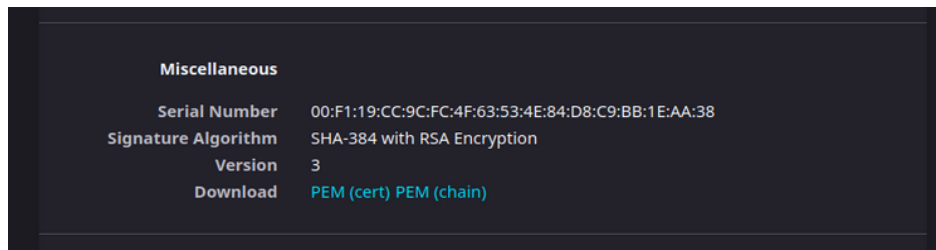


Figura C.6: Certificado para comunicación.

Para la creación de un token se accede a la gateway deseada en tts → API Key → Add API Key.

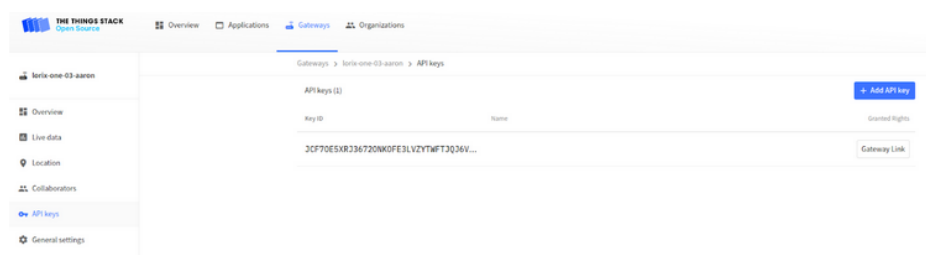


Figura C.7: Creación de una API Key.

Para el token solamente será necesario Grant Individual rights → Link as Gateway to a Gateway server for traffic exchange.

Add API key

Name

Rights*

☐ Grant all current and future rights
☒ Grant individual rights
☒ Select all

- ☐ Delete gateway
- ☐ View gateway information
- ☒ Link as Gateway to a Gateway Server for traffic exchange, i.e. write uplink and read downlink
- ☐ View gateway location
- ☐ Retrieve secrets associated with a gateway
- ☐ View and edit gateway API keys
- ☐ Edit basic gateway settings
- ☐ View and edit gateway collaborators
- ☐ View gateway status
- ☐ Write downlink gateway traffic
- ☐ Read gateway traffic
- ☐ Store secrets for a gateway

[Create API key](#)

Figura C.8: Permisos necesarios para el token.

C.2. Montaje de las antenas en Veterinaria y San Francisco

Tras actualizar el sistema operativo de todas las antenas Lorix One que tenían el software legacy y darlas de alta en el servidor ttn de sensoriZAR, se esperó hasta que se nos proporcionases los parámetros de configuración de la antena dentro de la red de UNIZAR para su configuración final y montaje en el tejado de los edificios de Veterinaria y San Francisco.

Configuración de la antena de veterinaria

Los parámetros de configuración de la antena de veterinaria son los siguientes.

- **IP:** 10.5.68.237
- **Mascara:** 255.255.255.0

- **Router:** 10.5.68.254
- **PoE:** SI (hay que probar la compatibilidad)
- **DNS:** 155.210.33.4, 155.210.3.12
- **NTP:** ntp.unizar.es
- **Salida Internet:** SI [NAT en FW de salida]

Para configurar la antenna con los parámetros de IP estática necesarios para su montaje, se accede a la interfaz gráfica de configuración y desde el apartado Network → Ethernet → Settings → IPV4, se configura la antenna como se muestra en la siguiente figura.

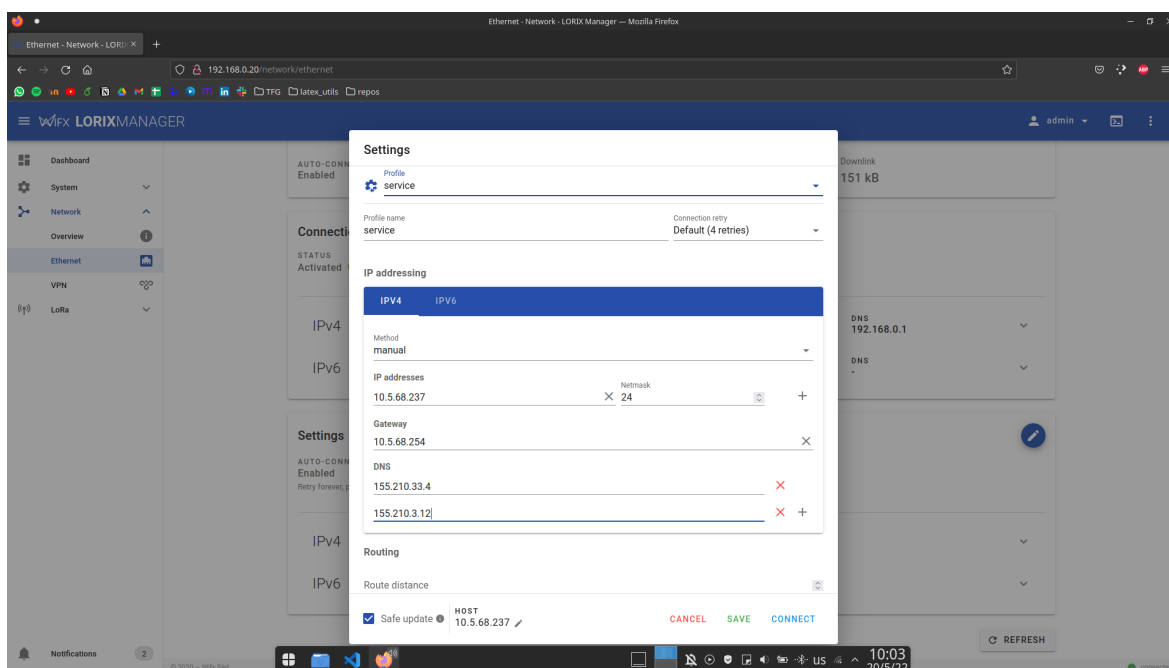


Figura C.9: Configuración de la antenna de veterinaria.

Configuración de la antenna de San Francisco

Los parámetros de configuración de la antenna de veterinaria son los siguientes.

- **IP:** 10.2.68.238
- **Mascara:** 255.255.255.0
- **Router:** 10.2.68.254
- **PoE:** SI (hay que probar la compatibilidad)
- **DNS:** 155.210.33.4, 155.210.3.12

— **NTP:** ntp.unizar.es

— **Salida Internet:** NO

Para configurar la antenna con los parámetros de IP estática necesarios para su montaje, se accede a la interfaz gráfica de configuración y desde el apartado Network → Ethernet → Settings → IPV4, se configura la antenna como se muestra en la siguiente figura.

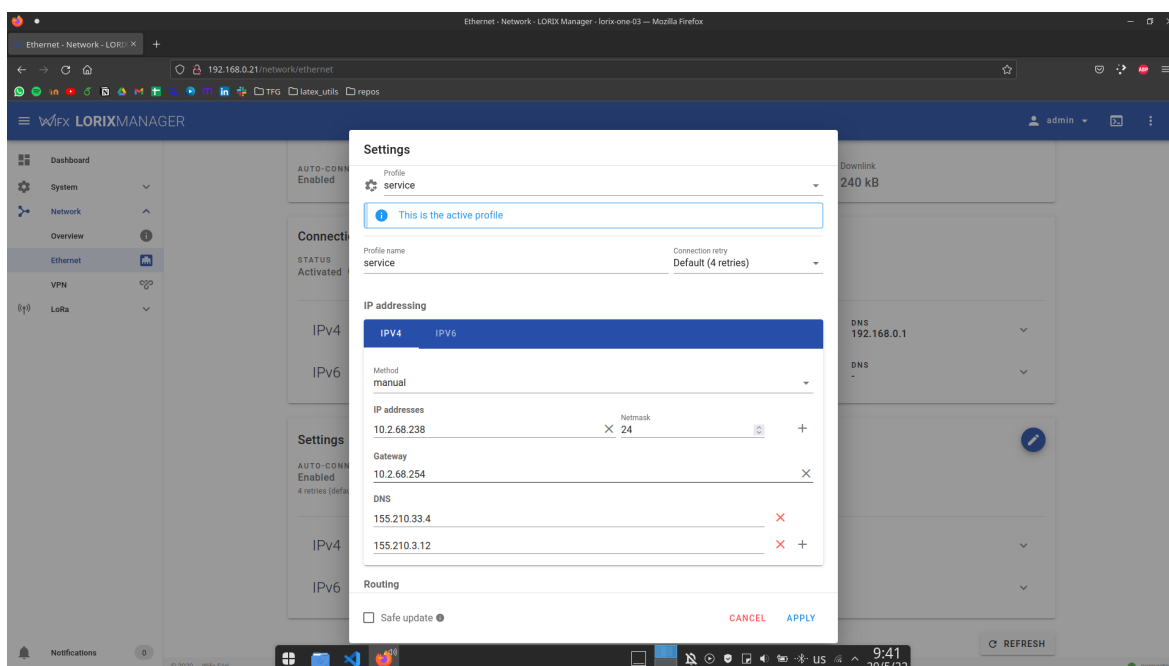


Figura C.10: Configuración de la antenna de interfaculatdes.

Dado que esta antenna no tiene salida a internet, es necesario la configuración del NTP para poder sincronizar los relojes de la antenna y el sistema. Esto se hace desde el apartado System → Regional settings → Time settings → Use custom servers, como se muestra en la siguiente figura.

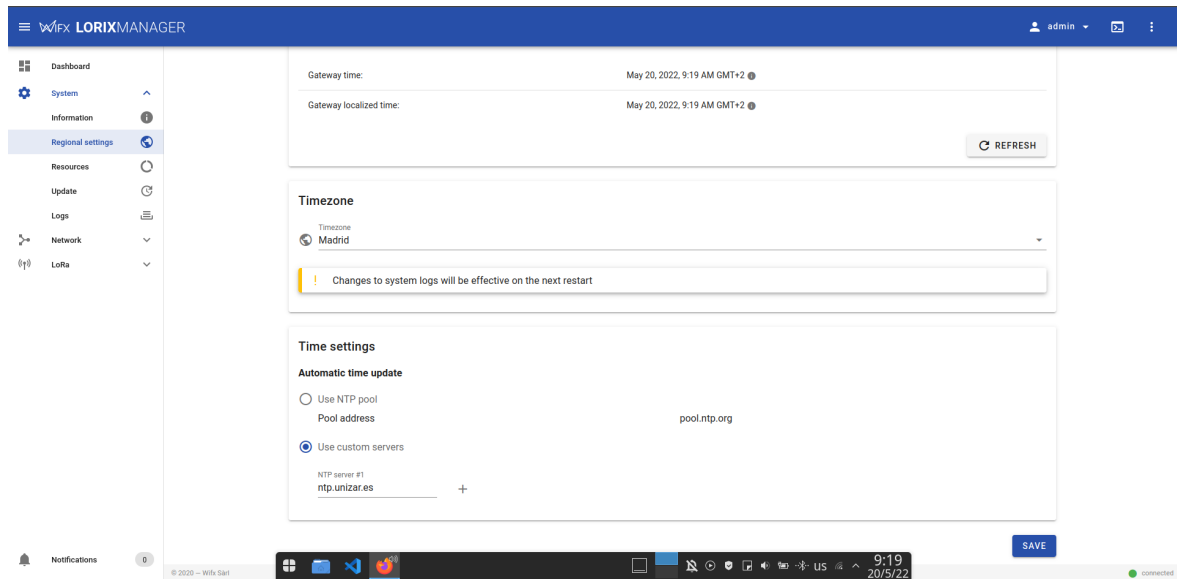


Figura C.11: Configuración del NTP.

Finalmente, para su montaje en el tejado de ambos edificios se contó con la ayuda del servicio de mantenimiento de la universidad, el cual fué el encargado de realizar la instalación de las antenas en el mástil de cada edificio, siendo nuestra labor la de comprobar que el puerto ethernet previamente instalado para la antena funcionaba correctamente y tras instalar la antena, comprobar esta se encendía y funcionaba sin problemas.

Anexos D

Resultados de las encuestas

En este anexo se muestran las gráficas y opiniones obtenidas con las encuestas realizadas a los estudiantes y personal de la biblioteca.

D.1. Encuesta realizada a los estudiantes

Primera pregunta

¿Cómo te has enterado de la existencia de NeoB?
36 respuestas

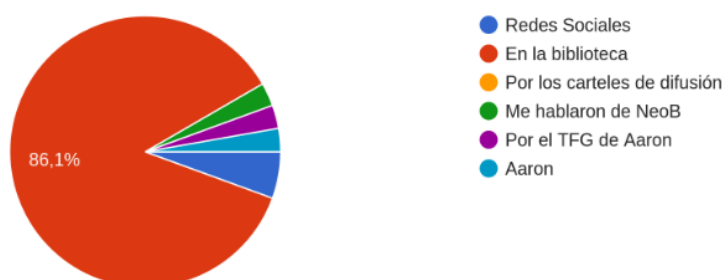


Figura D.1: Resultados de la primera pregunta.

Segunda pregunta

¿ Te parece NeoB sencillo de utilizar ?

36 respuestas

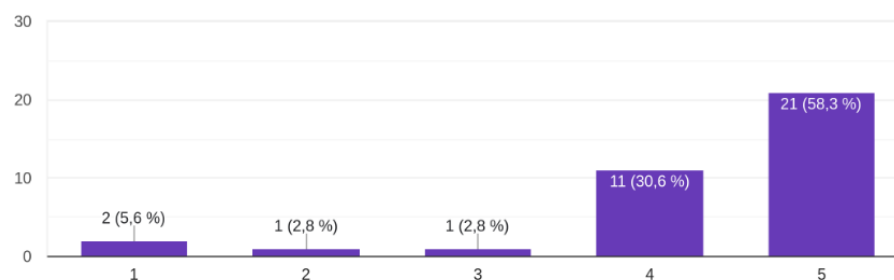


Figura D.2: Resultados de la segunda pregunta.

Tercera pregunta

¿Crees que NeoB ofrece un servicio de utilidad?

36 respuestas

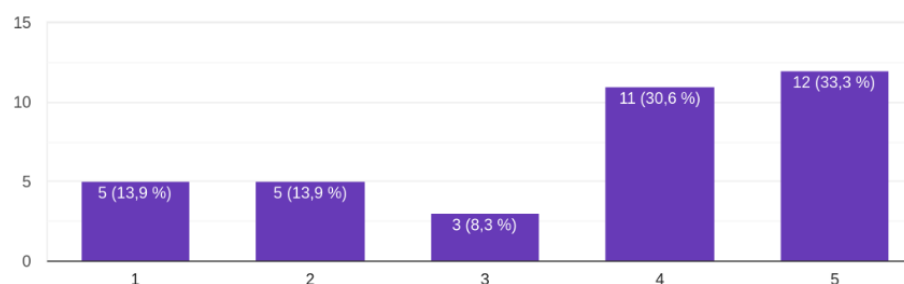


Figura D.3: Resultados de la tercera pregunta.

Cuarta pregunta

¿Has tenido algún problema con el sitio en la biblioteca antes?

36 respuestas

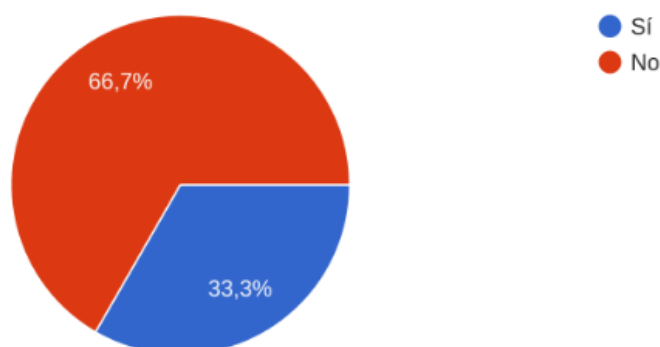


Figura D.4: Resultados de la cuarta pregunta.

Quinta pregunta

En caso de que la pregunta anterior sea sí, ¿Cuál fue el problema, crees que NeoB puede resolverlo?
Que no había sitio, si se reserva con prioridad a los universitarios funcionaría bien
Problema por falta de sitio cerca de los enchufes de puede resolver ya que se ve el plano de la sala
Biblioteca llena
No encontrar sitio. Si, puede resolverlo
Para saber la cantidad de aforo
llegar a la biblioteca y que no haya sitio, necesitar enchufe y que no haya sitio con enchufe
No
El problema fue que la gente estudia mucho y no habia sitio. Pues si lo hubiese reservado antes, podria haber tenido mi sitio.
Pelea por sillas en economicas
Para examenes la bibliotece se llena nada mas abrirse y hay que llegar a la sala antes de que abra para poder coger sitio, por lo que considero que si puede resolver el problema reservando desde casa
No sabia entrar, de momento se ha resuelto
Llegué y no había sitio
Que no había sitio. Creo que sí que puede resolverlo

Tabla D.1: Respuestas a la quinta pregunta

Sexta pregunta

Otros comentarios que nos desees hacer para mejorar el sistema o la biblioteca.
No es muy útil ya que a muchas personas se les olvidara quitar la reserva y ese sitio siempre permanecerá ocupado
Proporcionar x sitios de la biblioteca (no todos) que se pudieran reservar desde distancia para tener sitio asegurado.
cuando abandonas el sitio no se puede comunicar para que lo disponga otra persona
Las dos franjas de reserva (mañana o tarde) igual se quedan cortas ya que hay gente que va a estudiar en el horario de paso de una franja a otra
Mucha gente se va de su sitio y no finaliza su reserva por lo que el sitio se queda ocupado, podrias hacer algo para que fuese mas sencillo liberar el sitio
No está muy claro el procedimiento para reservar sitio desde fuera de la biblioteca a la peña se le olvida dejar como libre el sitio cuando se van

Tabla D.2: Respuestas a la sexta pregunta

D.2. Encuesta realizada al personal de la biblioteca

Primera pregunta



Figura D.5: Resultados de la primera pregunta.

Segunda pregunta

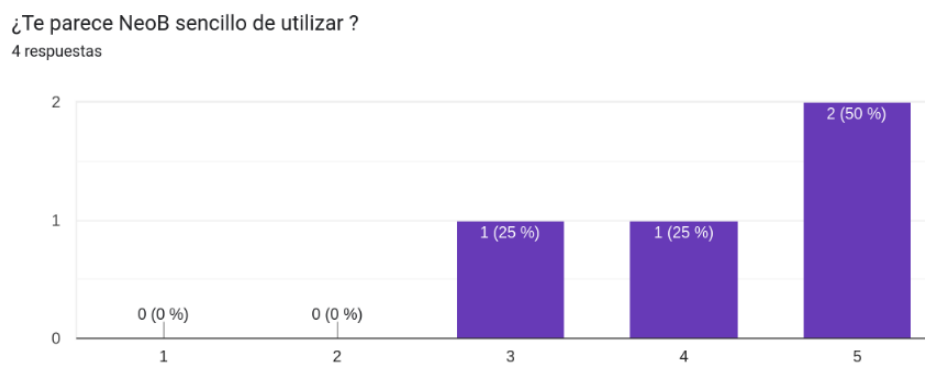


Figura D.6: Resultados de la segunda pregunta.

Tercera pregunta

¿Consideras que la formación que se impartió ha sido útil?
4 respuestas

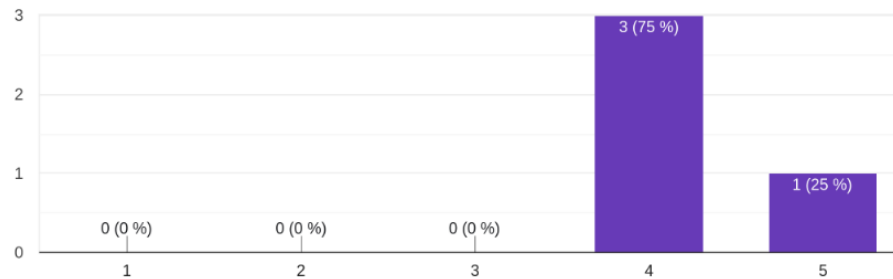


Figura D.7: Resultados de la tercera pregunta.

Cuarta pregunta

¿Alguna vez has detectado conflictos entre usuarios de la sala por el uso de puestos (por ejemplo, un usuario retira los apuntes del otro, alguien deja s... y no lo usa por un largo periodo de tiempo, etc.)?
4 respuestas

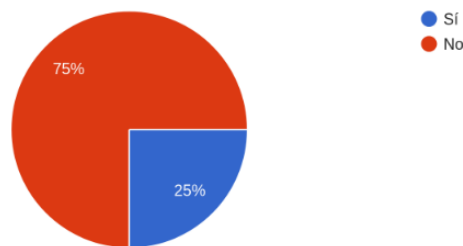


Figura D.8: Resultados de la cuarta pregunta.

Quinta pregunta

**En caso de que la pregunta anterior sea sí,
¿Cuál fue el problema, crees que NeoB puede resolverlo,
aunque sea parcialmente?**

Los problemas la verdad es que son en épocas muy puntuales, pero sí que podría ayudar. También es posible que al principio los pueda aumentar si se sientan porque está libre sin escanear y luego hay que levantarlo porque está reservado. La verdad es que hasta que no esté en funcionamiento plenamente no podemos saber si ayuda o lo aumenta.

Tabla D.3: Respuesta a la quinta pregunta

Sexta pregunta

Otros comentarios que consideres de utilidad para mejorar el sistema por por parte de los administradores.

Convendría que el panel que manejamos nosotros las letras estén fijas, porque si vamos a las últimas filas no vemos claro que letra es. Además vendría bien hacer más propaganda con las ventajas que pueden tener los usuarios para que realmente lo utilicen. Actualmente muchos no lo hacen.

Mayor difusión de la app para que los usuarios sean conocedores de la utilidad del sistema

Tabla D.4: Respuestas a la sexta pregunta

Anexos E

Scripts de automatización para la generación de los códigos QR

En este anexo se incluye el código de los scripts que han realizado para la automatización de la creación de las etiquetas que identifican cada uno de los asientos.

generarNumeracion.py

```
1 import sys
2 import string
3 import numpy as np
4
5 if len(sys.argv) < 3:
6     print("Usage: generarNumeracion.py <file_name> <n_letras> <n_filas>")
7     exit(1)
8
9 file_name = sys.argv[1]
10 n_letras = sys.argv[2]
11 n_filas = sys.argv[3]
12
13 f = open(file_name, "a")
14 letras = string.ascii_lowercase[:int(n_letras)]
15 filas = values = np.arange(0, int(n_filas), 1)
16
17 for fila in filas:
18     for letra in letras:
19         puesto = str(1+fila) + str(letra).upper() + '\n'
20         f.write(puesto)
21
22 f.close()
```

Listing E.1: Código del script generarNumeracion.py

qr_generator.py

```
1 import sys
2 import os
3 import qrcode
4 from qrcode.image.styledpil import StyledPILImage
```

```

5 from qrcode.image.styles.moduledrawers import RoundedModuleDrawer
6 from qrcode.image.styles.colormasks import VerticalGradientColorMask
7
8 #La url actual es la de Hypatia de Alejandria
9 url = 'http://neob.unizar.es:3398/escanear/CRE.1201.01.480/'
10 """
11     - Tomas Pollan: CRE.1201.00.400
12     - Hypatia: CRE.1201.01.480
13     - Ada Byron: CRE.1200.02.540
14     - Hemeroteca: CRE.1201.02.690
15 """
16 if len(sys.argv) < 2:
17     print("Usage: generarNumeracion.py <file_name>")
18     exit(1)
19
20 file_name = sys.argv[1]
21 folder_name = file_name.split('.')[0]
22
23 if os.path.exists(folder_name):
24     print('La carpeta', folder_name, 'ya existe')
25     print('Eliminando', folder_name, '...')
26     os.system('rm -rf ' + folder_name)
27
28 os.system('mkdir ' + folder_name)
29
30 with open(file_name) as file:
31     while puesto := file.readline():
32         qr = qrcode.QRCode(version=4, box_size=18,
33             border=1,error_correction=qrcode.constants.ERROR_CORRECT_H)
34         qr.add_data(url + puesto.rstrip())
35
36         qr_code = qr.make_image(image_factory=StyledPillImage,
37             module_drawer = RoundedModuleDrawer(),
38             color_mask=VerticalGradientColorMask(
39                 back_color=(255, 255, 255),
40                 top_color=(50, 165, 128),
41                 bottom_color=(2, 70, 103)),
42             embeded_image_path="./NeoB_logo.png")
43         qr_name = puesto[:len(puesto) - 1] + '.png'
44         print(qr_name)
45         qr_code.save(qr_name)
46         os.system('mv ' + qr_name + ' ./' + folder_name)

```

Listing E.2: Código del script qr_generator.py

creacion_etiquetas.py

```

1 import sys
2 import os
3
4 def fill_template(qr1, qr2,qr3, folder_name, biblioteca):
5     img1_name = str(qr1).split('.')[0]
6     img2_name = str(qr2).split('.')[0]
7     img3_name = str(qr3).split('.')[0]
8
9     template = """
10     \\begin{table}[H]
11     \\centering
12     \\begin{tabular}{ccc}

```

```

13     \\textbf{\\Large{"" + biblioteca + ""}} &
14     \\textbf{\\Large{"" + biblioteca + ""}} &
15     \\textbf{\\Large{"" + biblioteca + ""}} \\\\
16     \\includegraphics[width=0.28\\textwidth]
17         {"" + folder_name + '/' + qr1 + ""} \\vspace{0.2cm} &
18     \\includegraphics[width=0.28\\textwidth]
19         {"" + folder_name + '/' + qr2 + ""} &
20     \\includegraphics[width=0.28\\textwidth]
21         {"" + folder_name + '/' + qr3 + ""} \\\\
22     \\textbf{\\Huge{"" + img1_name + ""}} \\vspace{0.2cm} &
23     \\textbf{\\Huge{"" + img2_name + ""}} &
24     \\textbf{\\Huge{"" + img3_name + ""}} \\\\
25     \\textbf{\\large{Escanear al llegar al asiento}} &
26     \\textbf{\\large{Escanear al llegar al asiento}} &
27     \\textbf{\\large{Escanear al llegar al asiento}}
28     \\end{tabular}
29     \\end{table}
30     ""
31     return template
32
33     ###      MAIN      ###
34     ini = ""
35     \\documentclass{article}
36
37     % Language setting
38     % Replace 'english' with e.g. 'spanish' to change the document language
39     \\usepackage[spanish]{babel}
40
41     % Set page size and margins
42     % Replace 'letterpaper' with 'a4paper' for UK/EU standard size
43     \\usepackage[letterpaper,top=2cm,bottom=2cm,
44         left=1.5cm,right=1.5cm,marginparwidth=1.75cm]{geometry}
45
46     % Useful packages
47     \\usepackage{amsmath}
48     \\usepackage{graphicx}
49     \\usepackage[colorlinks=true, allcolors=blue]{hyperref}
50     \\usepackage{float}
51     %Cambio del tipo de letra
52     \\renewcommand{\\familydefault}{\\sfdefault}
53
54     \\begin{document}
55     ""
56
57     if len(sys.argv) < 3:
58         print("Usage: generarNumeracion.py <qr_folder> <nombre_biblioteca>")
59         exit(1)
60
61     folder_name = 'qrs/' + str(sys.argv[1])
62     biblioteca = "Hypatia de Alejandria"
63     file_name = str(sys.argv[1]) + '.txt'
64
65
66     if os.path.exists('main.tex'):
67         print('main.tex ya existe')
68         exit(1)
69
70

```

```

71 f = open('main.tex', "a")
72 f.write(ini)
73
74 qrs_line = ['None', 'None', 'None']
75 max_line = 0
76
77
78 with open(file_name) as file:
79     while qr := file.readline():
80         qrs_line[max_line] = qr.split('\n')[0] + '.png'
81         if max_line == 2:
82             table = fill_template(qrs_line[0],
83                                   qrs_line[1],
84                                   qrs_line[2],
85                                   folder_name,
86                                   biblioteca)
87             f.write(table)
88
89             max_line = 0
90             qrs_line = ['None', 'None', 'None']
91
92         else:
93             max_line+=1
94
95 table = fill_template(qrs_line[0],
96                       qrs_line[1],
97                       qrs_line[2],
98                       folder_name,
99                       biblioteca)
100 f.write(table)
101 f.write('\end{document}')
102
103 f.close()

```

Listing E.3: Código del script creación_etiquetas.py

Anexos F

Consultas e información extraída de ellas para la realización de las gráficas

En este apartado se detallan las consultas realizadas a la base de datos para obtener la información necesaria para la realización de las posteriores gráficas para el apoyo a la toma de decisiones.

Primera plantilla

Para realizar la primera plantilla es necesario conocer el número de reservas totales que hay en para cada una de las franjas horarias de estudio. Por ello, las consultas diseñadas para ello son las siguientes.

Reservas totales en el turno de mañanas

```
SELECT COUNT(FECHA) AS NRESERVAS FROM RESERVA WHERE  
    HORAINICIO='08:00' OR HORAINICIO='9:00' OR  
    HORAINICIO='10:00' OR HORAINICIO='11:00' OR  
    HORAINICIO='12:00' OR HORAINICIO='13:00' OR  
    HORAINICIO='08:15';
```

Reservas totales en el turno de tardes

```
SELECT COUNT(FECHA) AS NRESERVAS FROM RESERVA WHERE  
    HORAINICIO='14:00' OR HORAINICIO='15:00' OR  
    HORAINICIO='16:00' OR HORAINICIO='17:00' OR  
    HORAINICIO='18:00' OR HORAINICIO='18:00' OR  
    HORAINICIO='19:00' OR HORAINICIO='20:00';
```

Cabe destacar que la consulta se ha realizado de esta manera debido a que como se comenta con anterioridad, se comenzó la validación experimental con franjas de estudio de una hora y se finalmente se establecieron dos franjas en todo el día. Para casos futuros en los que las franjas se mantengan en dos franjas de estudio de 08:15 a 14:00 y de 14:00 a 20:30, las consultas se simplifican de la siguiente manera.

Reservas totales en el turno de mañanas

```
SELECT COUNT(FECHA) AS NRESERVAS FROM RESERVA WHERE  
    HORAINICIO='08:15';
```

Reservas totales en el turno de tardes

```
SELECT COUNT(FECHA) AS NRESERVAS FROM RESERVA WHERE  
    HORAINICIO='14:00';
```

Siguientes plantillas

Para estas plantillas se necesita conocer el número de reservas realizadas cada día, por lo que se realiza la siguiente consulta.

```
SELECT FECHA, COUNT(FECHA) AS NRESERVAS FROM RESERVA  
    GROUP BY FECHA ORDER BY NRESERVAS DESC;
```

Esta consulta devuelve el número de reservas que se ha realizado cada uno de los días en la sala en forma de lista. Una vez se dispone de dichos datos en crudo, se pueden filtrar los resultados por días, semanas o meses para a partir de ahí crear la plantilla deseada.

Asientos más utilizados

Para obtener los datos necesarios para esta plantilla se realiza la siguiente consulta.

```
SELECT IDASIENTO, COUNT(IDASIENTO) AS A FROM RESERVA  
    GROUP BY IDASIENTO ORDER BY A DESC LIMIT 20;
```

Como se observa en la sentencia, se ha tomado la decisión de limitar la salida a los 20 asientos mas usados, sin embargo, eliminado la parte de la consulta *LIMIT 20*, se mostrarán todos los resultados. Además de la gráfica que se genera, se ha tomado la decisión de añadir una tabla donde se ordenan los asientos mas utilizados en orden descendente.

Identificador de asiento	Número de veces reservado
1A	9
12J	8
9I	8
16A	7
12X	7
7A	7
22H	6
11X	6
3I	6
10X	6
21A	5
5X	5
1H	5
11H	5
18E	5
20H	5
9X	5
22A	5
19A	4
7X	4

Tabla F.1: Plazas mas utilizadas en la biblioteca

Otras consultas de interés

Además de las consultas necesarias para la elaboración de las gráficas, se han decidido añadir unas consultas que se consideran de interés para conocer mas datos sobre el comportamiento de los estudiantes.

Número de reservas totales en el sistema

```
SELECT COUNT(IDRESERVA) FROM RESERVA;
```

Número de reservas realizadas a través de la aplicación web

Dado que no se cuenta con un parámetro específico para indicar que una reserva se ha realizado a través de la aplicación web o insitu, se utiliza en parámetro *HORACONFIRMACION*, el cual en caso de haber realizado desde casa se actualiza a la hora en la que se confirmó la reserva, sin embargo, si se realiza insitu, al realizarse la reserva y su inmediato estudio, no se establece hora de confirmación, sino que se da comienzo al estudio directamente.

```
SELECT COUNT(IDRESERVA) FROM RESERVA WHERE HORACONFIRMACION!='NO CONFIRMADA';
```

Número de reservas realizadas insitu

```
SELECT COUNT(IDRESERVA) FROM RESERVA WHERE HORACONFIRMACION='NO CONFIRMADA';
```