

Universidade de Vigo

ESCOLA SUPERIOR DE ENXEÑARÍA INFORMÁTICA

Memoria do Traballo de Fin de Grao que presenta

D. Oscar Lista Rivera

para a obtención do Título de Graduado en Enxeñaría Informática

UVigoRes: Sistema de reserva de recursos



Setembro, 2024

Traballo de Fin de Grao N°: EI 23/24-64

Titor/a: Pedro Cuesta Morales

Cotitor/a: Arturo José Méndez Penín

Área de coñecemento: Linguaxes e Sistemas Informáticos

Departamento: Informática

ÍNDICE

Contenido

1	Introducción	1
2	Objetivos	1
3	Solución propuesta	2
4	Planificación y seguimiento	4
4.1	Planificación inicial	4
4.2	Seguimiento temporal	5
4.3	Diferencia entre la planificación inicial y el seguimiento temporal	7
5	Arquitectura.....	8
6	Tecnologías e integración de productos de terceros.....	9
6.1	Tecnologías empleadas	9
6.2	Herramientas de desarrollo	9
7	Especificación y análisis de requisitos	10
7.1	Actores	10
7.2	Requisitos del sistema.....	10
7.2.1	Requisitos funcionales	10
7.2.2	Requisitos no funcionales	11
7.2.3	Requisitos de interfaz	11
7.3	Diagramas de casos de uso general.....	12
7.4	Descripción de los casos de uso.....	13
8	Diseño de software	13
8.1	Diseño estático	13
8.1.1	Diagrama de clases	14
8.2	Diseño dinámico	14
9	Gestión de datos e información	17
9.1	Diseño lógico (MERE)	17
9.2	Diccionario de datos	18
9.3	Diseño físico	21
10	Pruebas	22
10.1	Pruebas de caja negra.....	23
10.2	Pruebas de caja blanca	27
11	Manual de usuario	28
11.1	Requisitos mínimos.....	28
11.2	Instalación	28

UVigoRes: Sistema de reserva de recursos

11.2.1	Descarga e instalación de XAMPP.....	28
11.2.2	Ubicación del proyecto.....	29
11.2.3	Importación de los datos.....	29
11.3	Ingreso al sistema.....	30
11.3.1	Vistas del administrador	32
11.3.2	Vistas del Personal de conserjería y Becario de infraestructura.....	44
11.3.3	Vistas de docente y estudiante.....	45
12	Principales aportaciones	49
13	Conclusiones.....	49
13.1	Conclusiones técnicas	49
13.2	Conclusiones personales	49
14	Vías de trabajo futuro	50
15	Referencias	50

ÍNDICE DE ILUSTRACIONES

Ilustración 1. Esquema MVC	3
Ilustración 2. Diagrama de Gantt.....	7
Ilustración 3. Arquitectura del sistema	8
Ilustración 4. Modelo de casos de uso	12
Ilustración 5. Diagrama de clases	14
Ilustración 6. DSS de añadir usuario	15
Ilustración 7. DSS de añadir recurso	16
Ilustración 8. DSS de reservar recurso	17
Ilustración 9. MERE.....	17
Ilustración 10. Diseño físico de datos.....	22
Ilustración 11. Panel de control de XAMPP	29
Ilustración 12. Vista de la página del login	29
Ilustración 13. Vista de la página de phpMyAdmin.....	30
Ilustración 14. Vista del login de la aplicación	31
Ilustración 15. Vista del menú principal del administrador.....	32
Ilustración 16. Vista de la gestión de usuarios	33
Ilustración 17. Vista del formulario de añadir usuario	34
Ilustración 18. Mensaje de usuario agregado correctamente.....	34
Ilustración 19. Vista de los datos de un usuario	35
Ilustración 20. Menú de editar usuario	35
Ilustración 21. Vista del formulario para editar los datos del usuario	36
Ilustración 22. Vista del formulario para cambiar la contraseña.....	36
Ilustración 23. Mensaje de error si las contraseñas no coinciden.....	37
Ilustración 24. Vista de la gestión de centros	37
Ilustración 25. Vista del formulario de añadir centro.....	38
Ilustración 26. Vista del formulario de editar centro.....	38
Ilustración 27. Vista de gestión de recursos	39
Ilustración 28. Vista del formulario de añadir recurso	40
Ilustración 29. Vista del formulario de editar recurso	40
Ilustración 30. Vista de reservas del día	41
Ilustración 31. Vista del histórico de reservas	41
Ilustración 32. Vista de todas las incidencias	42
Ilustración 33. Vista del formulario de añadir incidencia	42
Ilustración 34. Vista donde se ven los datos de la incidencia.....	43
Ilustración 35. Vista de asignar incidencia.....	43
Ilustración 36. Vista del menú de personal de conserjería	44
Ilustración 37. Vista del menú de becario de infraestructura	44
Ilustración 38. Vista de mis incidencias asignadas.....	45
Ilustración 39. Vista de docente y estudiante	45
Ilustración 40. Vista donde se muestran las reservas del día	46
Ilustración 41. Vista del histórico de reservas	46
Ilustración 42. Vista donde se selecciona el centro	47
Ilustración 43. Vista donde se selecciona el tipo de recurso	47
Ilustración 44. Vista donde se selecciona el recurso	48
Ilustración 45. Vista donde se selecciona la franja horaria	48
Ilustración 46. vista de mensaje de confirmación de reserva	48

ÍNDICE DE TABLAS

Tabla 1. Planificación inicial del proyecto	5
Tabla 2. Seguimiento temporal del proyecto	6
Tabla 3. Diccionario de datos de la tabla "Usuario"	18
Tabla 4. Diccionario de datos de la tabla "Centro"	18
Tabla 5. Diccionario de datos de la tabla "Franja"	19
Tabla 6. Diccionario de datos de la tabla "Recurso"	19
Tabla 7. Diccionario de datos de la tabla "Reserva"	19
Tabla 8. Diccionario de datos de la tabla "Centro_Becario"	20
Tabla 9. Diccionario de datos de la tabla "Centro_Conserje"	20
Tabla 10. Diccionario de datos de la tabla "Incidencia"	20
Tabla 11. Diccionario de datos de la tabla "Incidencia Asignada"	21
Tabla 12. Pruebas de caja negra del formulario de login	23
Tabla 13. Pruebas de caja negra del formulario de añadir usuario	24
Tabla 14. Pruebas de caja negra en el formulario de editar usuario	25
Tabla 15. Pruebas de caja negra en el formulario de cambiar contraseña	25
Tabla 16. Pruebas de caja negra del formulario de editar y añadir centro	26
Tabla 17. Pruebas de caja negra del formulario añadir y editar recurso	26
Tabla 18. Pruebas de caja blanca del formulario de login	27
Tabla 19. Pruebas de caja blanca de los formularios añadir y editar usuario	27

1 Introducción

En el entorno universitario, la gestión eficiente de recursos es fundamental para garantizar el buen funcionamiento de las actividades académicas y administrativas.

Con el incremento del número de estudiantes, docentes y personal administrativo en los centros universitarios, se hace imprescindible contar con sistemas de gestión que faciliten la administración y el uso adecuado de recursos como aulas, laboratorios, equipos audiovisuales, y otros materiales.

Actualmente, en muchos centros universitarios, la gestión de estos recursos se realiza de manera manual o mediante sistemas que no están totalmente integrados, lo que puede generar confusiones, reservas duplicadas, y una administración ineficaz de los recursos disponibles. Este escenario no solo dificulta el aprovechamiento óptimo de los recursos, sino que también puede afectar negativamente la experiencia educativa y la eficiencia administrativa.

Con el objetivo de mejorar esta situación, este Trabajo de Fin de Grado (TFG) propone el desarrollo de una aplicación web que facilite la gestión, consulta de disponibilidad y reserva de recursos en los centros universitarios de la Universidad de Vigo. La aplicación permitirá a los usuarios, incluyendo estudiantes, docentes, becarios de infraestructura, personal de conserjería y administradores, interactuar de manera eficiente con el sistema de reservas, adaptándose a las necesidades específicas de cada centro.

A través de esta plataforma, se busca optimizar la gestión de recursos, promover un uso más racional y organizado de los mismos, y proporcionar una herramienta intuitiva que simplifique tanto la administración como la experiencia del usuario.

La introducción debería incorporar la revisión de algún sistema parecido, para evaluar o justificar la necesidad de otra solución

2 Objetivos

El objetivo de este Trabajo de Fin de Grado (TFG), como ya se menciona anteriormente, es desarrollar una aplicación que facilite la gestión, consulta de disponibilidad y reserva de recursos en un centro universitario de la Universidad de Vigo.

A través de una página web, los usuarios registrados (estudiantes y docentes) podrán consultar la disponibilidad de recursos y realizar reservas. Estos recursos incluyen, entre otros, ordenadores portátiles, seminarios, aulas de teoría, laboratorios de informática, salas de juntas y salones de actos.

Los centros universitarios podrán crear diversos tipos de recursos según sus necesidades específicas, como cámaras de vídeo en la Facultad de Bellas Artes o laboratorios de química en la Facultad de Química. Las reservas estarán sujetas a restricciones determinadas por cada centro, especificando qué usuarios pueden reservar ciertos

recursos. Las reservas se realizarán en franjas horarias establecidas por cada centro universitario.

Además de gestionar las reservas, los usuarios podrán reportar incidencias relacionadas con las mismas y éstas serán atendidas por los becarios de infraestructura o el personal de conserjería del centro. Para asegurar el uso adecuado de los recursos, el personal de conserjería verificará si el usuario hizo uso del recurso y facilitará las llaves o el equipo necesario.

La aplicación será genérica y adaptable a los distintos centros universitarios de la Universidad de Vigo, permitiendo su uso tanto por estudiantes, docentes, becarios, personal de conserjería y administradores.

Una vez explicado el funcionamiento, veamos quién utilizará la aplicación.

Tenemos cinco usuarios: Estudiante, Docente, Becario de infraestructura, Personal de conserjería y Administrador.

- **Estudiante:** consulta y reserva recursos; puede crear una incidencia en un recurso.
- **Docente:** consulta y reserva recursos; puede crear una incidencia en un recurso.
- **Becario de infraestructura:** atiende las incidencias reportadas.
- **Personal de conserjería:** facilita los recursos y confirma su devolución; atiende las incidencias reportadas.
- **Administrador:** gestiona los recursos (crear, consultar, modificar, eliminar).

Todos los usuarios deben estar registrados en el sistema.

3 Solución propuesta

Se ha desarrollado una aplicación web programada en PHP, HTML, CSS y JavaScript. También se ha empleado la tecnología de MySQL.

Justificar de algún modo el porqué de haber elegido el Proceso Unificado

La arquitectura sigue el Modelo-Vista-Controlador (MVC) [1] explicado posteriormente en detalle y cuyo funcionamiento se muestra de forma simplificada en la siguiente imagen:

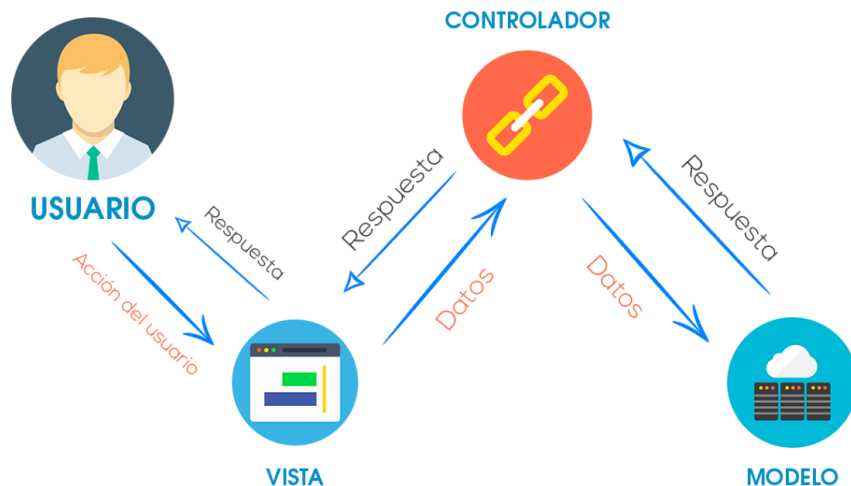


Ilustración 1. Esquema MVC

Las tecnologías usadas son PHP para la parte de backend y HTML y PHP con CSS para la parte del frontend.

Para la gestión de datos se ha utilizado MySQL junto con la herramienta para la gestión de los datos phpMyAdmin.

En lo referente a la metodología se ha optado por el Proceso Unificado (PU), un marco de desarrollo de software que se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental.

A continuación, se detalla un poco más estas características:

- **Dirigido por casos de uso:** los casos de uso reflejan lo que los usuarios necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. Los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen representan la realización de los casos de uso.
- **Centrado en la arquitectura:** la arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, o sea, describe los elementos del modelo que son más importantes para su construcción. El modelo de arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML.
- **Iterativo e Incremental:** una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros.

El PU se desarrolla de manera cíclica, donde cada ciclo se divide en cuatro fases: Inicio, Elaboración, Construcción y Transición. **EN VEZ DE GENERALIDADES, DECIR LO QUE HICE EN CADA FASE**

- **Inicio:** en la fase de inicio se define el negocio, la facilidad de realizar el proyecto, se presenta un modelo, visión, metas, deseos del usuario, plazos, costos y viabilidad.

- **Elaboración:** en esta fase se obtiene la visión refinada del proyecto a realizar, la implementación iterativa del núcleo de la aplicación, la resolución de riesgos altos, nuevos requisitos y se ajustan las estimaciones
- **Construcción:** esta abarca la evolución hasta convertirse en producto listo incluyendo requisitos mínimos. Aquí se afinan los detalles menores como los diferentes tipos de casos o los riesgos menores.
- **Transición:** en esta fase final, el programa debe estar listo para ser probado, instalado y utilizado por el cliente sin ningún problema. Una vez finalizada esta fase, se debe comenzar a pensar en futuras novedades para la misma.

4 Planificación y seguimiento

La planificación del implica la organización y estructuración de todas las actividades necesarias para completar el trabajo. Esta fase es crucial porque establece las bases para el desarrollo del proyecto y asegura que se cumplan los plazos y objetivos.

El seguimiento implica el control y la supervisión del progreso del trabajo para asegurarse de que se esté desarrollando según lo planificado y para hacer ajustes cuando sea necesario. Esta fase es importante para identificar problemas a tiempo y mantener el trabajo en el camino correcto.

El TFG tiene una duración estimada de 300 horas (correspondiente a 12 créditos ECTS). A continuación, se muestra la planificación del proyecto; tanto la esperada y planificada, como la real.

4.1 Planificación inicial

Para elaborar la planificación inicial del proyecto, se ha acordado la duración de cada actividad considerando la disponibilidad para llevarlas a cabo y estableciendo un plazo límite con el fin de cumplir con los objetivos establecidos en el tiempo previsto.

Inicialmente, se tenía previsto comenzar este proyecto a principios de enero de 2024. Sin embargo, por motivos docentes y laborales no he comenzado hasta principios de febrero de 2024 con las primeras reuniones, pero el análisis comenzó el día 15 de mayo de 2024 con la elaboración de la primera descripción del problema. De este modo, se estimó que la duración del proyecto abarcaría desde el 15 de julio de 2024 (fecha de la realización de la descripción y objetivos del problema) hasta el 31 de agosto de 2024. Por consiguiente, la planificación inicial contempló una duración de aproximadamente 108 días (alrededor de 15 semanas). A continuación, se presenta una tabla que resume las actividades planificadas junto con sus duraciones estimadas, las cuales se describen con mayor detalle en el apartado siguiente.

Fase	Actividad	Tiempo estimado (horas)
Inicio	Definición del alcance y objetivos	10
	Identificación de actores y casos de uso	20
Elaboración	Refinar requisitos	15
	Desarrollo arquitectura	20

	Planificar la construcción y pruebas	40
Construcción	Primera iteración	55
	Segunda iteración	45
	Tercera iteración	30
	Versión final del proyecto	20
Transición	Documentación	15
	Versión final del proyecto	30
	Total	300

Tabla 1. Planificación inicial del proyecto

4.2 Seguimiento temporal

El desarrollo del proyecto ha transcurrido de forma organizada y satisfactoria. Sin embargo, en algunas ocasiones, debido a motivos laborales, así como por razones personales, la diferencia entre la duración real de cada actividad y la duración estimada se incrementó. A continuación, se presenta una tabla que muestra un desglose más detallado de las actividades (en comparación con la Tabla 1) y una comparación entre el tiempo inicialmente estimado y el tiempo real. Además, se incluye una columna que indica la fase del proceso unificado a la que pertenece cada actividad.

Fase	Actividad	Descripción	Tiempo estimado (horas)	Tiempo real (horas)
Inicio	Definición del alcance y objetivos	Definir claramente los objetivos y el alcance del proyecto	5	5
	Identificación de actores y casos de uso	Identificar a los usuarios del sistema y sus interacciones	5	5
	Desarrollo del caso de negocio	Elaborar el caso de negocio justificando el proyecto	5	5
	Planificación inicial del proyecto	Desarrollar un plan inicial para el proyecto	10	10
	Revisión y aprobación de la fase de Inicio	Revisar y aprobar los entregables de la fase de inicio	5	5
Elaboración	Refinamiento de los requisitos	Detallar los requisitos del sistema	15	15
	Desarrollo de la arquitectura del sistema	Diseñar la arquitectura del sistema	20	20
	Planificación de la construcción	Planificar las actividades de construcción del sistema	20	20

UVigoRes: Sistema de reserva de recursos

	Pruebas y evaluación del prototipo	Probar y evaluar el prototipo inicial del sistema	10	10
	Revisión y aprobación de la fase de elaboración	Revisar y aprobar los entregables de la fase de elaboración	10	10
Construcción	Primera iteración	Gestión de usuarios	20	20
		Gestión de centros	15	15
		Gestión de recursos	20	20
	Segunda iteración	Reserva recursos	20	20
		Gestionar acceso	15	15
		Crear incidencia de recurso	15	15
	Tercera iteración	Gestionar incidencias	15	20
		Atender incidencias de recurso	15	15
	Versión final del proyecto	Pruebas unitarias	15	15
		Preparar el sistema para su implementación	5	5
Transición	Desarrollo de la documentación	Gestionar incidencias	10	15
	Implementación y despliegue	Capacitar a los usuarios y brindar soporte inicial	10	10
	Prueba de aceptación y ajustes finales	Revisar el proyecto y formalizar su cierre	10	10
	Corrección de la documentación	Solucionar errores en la documentación	5	5
	Revisión final y cierre del proyecto	Revisar el proyecto y formalizar su cierre.	5	5
			300	310

Tabla 2. Seguimiento temporal del proyecto

En este diagrama de Gantt se puede ver el desarrollo de todas las tareas realizadas.

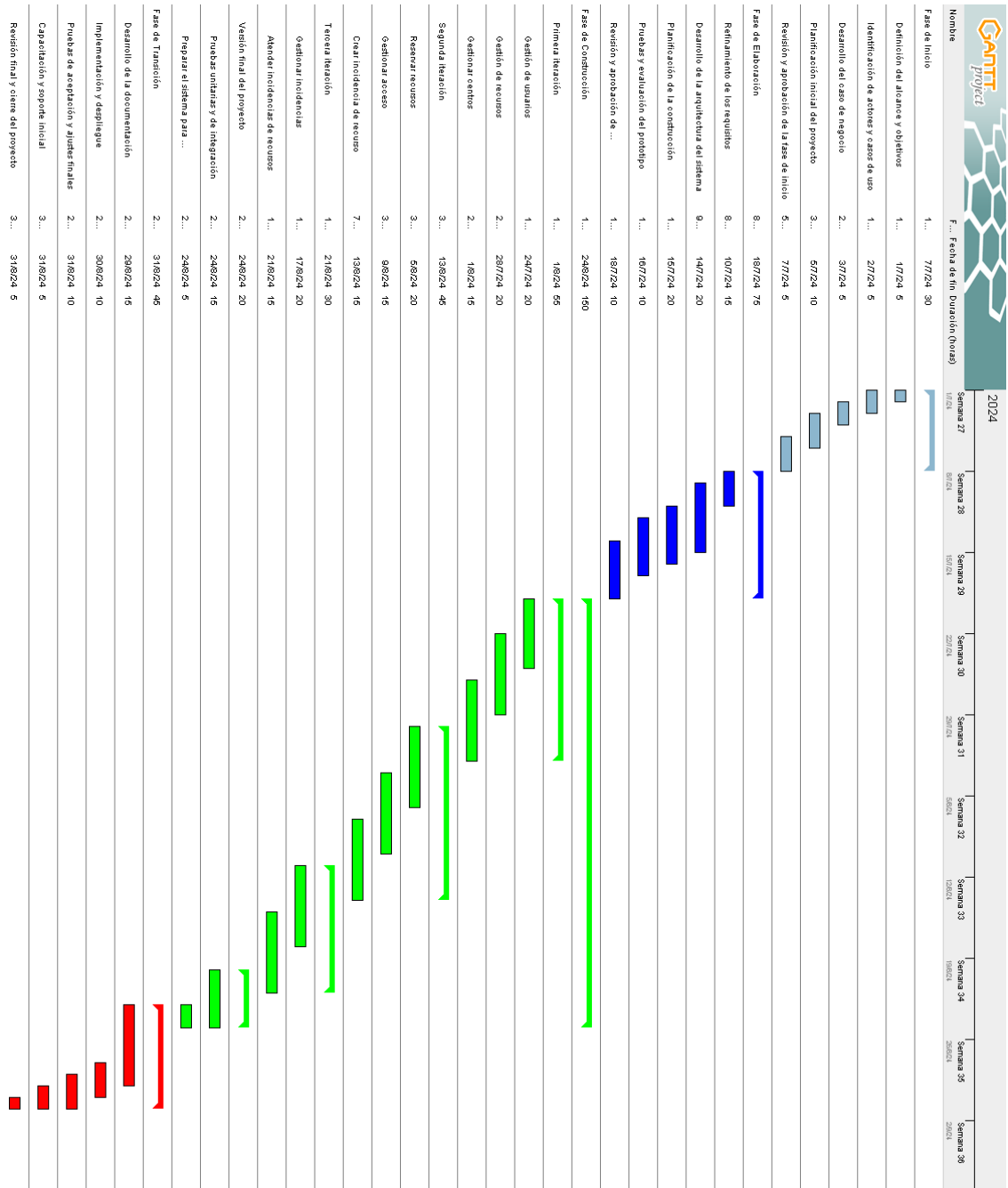


Ilustración 2. Diagrama de Gantt

4.3 Diferencia entre la planificación inicial y el seguimiento temporal

En la tabla anterior, se puede observar una comparación entre el tiempo estimado para cada actividad y su duración real (reflejadas en las dos últimas columnas de la tabla). El tiempo total estimado era de 300 horas, mientras que el tiempo real empleado en el

proyecto ha sido de 310 horas, es decir, 10 horas adicionales. Estas horas extra se deben principalmente a retrasos en la implementación, especialmente en la lógica de las restricciones de reservas y en la asignación de incidencias. En cuanto al total de días, la planificación inicial preveía que el proyecto se completaría el 31 de agosto de 2024, pero debido a las 10 horas adicionales mencionadas, así como a los problemas personales y académicos que surgieron a lo largo del desarrollo del proyecto, la finalización se produjo el 02 de septiembre de 2024, unos 2 días después de lo previsto.

5 Arquitectura

El sistema desarrollado consiste en una aplicación web diseñada para que los usuarios puedan reservar y consultar recursos en un centro universitario. La arquitectura utilizada en esta aplicación sigue el patrón Modelo-Vista-Controlador (MVC) orientado a objetos, comúnmente empleado en aplicaciones de escritorio. Este enfoque facilita la separación entre la lógica de los datos y su representación visual, permitiendo que un mismo conjunto de datos pueda ser presentado en diferentes formatos (como HTML, entre otros). A continuación, se muestra un esquema general del sistema desarrollado.

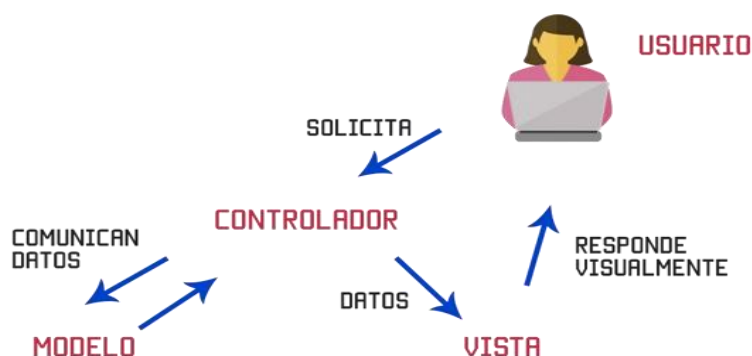


Ilustración 3. Arquitectura del sistema

El Modelo-Vista-Controlador (MVC) es un patrón de arquitectura de software que organiza una aplicación en tres componentes principales:

- **Modelo:** representa la lógica de negocio y los datos de la aplicación. Es responsable de gestionar el estado y la lógica de la aplicación, así como de interactuar con la base de datos. El modelo no está directamente relacionado con la interfaz de usuario.
- **Vista:** es la parte de la aplicación que interactúa con el usuario. Muestra los datos del modelo al usuario y envía las acciones del usuario al controlador. La vista se encarga de la presentación de la información, como páginas web HTML o interfaces gráficas.
- **Controlador:** actúa como intermediario entre el modelo y la vista. Recibe las entradas del usuario desde la vista (como clics o formularios enviados), procesa esa información (interactuando con el modelo si es necesario) y determina qué vista mostrar a continuación.

Este patrón permite una separación clara entre la lógica de negocio, la interfaz de usuario y la gestión de la interacción, lo que facilita el mantenimiento y la escalabilidad de la aplicación.

6 Tecnologías e integración de productos de terceros

En este apartado se describen adecuadamente las tecnologías utilizadas para el desarrollo del trabajo, justificando su utilización.

6.1 Tecnologías empleadas

A continuación, se detallan las tecnologías que se han utilizado:

- **HTML** (Hyper Text Markup Language): lenguaje de marcado predominante para la creación de páginas web estáticas. Basado en el empleo de etiquetas. Se ha optado por su uso por su facilidad de manejo, curva de aprendizaje relativamente baja y el gran soporte online debido a su gran uso.
- **PHP** (Hypertext Preprocessor): para crear la base y gran parte de la funcionalidad del sistema se ha optado por PHP, uno de los lenguajes de desarrollo web más conocidos y usados, además de tener un gran soporte [2].
- **CSS** (Cascade Style Sheet): usadas para la apariencia y estilo de la aplicación Web. Su finalidad es crear unos estilos de diseño. CCS está altamente compenetrado con HTML para decorar las vistas del sistema.
- **SQL** (Structured Query Language): lenguaje de manejo y gestión de base de datos por antonomasia, de ahí que se haya optado por usarlo para gestionar la base de datos del sistema.

6.2 Herramientas de desarrollo

Por otro lado, se detallan las herramientas de desarrollo utilizadas:

- **Chrome y Firefox**: estos navegadores web fueron utilizados para probar la aplicación web debido a su amplia popularidad y velocidad.
- **GitHub**: es un sistema de control de versiones gratuito y de código abierto, elegido por ser el más utilizado en su campo. Se utilizó para gestionar las versiones del proyecto de manera ordenada.
- **Microsoft Word**: software de procesamiento de texto utilizado para redactar este documento, seleccionado por su calidad y mi familiaridad con la herramienta.
- **Visual Studio Code**: editor de texto utilizado para la implementación del código en PHP, HTML y CSS. Dado que ninguno de estos lenguajes requiere compilación, este editor de texto es suficiente para desarrollar el sistema sin la necesidad de un entorno de desarrollo integrado (IDE).
- **Visual Paradigm**: herramienta de diseño de software empleada para la creación de diagramas y otros artefactos durante el desarrollo del proyecto. Fue elegida debido a mi experiencia previa con ella.

- **GanttProject:** una interfaz de usuario sencilla, intuitiva y que permite desglosar el trabajo, construir un diagrama de Gantt, asignar recursos y calcular los costes del proyecto.

7 Especificación y análisis de requisitos

En los procesos de desarrollo de software, los casos de uso se emplean para capturar los requisitos funcionales del sistema. Estos casos de uso describen de manera concisa las necesidades de los usuarios que interactúan con el sistema (actores) y los criterios que determinan cuándo un caso de uso se considera completado.

Al comienzo del proyecto, se identifican todos los usuarios especificando los roles que desempeñan, la funcionalidad que requieren y el valor que esta funcionalidad les proporciona.

7.1 Actores

En el sistema se identifican 5 tipos de actores:

- **Estudiante:** consulta y reserva recursos; puede crear una incidencia en un recurso.
- **Docente:** consulta y reserva recursos; puede crear una incidencia en un recurso.
- **Becario de infraestructura:** atiende las incidencias reportadas.
- **Personal de conserjería:** facilita los recursos y confirma su devolución; atiende las incidencias reportadas.
- **Administrador:** gestiona los recursos (crear, consultar, modificar, eliminar).

7.2 Requisitos del sistema

Los requisitos del sistema se pueden definir como las características principales que puede realizar el sistema.

Se distinguen tres tipos de requisitos en el desarrollo del sistema: requisitos funcionales, requisitos no funcionales y requisitos de interfaz. A continuación, se detallan los identificadores para cada tipo de requisito:

- **RFXX:** identificador de los requisitos funcionales (donde XX es el número asignado al requisito). Estos requisitos se relacionan con las funcionalidades específicas del sistema y responden a la pregunta "¿Qué hace el sistema?".
- **RNFX:** identificador de los requisitos no funcionales (donde XX es el número asignado al requisito). Estos requisitos se refieren a las características del sistema, como rendimiento, integridad y seguridad, y responden a la pregunta "¿Cómo funciona el sistema?".
- **RIFX:** identificador de los requisitos de interfaz (donde XX es el número asignado al requisito). Estos requisitos están relacionados con la funcionalidad de la interfaz de la aplicación, incluyendo aspectos como formatos y tiempos de respuesta.

7.2.1 Requisitos funcionales

A continuación, se explican los distintos requisitos funcionales:

- **RF01:** gestión de usuarios. Los usuarios deben estar registrados en el sistema, y es responsabilidad del Administrador registrar a cada usuario, especificando su tipo.
- **RF02:** gestión de recursos. El Administrador tiene la capacidad de añadir, consultar, modificar y eliminar recursos.
- **RF03:** consultar recursos. Los usuarios registrados tienen la posibilidad de consultar los recursos disponibles según las franjas horarias que les interesen.
- **RF04:** reservar recursos. Los usuarios con perfiles de Estudiante y Docente pueden reservar recursos en las franjas horarias disponibles. Cada centro universitario establece sus propias restricciones para la reserva de recursos y las franjas horarias en las que se pueden realizar las reservas. Una vez que un recurso es reservado, pasa a estar no disponible.
- **RF05:** gestionar acceso. El Personal de conserjería es responsable de controlar el uso de los recursos y facilitar su acceso a los usuarios.
- **RF06:** crear incidencias de recurso. Los usuarios con perfiles de Estudiante, Docente y Personal de conserjería pueden reportar incidencias asociadas a un recurso si encuentran algún fallo. Cuando se crea una incidencia, el recurso afectado se marca como no disponible para su reserva hasta que la incidencia se resuelva.
- **RF07:** gestionar incidencias. El Administrador tiene la tarea de revisar y gestionar las incidencias reportadas en los recursos. Según el tipo de incidencia, esta puede ser asignada a un Becario de infraestructura o al Personal de conserjería para su resolución.
- **RF08:** atender incidencias de recurso. Los Becarios de infraestructura y el Personal de conserjería pueden ver las incidencias reportadas y el estado de cada una. Una vez que una incidencia es resuelta, se marca como tal y el recurso correspondiente vuelve a estar disponible para su reserva.

7.2.2 Requisitos no funcionales

A continuación, se explican los distintos requisitos no funcionales:

- **RNF01:** adaptabilidad y Generalidad. La aplicación debe ser genérica y adaptable a distintos centros de estudios de la Universidad de Vigo. Debe ser utilizada por estudiantes, becarios de infraestructura, personal de conserjería y docentes.
- **RNF02:** tecnologías Gratuitas. El desarrollo de la aplicación se realizará utilizando tecnologías gratuitas.
- **RNF03:** eficiencia y Rendimiento. La aplicación debe tener un rendimiento óptimo para gestionar reservas y recursos sin demoras significativas.
- **RNF04:** seguridad. Se garantizará la seguridad de los datos de los usuarios y las reservas realizadas. Se implementarán medidas de protección contra posibles ataques o intrusiones.
- **RNF05:** escalabilidad. La aplicación debe ser escalable para manejar un aumento potencial en el número de usuarios y recursos gestionados.

7.2.3 Requisitos de interfaz

A continuación, se explican los distintos requisitos de interfaz:

- **RI01:** interfaz intuitiva. La interfaz de nuestro sistema debe ser clara y fácil de usar, permitiendo a los usuarios acceder sin dificultad a las opciones disponibles y sin confusión respecto a los iconos o enlaces de la aplicación.
- **RI02:** compatibilidad con navegadores. El sistema debe funcionar de manera uniforme en los navegadores principales (Google Chrome y Mozilla Firefox). Estos dos navegadores se han elegido por ser los más utilizados.
- **RI03:** acceso a las vistas. El sistema distingue entre los diferentes tipos de usuarios y les proporciona las vistas correspondientes. Un estudiante o docente no podrá visualizar lo que ve un becario de infraestructura o personal de conserjería, y viceversa.

7.3 Diagramas de casos de uso general

Una vez definidos los requisitos funcionales y los actores del sistema, se puede crear el diagrama de casos de uso. Este diagrama muestra, fundamentalmente, los diferentes casos de uso (es decir, las acciones que generan algún resultado para los actores) y cómo los actores interactúan con ellos. El diagrama de casos de uso para UVigoRes se presenta en la imagen a continuación.

- En el diagrama de casos de uso, el Administrador ya puede Gestionar Recursos, por lo tanto, ya incluye lo de Consultar recursos, yo no lo volvería a unir. El actor Usuario supongo que es la superclase de Estudiante y Docente, yo indicaría a estos Actores también con la relación de herencia. En este actor, los casos de uso de Reservar recursos y Crear incidencia de recurso no se relacionan ni con include ni con extend, no se relacionan para nada, piénsalo. Al personal de conserjería le pondría mejor Gestionar acceso al recurso.

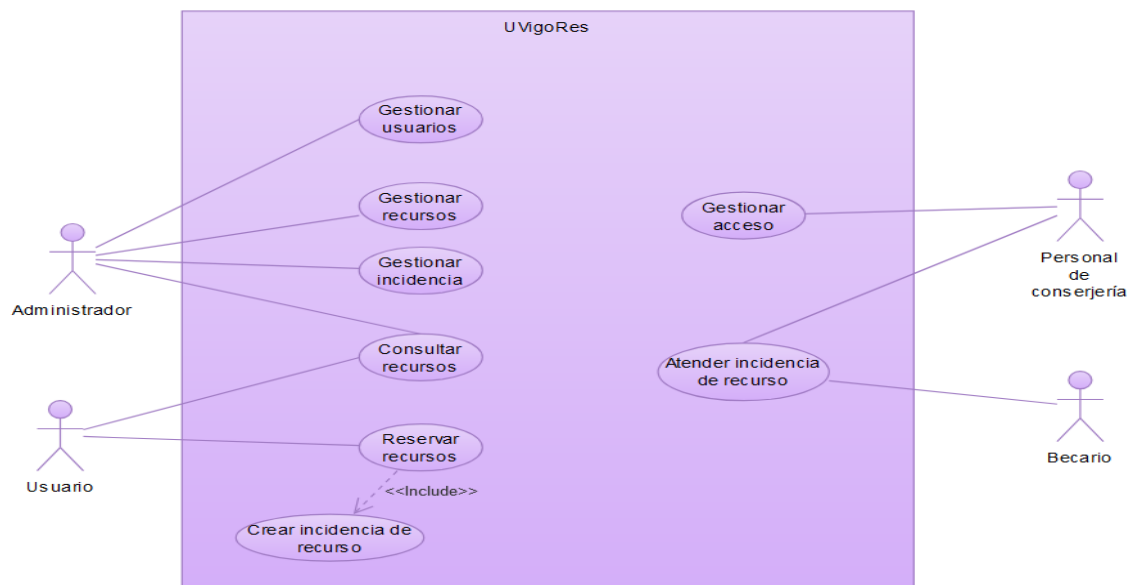


Ilustración 4. Modelo de casos de uso

7.4 Descripción de los casos de uso

Se muestra a continuación las descripciones de los casos de uso relativas a la gestión de recursos. Se ha incluido sólo esto por entender que es el caso de uso más importante en lo referente al sistema. **en el análisis no se describe un caso de uso con palabras como pulsar en, no se dan detalles de diseño. Y faltaría ese modelo de dominio que pusiste en diseño.**

Consultar recurso

El usuario administrador y el conserje pueden consultar todos los recursos desde el menú de “Gestionar recursos”. Allí se muestran todos los recursos del sistema con los datos del tipo descripción, disponibilidad y el centro en donde está el recurso.

Ver recursos

Para ver la información de un recurso en concreto, el usuario administrador o conserje, sólo tiene que ir al recurso que quiere consultar y en él pulsar en “Ver recurso”. Se mostrará el tipo de recurso, la descripción, la disponibilidad y en qué centro está el recurso.

Añadir recurso

El administrador pulsa en “Añadir recurso” y se le muestra un formulario con los datos del tipo, descripción y dos desplegables: uno de si el recurso está disponible y otro del centro en donde se va a añadir ese recurso. El administrador rellena los datos y confirma la inserción. El recurso se crea e insertan en la base de datos.

Editar recuso

Los recursos se pueden editar. Para ello, el administrador solo tiene que pulsar en “Editar recurso” y se le muestra un formulario con los datos del recurso en donde el usuario solo tiene que cambia los datos de tipo, descripción, disponibilidad y centro si los desea. Los datos de disponibilidad y centro son dos desplegables que recuperan las opciones disponibles. Una vez hecho los cambios deseados el usuario solo tiene que pulsar en “Guardar Cambios” y los datos del recurso se actualizarán en la base de datos.

Eliminar recurso **EXPLICAR**

DESCRIBE MÁS, P.E. CÓMO SE GESTIONAN LAS INCIDENCIAS

8 Diseño de software

8.1 Diseño estático

El diseño estático de software se enfoca en analizar y estructurar el software sin necesidad de ejecutar el programa. Este tipo de diseño examina la estructura de las clases (un grupo de objetos que tienen atributos y operaciones similares) y los objetos que componen el

sistema. Se centra particularmente en la representación de esas clases y objetos, así como en su comportamiento, las relaciones entre ellos y otros elementos del sistema.

8.1.1 Diagrama de clases

Los diagramas de clases son uno de los tipos de diagramas más valiosos en UML, ya que representan de manera clara la estructura de un sistema específico. Estos diagramas modelan las clases del sistema, junto con sus atributos, operaciones y las relaciones que existen entre los distintos objetos. **ESTÁ MAL**

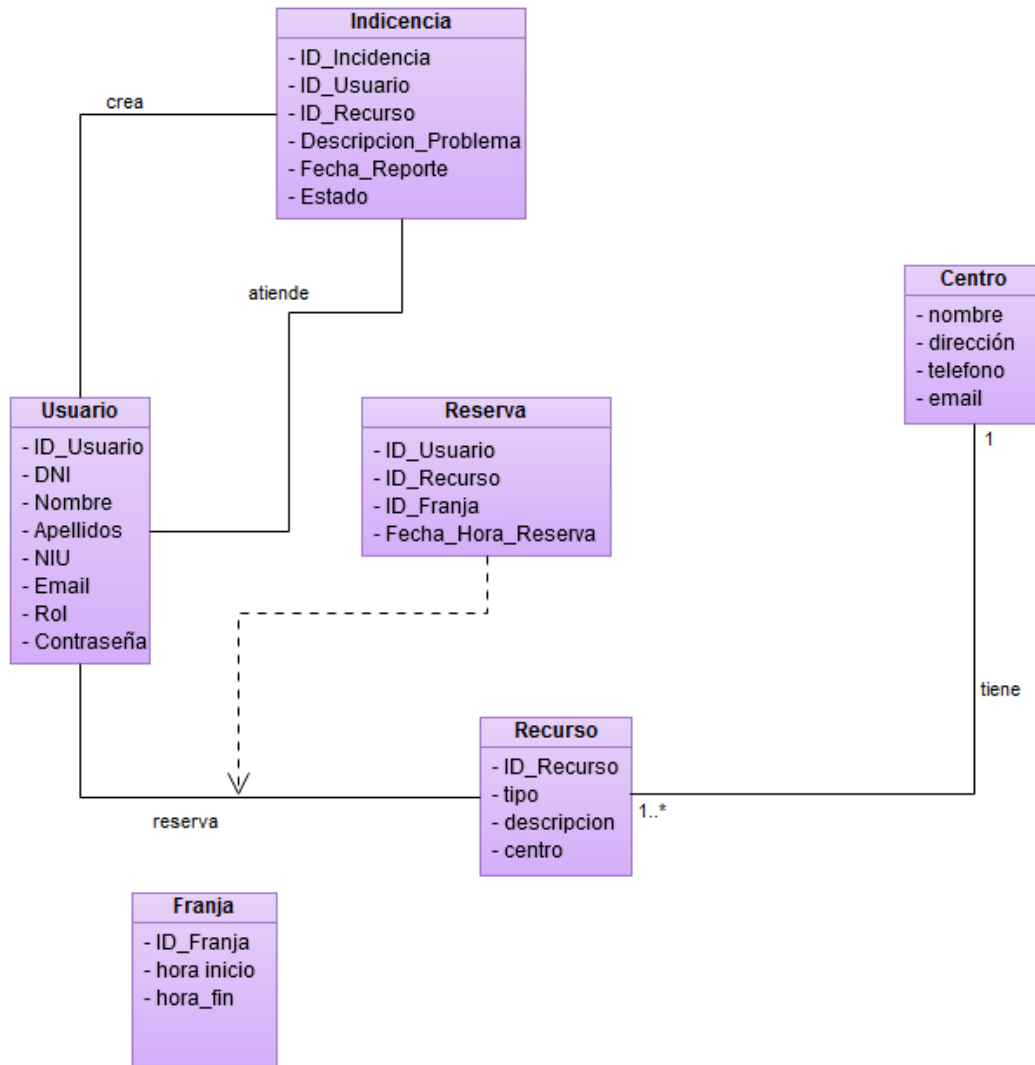


Ilustración 5. Diagrama de clases

8.2 Diseño dinámico

El diseño dinámico de software se enfoca en describir cómo se comporta un sistema a lo largo del tiempo. Los diagramas de secuencia del sistema (DSS) son herramientas eficaces para representar este tipo de diseño, ya que muestran cómo interactúan los diferentes componentes del sistema en distintos momentos. A continuación, se presentan los DSS correspondientes a las principales funcionalidades del sistema, como el añadir usuarios, la creación de recursos y la reserva recurso. Estas funciones se han seleccionado

porque son las más representativas, y las demás acciones no se muestran para evitar redundancias, ya que son muy similares a las mencionadas.

"Los diagramas de secuencia del sistema (DSS)" no son los que estás dibujando, los DSS sólo incluyen el Actor frente al Sistema, y forman parte del análisis, tú estás pintando un DS sin más, un diagrama de secuencia. Todos los objetos deben estar subrayados, y la Base de Datos si la representas como un objeto más también lleva los dos puntos. De todos modos no creo que lo estés explicando bien, seguramente creas varias vistas en cada caso de uso, y a lo mejor accedes a más de un elemento del modelo. ¿Cuando seleccionas añadir un usuario no se genera una nueva vista con el formulario, por ejemplo? ¿Y el mensaje de éxito va desde la base de datos a la vista o va al controlador, o es una nueva vista? En el DS de añadir recurso no hay elemento Recurso, creo que te has equivocado. El DS Reservar recurso parece mejor, pero no tienes que poner todas las vistas a la izquierda, en UML se ponen los objetos a medida que interaccionan. ¿Qué es la línea discontinua después del Controlador?

- Añadir usuario

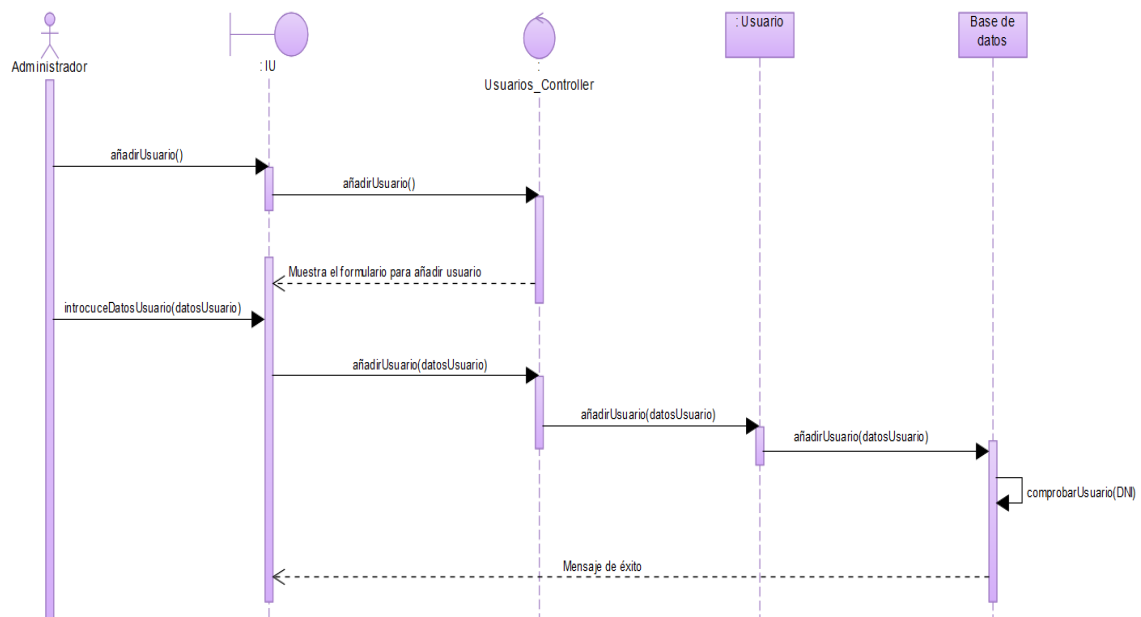


Ilustración 6. DSS de añadir usuario

- Añadir recurso

UVigoRes: Sistema de reserva de recursos

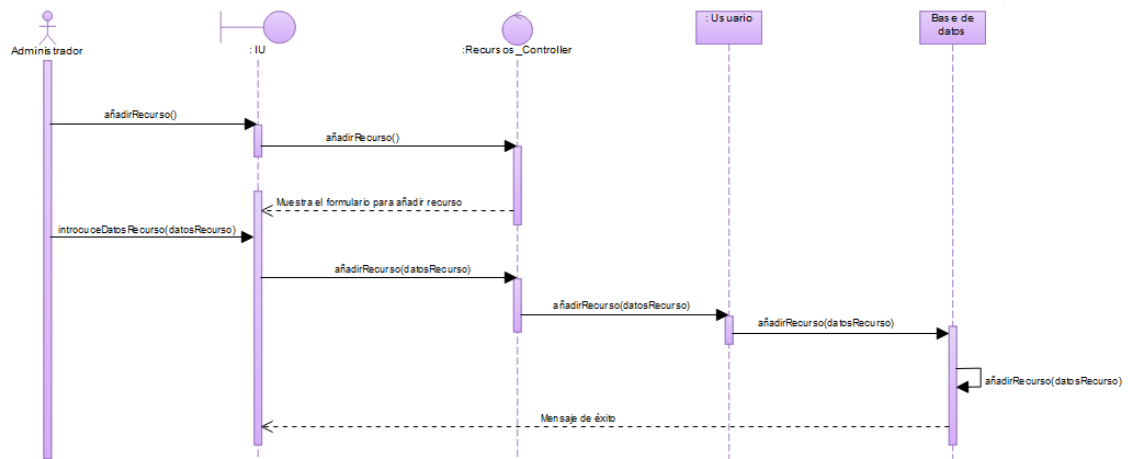


Ilustración 7. DSS de añadir recurso

- Reservar recurso

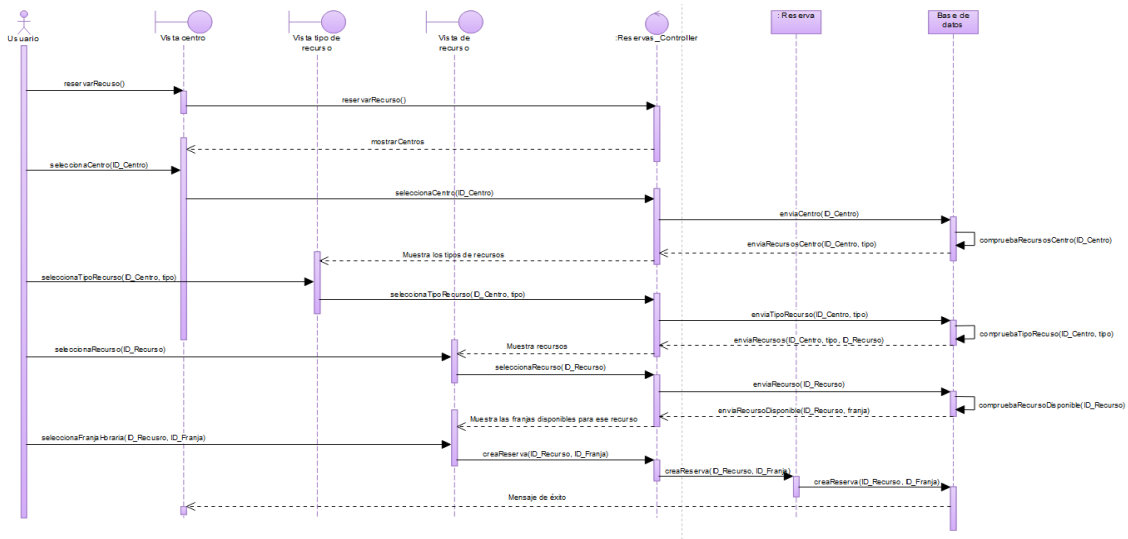


Ilustración 8. DSS de reservar recurso

9 Gestión de datos e información

En esta sección se detalla toda la información que gestiona la aplicación, con el objetivo de proporcionar a los usuarios la capacidad de administrar las funcionalidades disponibles en la plataforma.

9.1 Diseño lógico (MERE)

A continuación, se muestra modelo Entidad-Relación (MERE) de los datos de la aplicación. **AÑADIR CAMPOS A MERE**

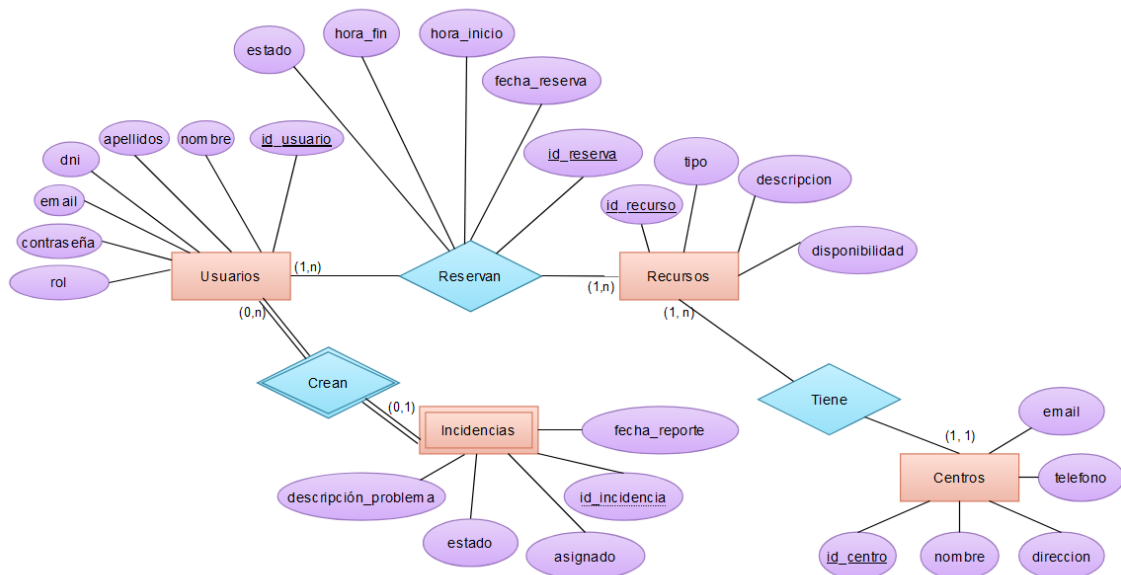


Ilustración 9. MERE

9.2 Diccionario de datos

A continuación, se muestran las tablas con las características de los datos del sistema.

- Usuarios

Nombre de la tabla: **usuario**

Descripción: tabla donde se almacenan los usuarios.

Campo	Tipo de dato	Descripción
ID_Usuario	Int	ID del usuario
DNI	Varchar	DNI del usuario
Nombre	Varchar	Nombre del usuario
Apellidos	Varchar	Apellidos del usuario
NIU	Varchar	NIU del usuario
Email	Varchar	Email del usuario
Rol	Enumerado	Tipo de usuario: Estudiante, Docente, Becario de infraestructura, Personal de conserjería o Admin
Contraseña	Varchar	Contraseña del usuario

Tabla 3. Diccionario de datos de la tabla "Usuario"

Campo clave: ID_Usuario

- Centros

Nombre de la tabla: **centro**

Descripción: tabla donde se almacenan los usuarios.

Campo	Tipo de dato	Descripción
ID_Centro	Int	ID del centro
Nombre	Varchar	Nombre del centro
Direccion	Varchar	Dirección del centro
Telefono	Varchar	Teléfono del centro
Email	Varchar	Email del centro

Tabla 4. Diccionario de datos de la tabla "Centro"

Campo clave: ID_Centro

- Franja

Nombre de la tabla: **franja**

Descripción: tabla donde se almacenan las franjas horarias de las reservas.

Campo	Tipo de dato	Descripción
-------	--------------	-------------

UVigoRes: Sistema de reserva de recursos

ID_Franja	Int	ID de la franja
Hora_Inicio	time	Hora de inicio de la franja
Hora_Fin	time	Hora de fin de la franja

Tabla 5. Diccionario de datos de la tabla "Franja"

Campo clave: ID_Franja

- Recurso

Nombre de la tabla: **recurso**

Descripción: tabla donde se almacenan los recursos.

Campo	Tipo de dato	Descripción
ID_Recurso	Int	ID del recurso
Tipo	Varchar	Hora de inicio de la franja
Descripcion	Text	Hora de fin de la franja
Disponibilidad	Enumerado	Estado del recurso: Disponible o No disponible
ID_Centro	Int	ID del centro al que pertenece el recurso

Tabla 6. Diccionario de datos de la tabla "Recurso"

Campo clave: ID_Recurso

- Reserva

Nombre de la tabla: **reserva**

Descripción: tabla donde se almacenan las reservas realizadas

AÑADIR FILA HORA DISFRUTADA

Campo	Tipo de dato	Descripción
ID_Reserva	Int	ID de la reserva
ID_Usuario	Int	ID del usuario que hizo la reserva
ID_Recurso	Int	ID del recurso
Fecha_Hora_Reserva	Datetime	Fecha y hora cuando se hizo la reserva
ID_Franja	Int	ID de la franja de hora reservada
Estado	Enumerado	Estado de la reserva: Confirmada o No confirmada

Tabla 7. Diccionario de datos de la tabla "Reserva"

Campo clave: ID_Reserva

- Centro Becario

Nombre de la tabla: **centro_becario**

Descripción: tabla donde se almacenan los becarios asignados a los centros

Campo	Tipo de dato	Descripción
ID_Centro	Int	ID del centro
ID_Usuario	Int	ID del usuario con Rol de tipo Becario

Tabla 8. Diccionario de datos de la tabla "Centro_Becario"

Campos clave: ID_Centro e ID_Usuario

- **Centro Conserje**

Nombre de la tabla: **centro_conserje**

Descripción: tabla donde se almacenan los conserjes asignados a los centros

Campo	Tipo de dato	Descripción
ID_Centro	Int	ID del centro
ID_Usuario	Int	ID del usuario con Rol de tipo Conserje

Tabla 9. Diccionario de datos de la tabla "Centro_Conserje"

Campos clave: ID_Centro e ID_Usuario

- **Incidencia**

Nombre de la tabla: **incidencia**

Descripción: tabla donde se almacenan las incidencias

Campo	Tipo de dato	Descripción
ID_Incidencia	Int	ID del centro
ID_Usuario	Int	ID del usuario que realizó la incidencia
ID_Recurso	Int	ID del recurso con la incidencia
Descripcion_Problema	Text	Explicación de la incidencia
Fecha_Reporte	Datetime	Fecha de la creación de la incidencia
Estado	Enumerado	Si la incidencia está en estado Pendiente o Resuelta
Asignada	Tinyint	Si está asignada la incidencia o no

Tabla 10. Diccionario de datos de la tabla "Incidencia"

Campos clave: ID_Incidencia

- **Incidencia Asignada**

Nombre de la tabla: **incidencia_asignacion**

No hay descripción??

Campo	Tipo de dato	Descripción
ID_Incidencia	Int	ID de la incidencia
ID_Usuario	Int	ID del usuario asignado a la incidencia
Fecha_Asignacion	Datetime	Fecha de asignación de la incidencia

Tabla 11. Diccionario de datos de la tabla "Incidencia Asignada"

Campos clave: ID_Incidencia e ID_Usuario

9.3 Diseño físico

En los apartados anteriores se ha detallado el esquema lógico para la gestión de datos del sistema (MERE) y se ha descrito utilizando el diccionario de datos. Es esencial comprender que el diseño lógico de una base de datos refleja cómo modelamos la realidad, representando las entidades que necesitamos almacenar, como clientes, salas, o locales. En contraste, el diseño físico se enfoca en la eficiencia operativa de la base de datos, optimizando su desempeño y evitando duplicaciones e inconsistencias de información. Por esta razón, el diseño físico incluye todas las tablas de la base de datos, ilustrando su estructura interna.

La imagen siguiente muestra el diseño físico de la base de datos para la aplicación UVigoRes.

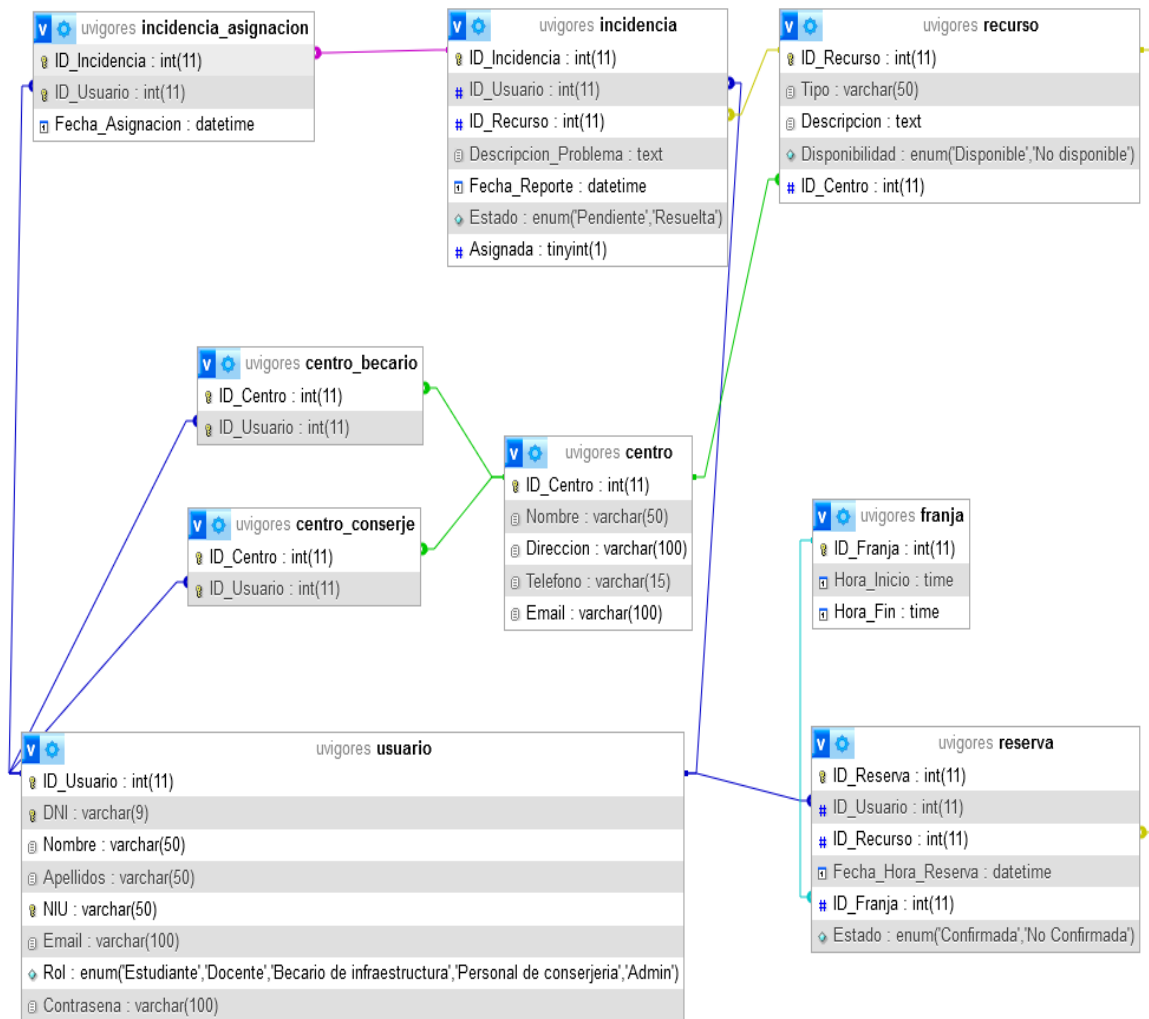


Ilustración 10. Diseño físico de datos

10 Pruebas

Las pruebas consisten en ejecutar un sistema o componente bajo condiciones o requisitos específicos, para evaluar algún aspecto del mismo. El propósito de estas pruebas es identificar y documentar los defectos que puedan comprometer la calidad del software, asegurar que el software funcione según lo previsto, verificar que se cumplan los requisitos establecidos y confirmar que estos requisitos se han implementado correctamente.

Es importante destacar que estas pruebas no pueden garantizar la ausencia de fallos; su objetivo principal es identificar cualquier defecto presente en el software. Cada prototipo que se desea entregar al final de cada iteración debe ser sometido a pruebas y evaluaciones.

En este contexto, se han llevado a cabo dos tipos de pruebas: las pruebas de caja negra, que examinan las especificaciones funcionales, y las pruebas de caja blanca, que se centran en las pruebas del camino básico. Estas últimas permiten al diseñador de pruebas medir la complejidad lógica de un diseño procedimental y utilizar esa medida como referencia para definir un conjunto básico de rutas de ejecución.

10.1 Pruebas de caja negra

Las técnicas de caja negra se basan en la experiencia y el conocimiento para identificar posibles problemas que un software podría generar. Al realizar pruebas de caja negra, es importante adoptar la perspectiva de un usuario final y probar las funciones más comunes del sistema. A continuación, se presentan una serie de tablas que contienen las pruebas de caja negra aplicadas a los formularios correspondientes del sistema.

Login

Campo	Entrada	Resultado esperado	Resultado obtenido	Detalles
DNI	12345678Z	Éxito	Éxito	DNI correcto
DNI	12345678A	Fallo	Fallo	Letra DNI incorrecta
DNI	1234567891A	Fallo	Fallo	Más de 9 caracteres
DNI	123456 78A	Fallo	Fallo	DNI con espacios
DNI	123456Z	Fallo	Fallo	Menos de 9 caracteres
Contraseña	Asdf1234	Éxito	Éxito	Caracteres alfanuméricos y al menos una mayúscula
Contraseña	asdf1234	Fallo	Fallo	Falta un carácter en mayúscula
Contraseña	Tamaño menor de 8 caracteres	Fallo	Fallo	Debe tener al menos 8 caracteres
Contraseña	Tamaño mayor de 128 caracteres	Fallo	Fallo	Debe tener menos de 128 caracteres
Contraseña	Vacía	Fallo	Fallo	No puede estar vacía

Tabla 12. Pruebas de caja negra del formulario de login

Añadir usuario

Campo	Entrada	Resultado esperado	Resultado obtenido	Detalles
DNI	12345678Z	Éxito	Éxito	DNI correcto
DNI	12345678A	Fallo	Fallo	Letra DNI incorrecta
DNI	1234567891A	Fallo	Fallo	Más de 9 caracteres
DNI	123456 78A	Fallo	Fallo	DNI con espacios
DNI	123456Z	Fallo	Fallo	Menos de 9 caracteres
Nombre	Oscar	Éxito	Éxito	Nombre correcto
Nombre	Oscar-1	Fallo	Fallo	Solo debe contener letras
Apellidos	Lista Rivera	Éxito	Éxito	Apellidos correctos

UVigoRes: Sistema de reserva de recursos

Apellidos	Lista Rivera 1	Fallo	Fallo	Solo debe contener letras
NIU	005000038457	Éxito	Éxito	NIU correcto
NIU	105000038457	Fallo	Fallo	El NIU debe ser de la forma 0050000XXXXX
NIU	00500003845734	Fallo	Fallo	El NIU debe ser de la forma 0050000XXXXX
NIU	005000038	Fallo	Fallo	El NIU debe ser de la forma 0050000XXXXX
Correo electrónico	olrivera@gmail.com	Éxito	Éxito	Correo electrónico correcto
Correo electrónico	olrivera@	Fallo	Fallo	Formato incorrecto
Correo electrónico	olrivera@gmail	Fallo	Fallo	Formato incorrecto
Contraseña	Asdf1234	Éxito	Éxito	Caracteres alfanuméricos y al menos una mayúscula
Contraseña	asdf1234	Fallo	Fallo	Falta un carácter en mayúscula
Contraseña	Tamaño menor de 8 caracteres	Fallo	Fallo	Debe tener al menos de 8 caracteres
Contraseña	Tamaño mayor de 128 caracteres	Fallo	Fallo	Debe tener menos de 128 caracteres
Contraseña	Vacía	Fallo	Fallo	No puede estar vacía
Todos	Vacía	Fallo	Fallo	Los campos no pueden estar vacíos

Tabla 13. Pruebas de caja negra del formulario de añadir usuario

Editar usuario

Campo	Entrada	Resultado esperado	Resultado obtenido	Detalles
DNI	12345678Z	Éxito	Éxito	DNI correcto
DNI	12345678A	Fallo	Fallo	Letra DNI incorrecta
DNI	1234567891A	Fallo	Fallo	Más de 9 caracteres
DNI	123456 78A	Fallo	Fallo	DNI con espacios
DNI	123456Z	Fallo	Fallo	Menos de 9 caracteres
Nombre	Oscar	Éxito	Éxito	Nombre correcto
Nombre	Oscar-1	Fallo	Fallo	Solo debe contener letras
Apellidos	Lista Rivera	Éxito	Éxito	Apellidos correctos
Apellidos	Lista Rivera 1	Fallo	Fallo	Solo debe contener letras
NIU	005000038457	Éxito	Éxito	NIU correcto
NIU	105000038457	Fallo	Fallo	El NIU debe ser de la forma 0050000XXXXX
NIU	00500003845734	Fallo	Fallo	El NIU debe ser de la forma 0050000XXXXX

UVigoRes: Sistema de reserva de recursos

NIU	005000038	Fallo	Fallo	El NIU debe ser de la forma 0050000XXXXX
Correo electrónico	olrivera@gmail.com	Éxito	Éxito	Correo electrónico correcto
Correo electrónico	olrivera@	Fallo	Fallo	Formato incorrecto
Correo electrónico	olrivera@gmail	Fallo	Fallo	Formato incorrecto
Todos	Vacía	Fallo	Fallo	Los campos no pueden estar vacíos

Tabla 14. Pruebas de caja negra en el formulario de editar usuario

Cambiar contraseña

Campo	Entrada	Resultado esperado	Resultado obtenido	Detalles
Contraseña	Asdf1234	Éxito	Éxito	Caracteres alfanuméricos y al menos una mayúscula
Contraseña	asdf1234	Fallo	Fallo	Falta un carácter en mayúscula
Contraseña	Tamaño menor de 8 caracteres	Fallo	Fallo	Debe tener al menos 8 caracteres
Contraseña	Tamaño mayor de 128 caracteres	Fallo	Fallo	Debe tener menos de 128 caracteres
Contraseña	Vacía	Fallo	Fallo	No puede estar vacía

Tabla 15. Pruebas de caja negra en el formulario de cambiar contraseña

Añadir y editar centro

Campo	Entrada	Resultado esperado	Resultado obtenido	Detalles
Nombre	Escola Superior de Exeñaría Informática	Éxito	Éxito	Nombre correcto
Nombre	123-123	Fallo	Fallo	No tiene ninguna letra
Dirección	Edificio Politécnico s/n, 32004 Ourense	Éxito	Éxito	Dirección correcta
Dirección	Edificio Politécnico_s/n, 32004 Ourense	Fallo	Fallo	Solo debe contener letras, números, puntos, comas y “.”
Teléfono	+34988387000	Éxito	Éxito	Teléfono correcto
Teléfono	988387000	Éxito	Éxito	Teléfono correcto
Teléfono	617595451	Éxito	Éxito	Teléfono correcto
Teléfono	722595451	Éxito	Éxito	Teléfono correcto
Teléfono	+35988387000	Fallo	Fallo	Prefijo incorrecto
Teléfono	9883870	Fallo	Fallo	No tiene al menos 9 dígitos

UVigoRes: Sistema de reserva de recursos

Teléfono	517595451	Fallo	Fallo	Debe empezar por 9, 6 o 7
Correo electrónico	info@esei.uvigo.es	Éxito	Éxito	Correo electrónico correcto
Correo electrónico	info@	Fallo	Fallo	Formato incorrecto
Correo electrónico	info@esei	Fallo	Fallo	Formato incorrecto
Todos	Vacía	Fallo	Fallo	Los campos no pueden estar vacíos

Tabla 16. Pruebas de caja negra del formulario de editar y añadir centro

Añadir y editar recurso

Campo	Entrada	Resultado esperado	Resultado obtenido	Detalles
Tipo	Portátil	Éxito	Éxito	Tipo correcto
Tipo	Portatil-1	Fallo	Fallo	Solo debe contener letras
Descripción	Lenovo Legion 7 - (Intel Core i7-10750H, 16GB RAM)	Éxito	Éxito	Descripción correcta
Todos	Vacía	Fallo	Fallo	Los campos no pueden estar vacíos

Tabla 17. Pruebas de caja negra del formulario añadir y editar recurso

Estas han sido las pruebas de caja negra realizadas para UVigoRes. En términos generales, se pueden describir de la siguiente manera:

- **Verificación de acceso restringido:** se ha comprobado que no es posible acceder a funcionalidades que no corresponden al usuario activo mediante la manipulación de solicitudes HTTP.
- **Validación de campos específicos:** se han realizado pruebas para asegurar que los campos ingresados sean correctos, como en el caso del NIU, donde debe seguir un formato concreto.
- **Verificación de mensajes de error y validación:** se ha asegurado que el sistema genere mensajes de error o de validación cuando se realizan acciones que impactan en la base de datos.
- **Validación de vistas según el rol del usuario:** se ha comprobado que todas las vistas de la aplicación se presenten adecuadamente según el rol del usuario que interactúa con el sistema.
- **Verificación de operaciones en la base de datos:** se ha comprobado que todas las operaciones relacionadas con la base de datos (como inserción, modificación, eliminación y consulta) se ejecutan correctamente para cada entidad gestionada por el sistema (como locales, clientes, etc.).

Estas pruebas permiten garantizar el correcto funcionamiento del sistema bajo distintas condiciones de uso.

10.2 Pruebas de caja blanca

Las pruebas de caja blanca garantizan que la lógica interna del sistema se comporta según lo especificado y que todos los componentes internos han sido probados adecuadamente.

A continuación, se presentan dos tablas que muestran las pruebas de caja blanca para los formularios de inicio de sesión y registro (este último es el mismo formulario utilizado para editar usuarios). Las acciones aplicadas en los formularios de inicio login y registro también se aplican a otros formularios del sistema, por lo que no se incluyen tablas adicionales para evitar repetición innecesaria.

Login

Descripción	Resultado esperado	Resultado obtenido	Detalles
El DNI no existe	Fallo	Fallo	Mensaje señalando que el DNI no existe
La contraseña para este usuario no es correcta	Fallo	Fallo	Mensaje señalando que no corresponde de la contraseña
DNI y contraseña corresponden al usuario	Éxito	Éxito	Acceso al sistema

Tabla 18. Pruebas de caja blanca del formulario de login

Añadir y editar usuario

Descripción	Resultado esperado	Resultado obtenido	Detalles
Ya existe un usuario con ese DNI o NIU	Fallo	Fallo	Mensaje señalando que ya existe un usuario con ese DNI o NIU
Registra un usuario con los datos de campos vacíos	Fallo	Fallo	Mensaje señalando que no puedes dejar campos vacíos
Los datos se añaden correctamente	Éxito	Éxito	El usuario se añade al sistema

Tabla 19. Pruebas de caja blanca de los formularios añadir y editar usuario

A continuación, se describen las pruebas realizadas en los formularios de inicio de sesión y registro. De manera general, las pruebas de caja blanca del sistema incluyen:

- **Prueba de rutas básicas:** esta prueba evalúa la complejidad del diseño del procedimiento y asegura que cada ruta posible en el código sea ejecutada al menos una vez mediante un conjunto de casos de prueba diseñados específicamente para este propósito.
- **Pruebas de bucles:** esta técnica permite verificar la correcta implementación de los bucles en el código de la aplicación. Los bucles han sido clasificados en cuatro

categorías: simples, concatenados, anidados y no estructurados, para asegurar que cada tipo funcione de manera adecuada.

PRUEBAS DE TRANSICIÓN, DEL DESPLIEGUE PARA EL CLIENTE

11 Manual de usuario

El manual de usuario, o guía de usuario, es un documento técnico diseñado para ayudar a las personas a utilizar un sistema específico.

En los siguientes apartados se guiará al usuario por la aplicación.

11.1 Requisitos mínimos

Para ejecutar este sistema, los dos requisitos mínimos principales son tener el sistema operativo Windows 10 y una versión de PHP que se encuentre entre PHP 8.1 y PHP 9.2 (la última versión de PHP 7). Las versiones anteriores de PHP ya no reciben soporte, por lo que los usuarios de esta versión deben actualizarla lo antes posible, ya que pueden estar expuestos a vulnerabilidades de seguridad sin parche.

11.2 Instalación

11.2.1 Descarga e instalación de XAMPP

Como mencionamos anteriormente, para poder ejecutar la aplicación debemos instalar previamente XAMPP. Para facilitarlo, se puede descargar desde este enlace:

<https://www.apachefriends.org/es/index.html>

Se muestran diferentes versiones de XAMPP, así como los sistemas operativos con los que es compatible (actualmente Windows, Linux y OS X). Se debe seleccionar el sistema operativo que se utiliza y la versión más reciente disponible de XAMPP. Además, es importante considerar la versión de PHP instalada (este trabajo se realiza con PHP 8.2).

Después de descargarlo, se ejecuta el archivo ejecutable y se sigue el proceso de instalación guiado. Al completar la instalación, la plataforma se ubicará en el disco C: en una carpeta llamada xampp. Dentro de esa carpeta, hay un archivo llamado xampp-control, un panel de control para conectarse a Apache y MySQL. También se puede abrir desde el explorador de aplicaciones. Una vez abierto se debe hacer clic en "Start" para Apache y MySQL. Si todo fue bien, el fondo de estos dos nombres debe estar en verde.

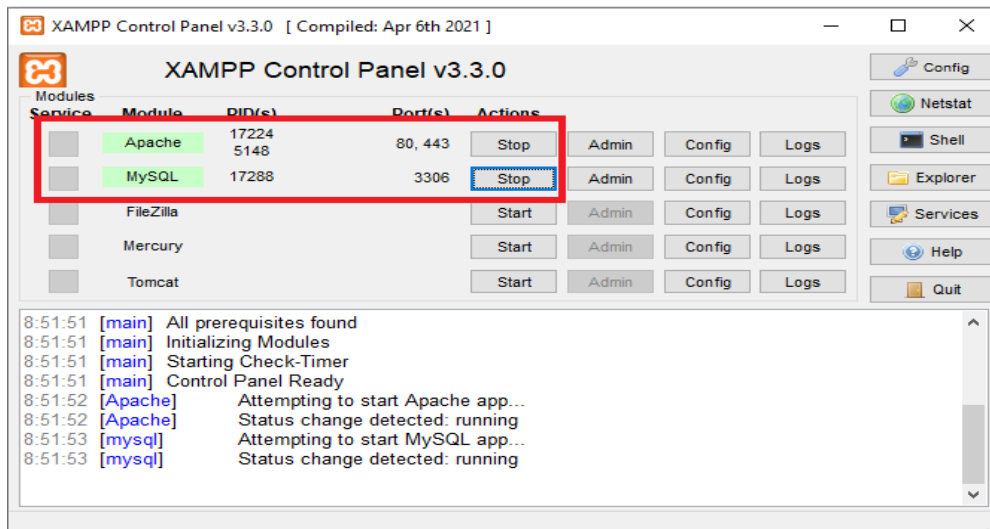


Ilustración 11. Panel de control de XAMPP

11.2.2 Ubicación del proyecto

Para poner en funcionamiento nuestro proyecto debemos colocar la carpeta que contiene el proyecto (llamada UVigoRes) en la ruta C:\xampp\htdocs. A continuación, se debe poner en el navegador (Firefox o Chrome) la siguiente dirección: localhost/UVigoRes/index.php

Si todo ha salido correctamente se va a visualizar la vista para poder ingresar en la aplicación:

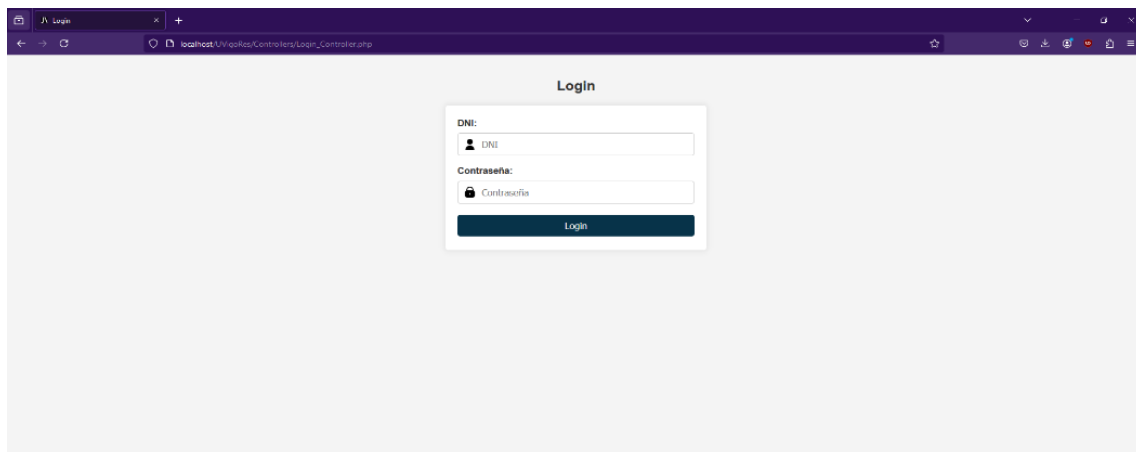


Ilustración 12. Vista de la página del login

11.2.3 Importación de los datos

Para poder utilizar la aplicación se necesitan datos. Para obtenerlos, se debe importar el script sql en el gestor de base de datos. Como XAMPP nos provee de uno lo haremos en phpMyadmin. Para ello tenemos que poner en el navegador (Firefox o Chrome) la siguiente dirección: localhost/phpmyadmin

Si todo ha salido correctamente se va a visualizar la página de phpMyAdmin:

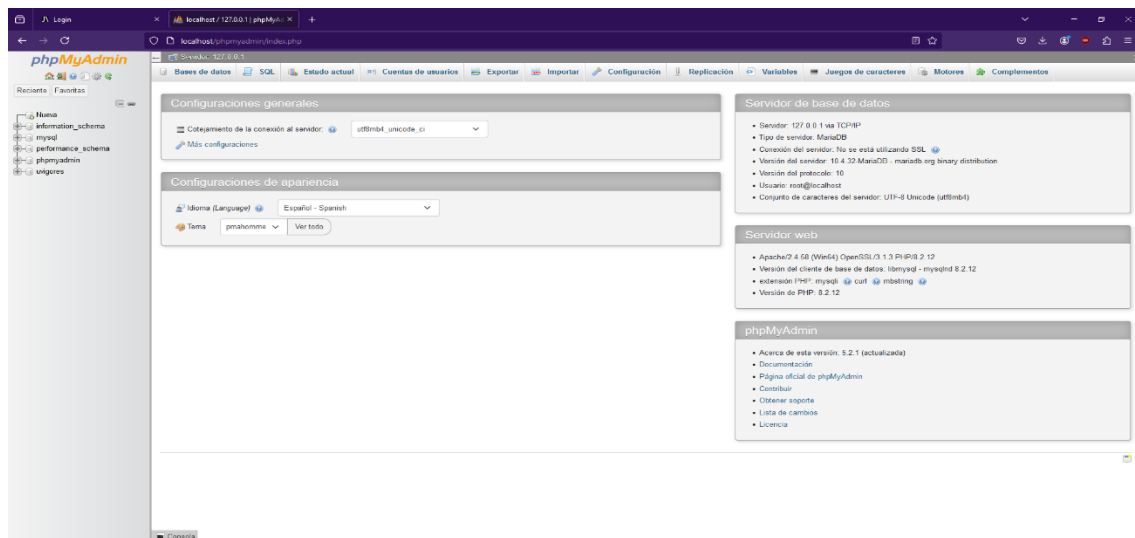


Ilustración 13. Vista de la página de phpMyAdmin

Para importar nuestra base de datos debemos ir a la opción Importar del menú superior. Tendremos que seleccionar nuestro script sql (ubicado en C:/xampp/htdocs/UVigoRes/UVigoRes.sql) para crear la base de datos. Hacemos clic en “Continuar” y si todo va bien el proceso de instalación habrá terminado.

Una vez hecho esto debemos consultar el archivo README.md para consultar los usuarios con sus datos para acceder a las distintas opciones de la aplicación.

11.3 Ingreso al sistema

La aplicación distingue 5 tipos de usuario que la usan: administrador, personal de conserjería, becario de infraestructura, docente y estudiante.

Para utilizar la aplicación es necesario estar registrado en el sistema.

Login

UVigoRes: Sistema de reserva de recursos

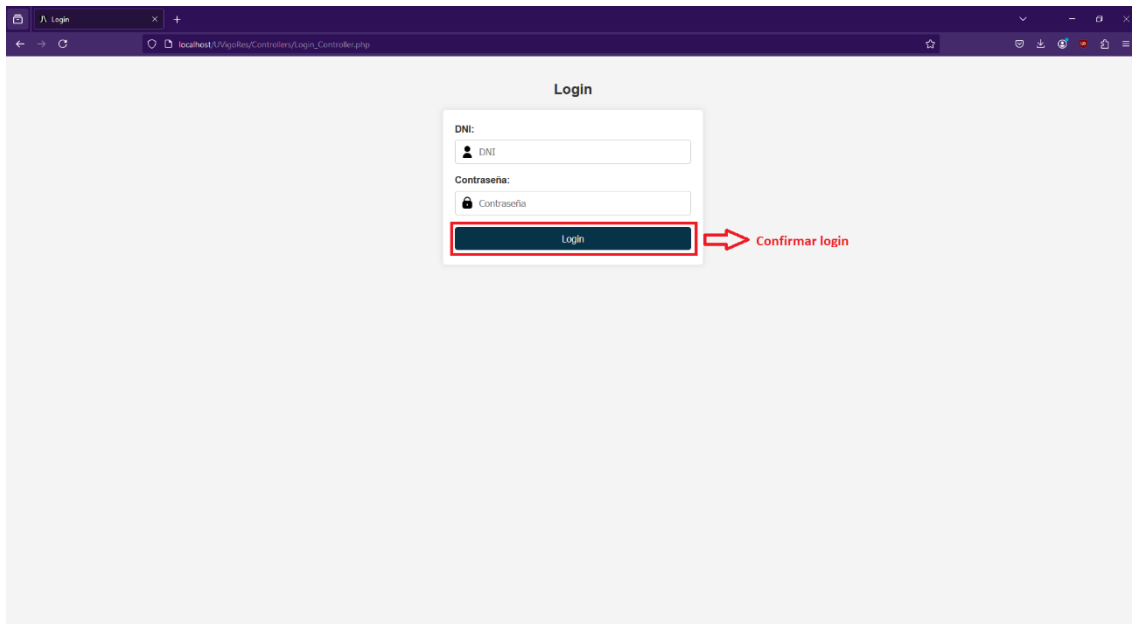


Ilustración 14. Vista del login de la aplicación

La primera vista de la aplicación es la del login, donde se debe introducir el DNI y la contraseña para acceder a la aplicación. Una vez introducidos los datos, si pulsamos en el botón “Login” y los datos son correctos, podemos acceder a la aplicación.

11.3.1 Vistas del administrador

Menú del administrador

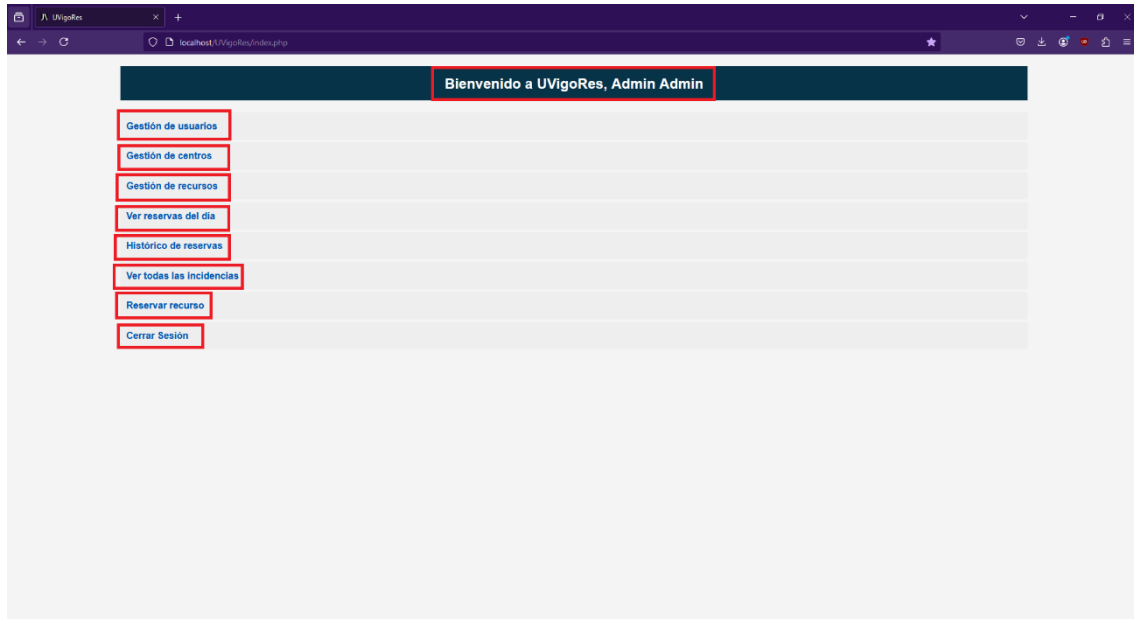


Ilustración 15. Vista del menú principal del administrador

La vista principal del administrador ofrece un menú donde puede gestionar los datos de la aplicación.

Empezando de arriba abajo, la aplicación nos da la bienvenida con un mensaje en donde se nos muestra el nombre y apellidos del usuario.

Más abajo tenemos los menús que haciendo clic en ellos nos dirigen a las distintas funcionalidades que explicaremos con detalle más adelante, que son:

- **Gestionar usuarios:** nos redirige a la vista de usuarios para la gestión de los usuarios
- **Gestión de centros:** nos redirige a la vista de usuarios para la gestión de los centros
- **Gestión de recursos:** nos redirige a la vista de usuarios para la gestión de los recursos
- **Ver reservas del día:** nos dirige a la vista donde se muestran las reservas de ese día
- **Histórico de reservas:** nos dirige a la vista donde se muestran un histórico de reservas
- **Ver todas las incidencias:** nos dirige a la vista donde se muestran las incidencias
- **Reservas recurso:** nos dirige a la vista donde se reservan los recursos
- **Cerrar Sesión:** cierra la sesión del usuario y nos redirige a la página del login

Gestionar usuarios

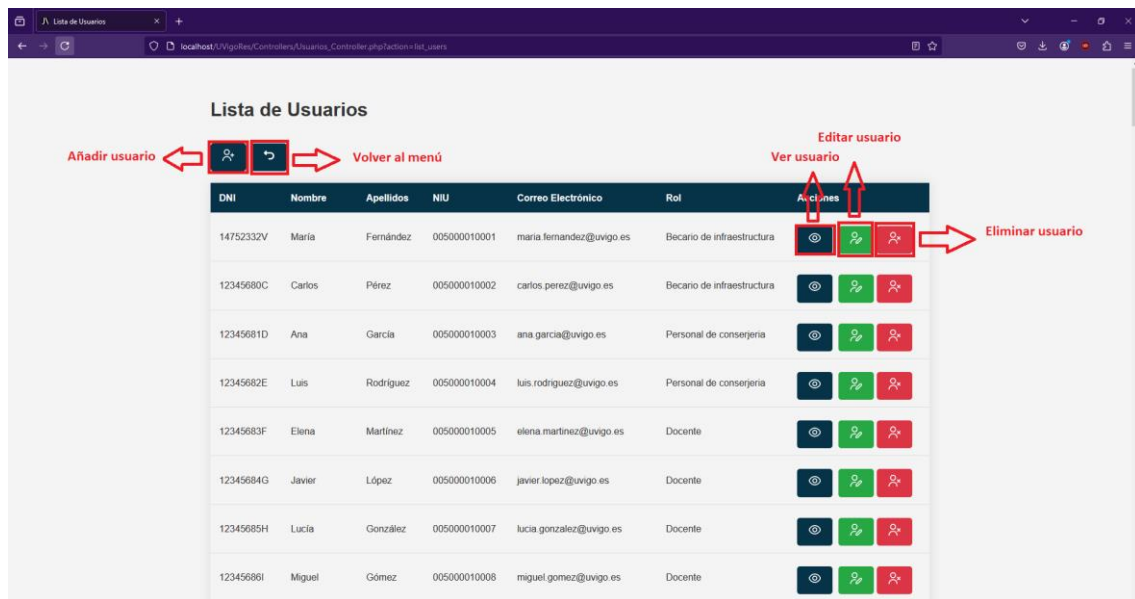


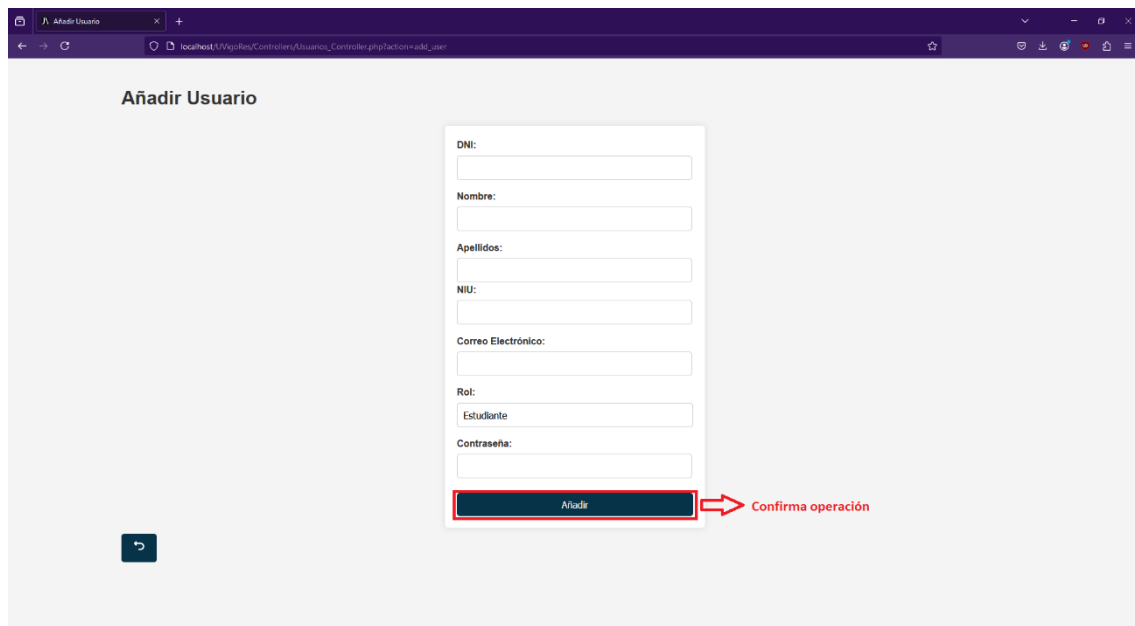
Ilustración 16. Vista de la gestión de usuarios

En esta vista se listan todos los usuarios del sistema junto con sus datos. Además, aparecen una serie de botones que si los clicamos nos llevarán a sus respectivas vista que se mostrará más adelante. Estos son:

- **Añadir usuario:** nos mostrará un formulario donde una vez cubiertos los datos del usuario se añade al sistema
- **Volver al menú:** este botón nos devuelve a la vista del menú principal
- **Ver usuario:** se nos muestra la información detallada del usuario
- **Editar usuario:** se nos mostrará un menú para elegir qué queremos editar del usuario
- **Eliminar usuario:** con este botón podremos eliminar un usuario, siempre con previa confirmación

Se ha optado por poner iconos y colores en los botones para que las acciones sean intuitivas, además de añadir un título si el usuario pone el cursor encima de los botones donde indica la opción del botón.

Añadir usuario



The screenshot shows a web browser window with the title 'Añadir Usuario'. The URL is 'localhost/UVigoRes/Controleros/Usuarios_Controller.php?action=add_user'. The form is titled 'Añadir Usuario' and contains the following fields: DNI, Nombre, Apellidos, NIU, Correo Electrónico, Rol (a dropdown menu with 'Estudiante' selected), and Contraseña. At the bottom of the form is a dark blue button labeled 'Añadir'. A red rectangular box highlights this button, and a red arrow points from the text 'Confirma operación' to the button.

Ilustración 17. Vista del formulario de añadir usuario

En esta vista el administrador puede añadir un usuario completando los datos del formulario. Todos los campos deben tener datos que serán validados. Además, el campo Rol, es un selector donde el administrador escoge el rol del usuario que va a crear.

Una vez los datos estén correctos y todos los campos rellenos, al clicar en “Añadir” se insertará el usuario en el sistema si todo ha salido correctamente.

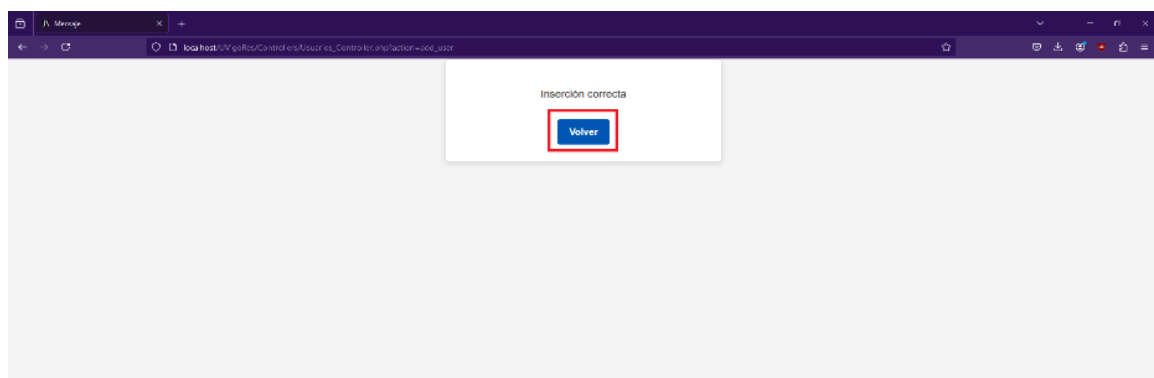


Ilustración 18. Mensaje de usuario agregado correctamente

Si pulsamos el botón de “Volver” seremos redirigidos a la página de gestión de usuarios de nuevo.

Ver usuario

Este botón muestra los datos de un usuario concreto.

UVigoRes: Sistema de reserva de recursos

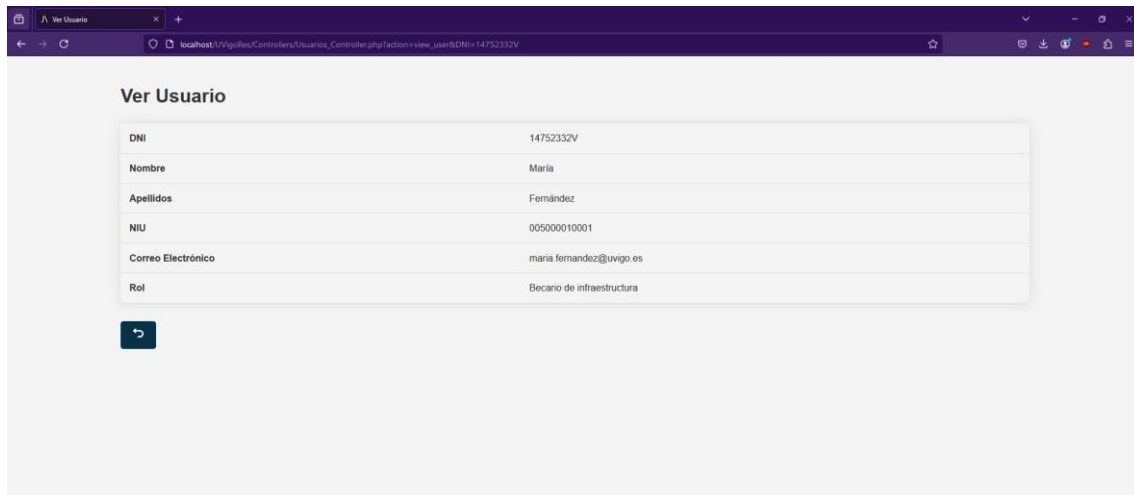


Ilustración 19. Vista de los datos de un usuario

Si pulsamos el botón de “Atrás” volveremos a la vista de gestión de usuarios.

Editar usuario

Esta vista tiene dos opciones: editar los datos de usuario, donde se podrán modificar los datos personales del mismo y cambiar la contraseña del usuario.

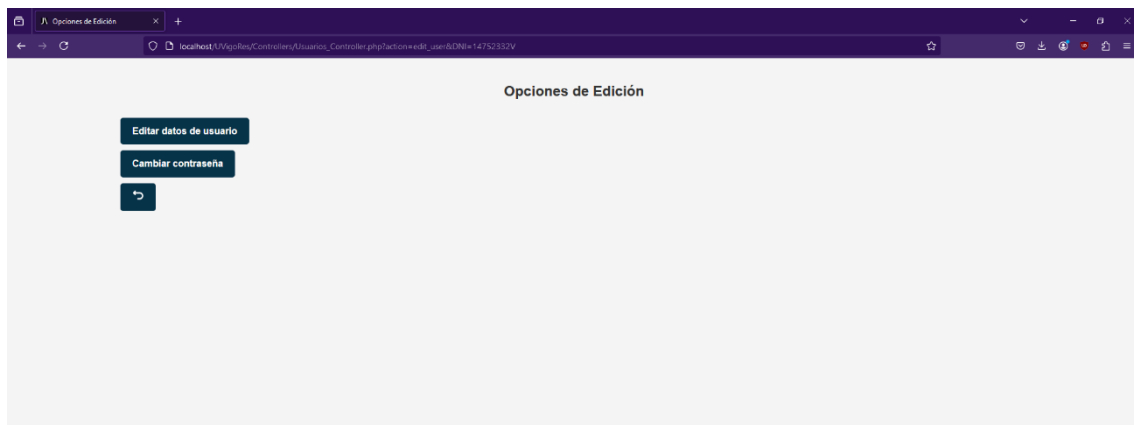
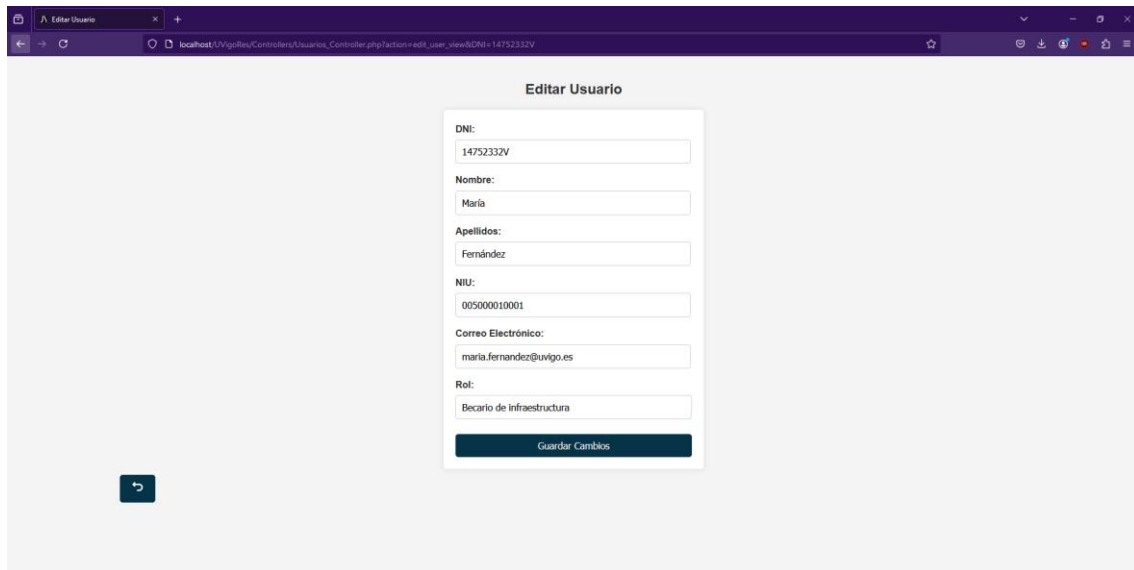


Ilustración 20. Menú de editar usuario

Si pulsamos en “Editar datos de usuario” se nos mostrará el siguiente formulario para editar los datos del usuario:

UVigoRes: Sistema de reserva de recursos



The screenshot shows a web browser window with the title 'Editar Usuario'. The address bar shows the URL 'localhost/UVigoRes/Controllers/Usuarios_Controller.php?action=edit_user_cnew&DNI=14752332V'. The form itself is titled 'Editar Usuario' and contains the following fields:

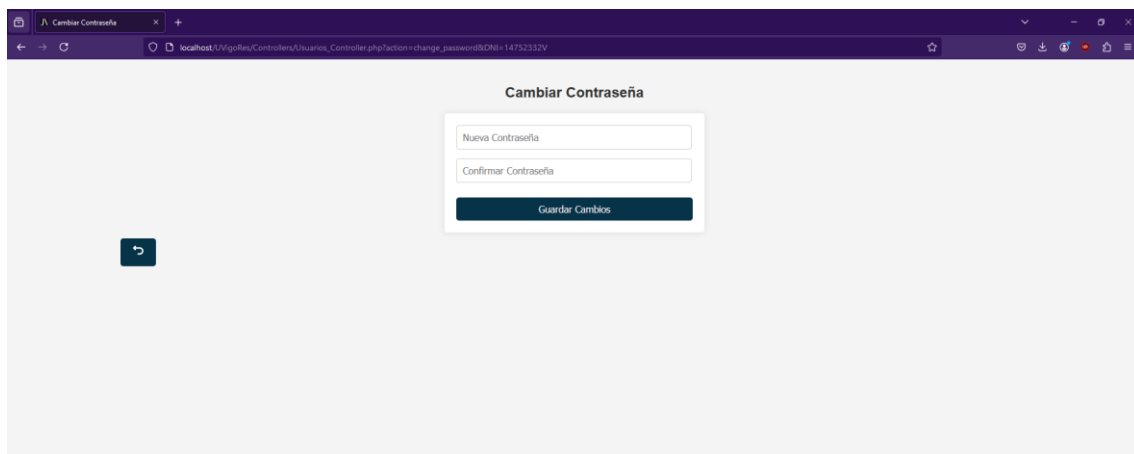
- DNI: 14752332V
- Nombre: María
- Apellidos: Fernández
- NIU: 005000010001
- Correo Electrónico: maria.fernandez@uvigo.es
- Rol: Becario de Infraestructura

At the bottom of the form is a dark blue button labeled 'Guardar Cambios'. There is also a small blue button with a circular arrow icon on the left side of the form.

Ilustración 21. Vista del formulario para editar los datos del usuario

En este formulario se pueden modificar los datos del usuario. Todos los datos se verifican que estén correctos. Una vez verificados, si pulsamos en “Guardar Cambios” los datos serán modificados y se nos mostrará un mensaje de confirmación.

Si optamos por solo modificar la contraseña del usuario, lo que veremos será el siguiente formulario:



The screenshot shows a web browser window with the title 'Cambiar Contraseña'. The address bar shows the URL 'localhost/UVigoRes/Controllers/Usuarios_Controller.php?action=change_password&DNI=14752332V'. The form itself is titled 'Cambiar Contraseña' and contains the following fields:

- Nueva Contraseña
- Confirmar Contraseña

At the bottom of the form is a dark blue button labeled 'Guardar Cambios'. There is also a small blue button with a circular arrow icon on the left side of the form.

Ilustración 22. Vista del formulario para cambiar la contraseña

En esta vista se nos pedirá que escribamos la contraseña dos veces y se verificará que en ambos campos sean la misma. De no ser así se nos mostrará el siguiente mensaje:

UVigoRes: Sistema de reserva de recursos

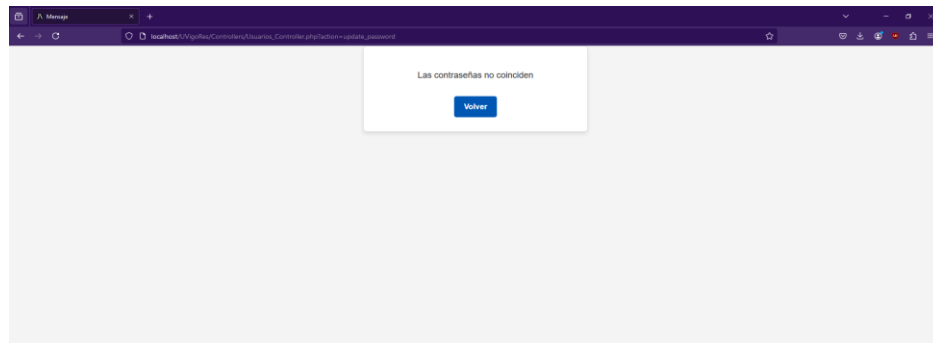


Ilustración 23. Mensaje de error si las contraseñas no coinciden

Si pulsamos el botón de “Volver” regresaremos al menú de gestión de usuarios.

Gestión de centros

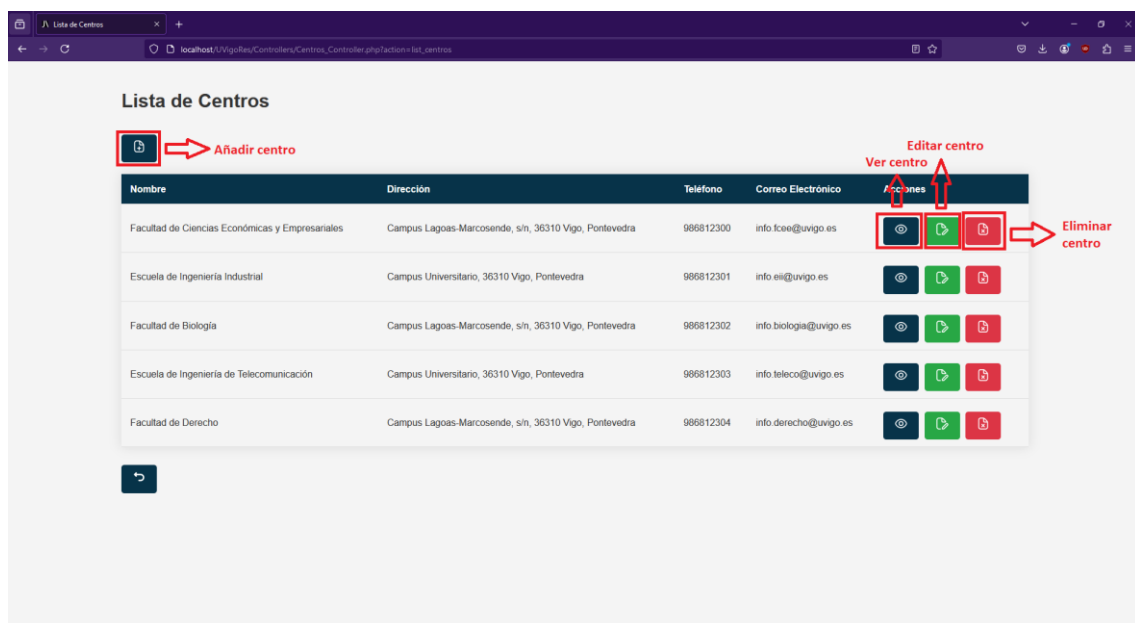


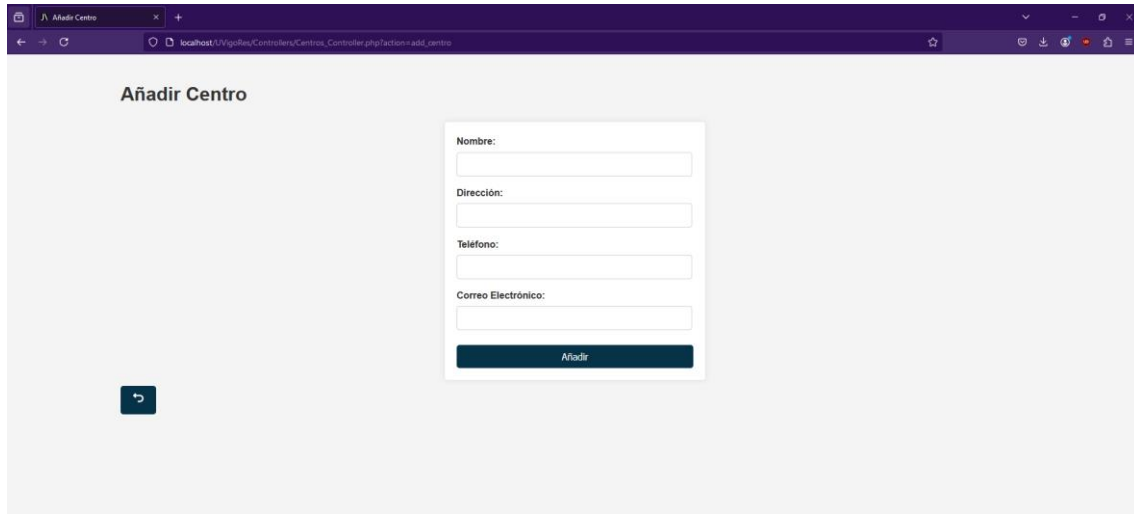
Ilustración 24. Vista de la gestión de centros

En esta vista el administrador consulta todos los datos de los centros que existen en el sistema, además de las siguientes opciones:

- **Añadir centro:** nos mostrará un formulario donde una vez cubiertos los datos del centro se añade al sistema
- **Volver al menú:** este botón nos devuelve a la vista del menú principal
- **Ver centro:** se nos muestra la información detallada del centro
- **Editar centro:** se nos mostrará un formulario similar al de añadir un centro
- **Eliminar centro:** con este botón podremos eliminar un centro, siempre con previa confirmación

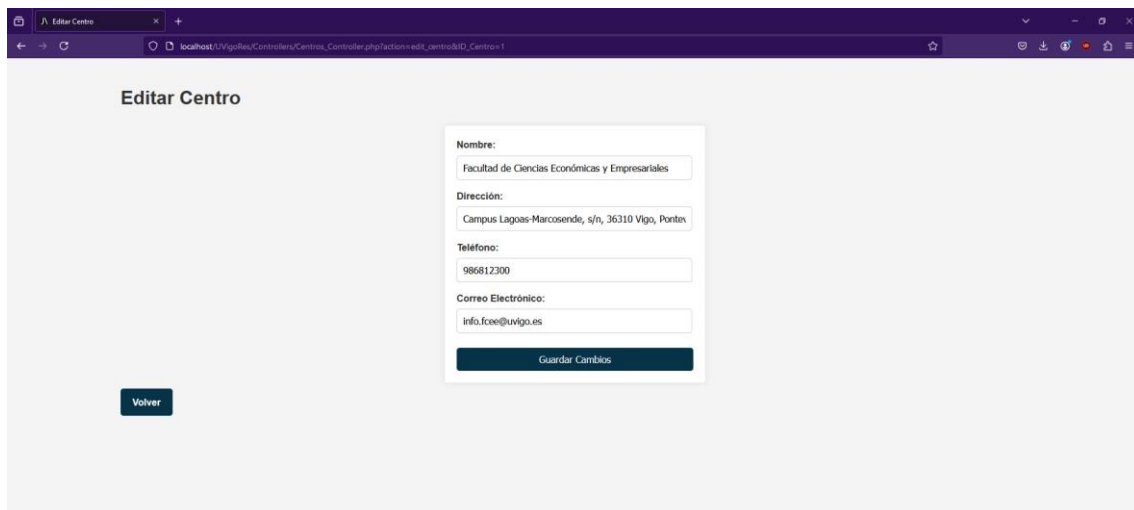
También se ha optado por poner iconos y colores en los botones para que las acciones sean intuitivas, además de añadir un título si el usuario pone el cursor encima de los botones donde indica la opción del botón.

Añadir y editar centro



The screenshot shows a web browser window with the title 'Añadir Centro'. The address bar shows the URL 'localhost:114/galera/Controlers/Centros_Controller.php?action=add_centro'. The main content area has a heading 'Añadir Centro' and a form with the following fields: 'Nombre:', 'Dirección:', 'Teléfono:', and 'Correo Electrónico:'. Each field has a corresponding text input box. Below the inputs is a dark blue button labeled 'Añadir'. In the bottom left corner, there is a dark blue button with a white left-pointing arrow.

Ilustración 25. Vista del formulario de añadir centro



The screenshot shows a web browser window with the title 'Editar Centro'. The address bar shows the URL 'localhost:114/galera/Controlers/Centros_Controller.php?action=edit_centro&ID_Centros=1'. The main content area has a heading 'Editar Centro' and a form with the following fields: 'Nombre:', 'Dirección:', 'Teléfono:', and 'Correo Electrónico:'. Each field has a corresponding text input box containing pre-filled data: 'Facultad de Ciencias Económicas y Empresariales', 'Campus Lagoas-Marcosende, s/n, 36310 Vigo, Ponte', '986812300', and 'info.fcce@uvigo.es'. Below the inputs is a dark blue button labeled 'Guardar Cambios'. In the bottom left corner, there is a dark blue button labeled 'Volver'.

Ilustración 26. Vista del formulario de editar centro

Los formularios de añadir y editar centro son similares, teniendo la diferencia que el formulario de editar recupera y muestra los datos del centro que se sea editar. Ambos formularios realizan validaciones de los datos para que sean coherentes y ningún dato esté vacío.

En el formulario de añadir centro los datos se guardan en el sistema si pulsamos en “Añadir”; mientras que en el formulario de editar centro tendremos que pulsar en el botón de “Guardar cambios”.

En ambos formularios si damos al botón de “Atrás” regresaremos al menú de gestión de centros.

Gestión de recurso

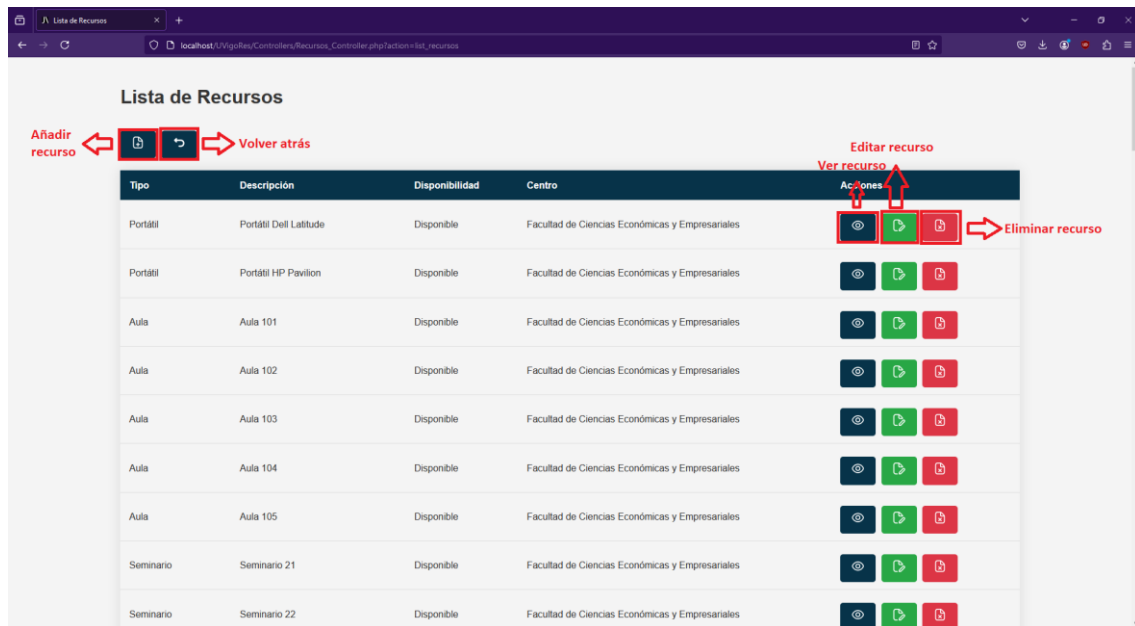


Ilustración 27. Vista de gestión de recursos

En esta vista el administrador consulta todos los datos de los recursos que existen en el sistema, además de las siguientes opciones:

- **Añadir recurso:** nos mostrará un formulario donde una vez cubiertos los datos del recurso se añade al sistema
- **Volver al menú:** este botón nos devuelve a la vista del menú principal
- **Ver recurso:** se nos muestra la información detallada del recurso
- **Editar recurso:** se nos mostrará un formulario similar al de añadir un recurso
- **Eliminar centro:** con este botón podremos eliminar un recurso, siempre con previa confirmación

Se ha optado por poner iconos y colores en los botones para que las acciones sean intuitivas, además de añadir un título si el usuario pone el cursor encima de los botones donde indica la opción del botón.

Añadir recurso

UVigoRes: Sistema de reserva de recursos

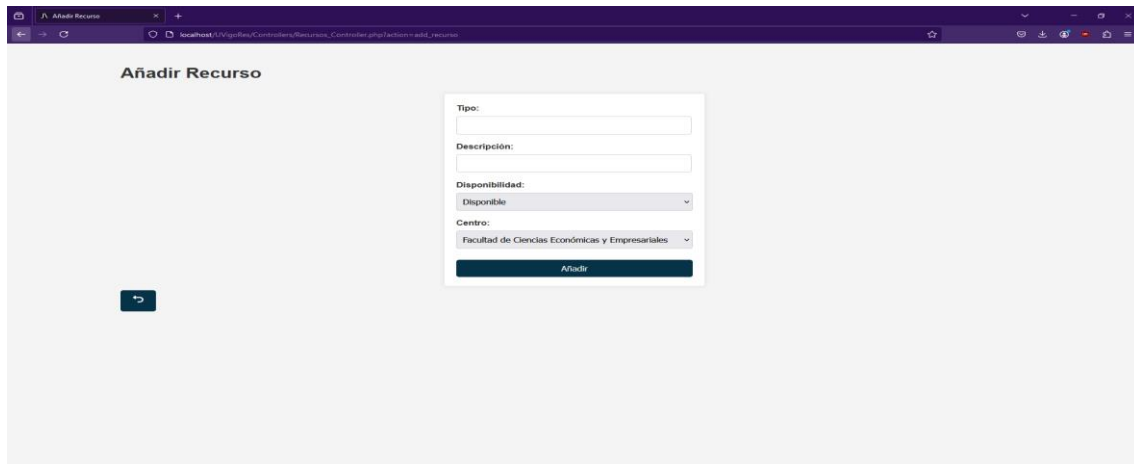


Ilustración 28. Vista del formulario de añadir recurso

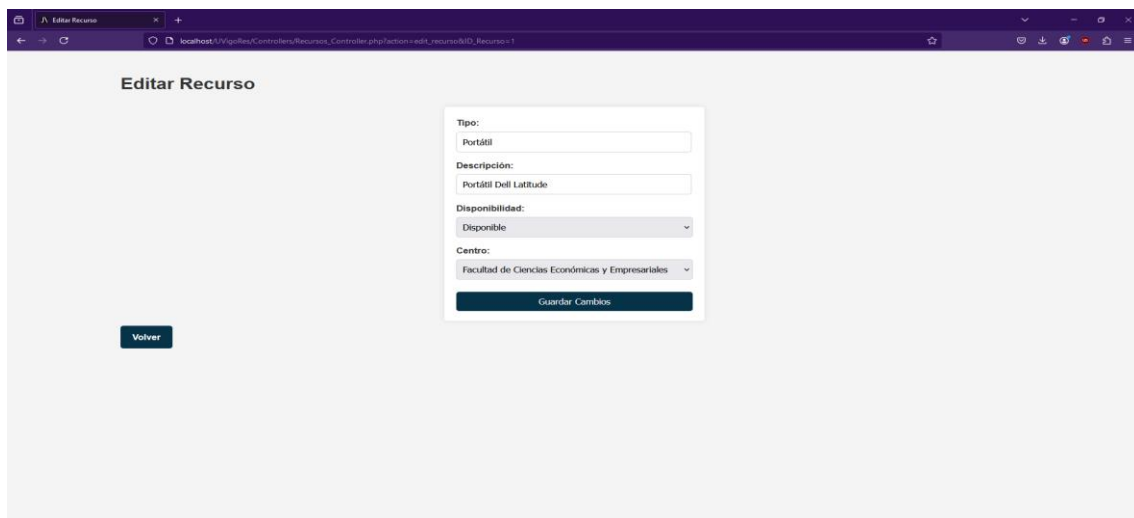


Ilustración 29. Vista del formulario de editar recurso

Los formularios de añadir y editar recurso son similares, teniendo la diferencia que el formulario de editar recupera y muestra los datos del centro que se sea editar. Ambos formularios realizan validaciones para que ningún dato esté vacío.

En el formulario de añadir centro los datos se guardan en el sistema si pulsamos en “Añadir”; mientras que en el formulario de editar recurso tendremos que pulsar en el botón de “Guardar cambios”.

Ver reservas del día

En esta vista se muestran todas las reservas del día, así como todos los datos de ésta. Además, el administrador puede confirmar la reserva o eliminarla. Si la reserva está confirmada, no aparece el botón de confirmar.

Si eliminamos la reserva, previa confirmación, nos saldrá un mensaje confirmando el borrado.

UVigoRes: Sistema de reserva de recursos

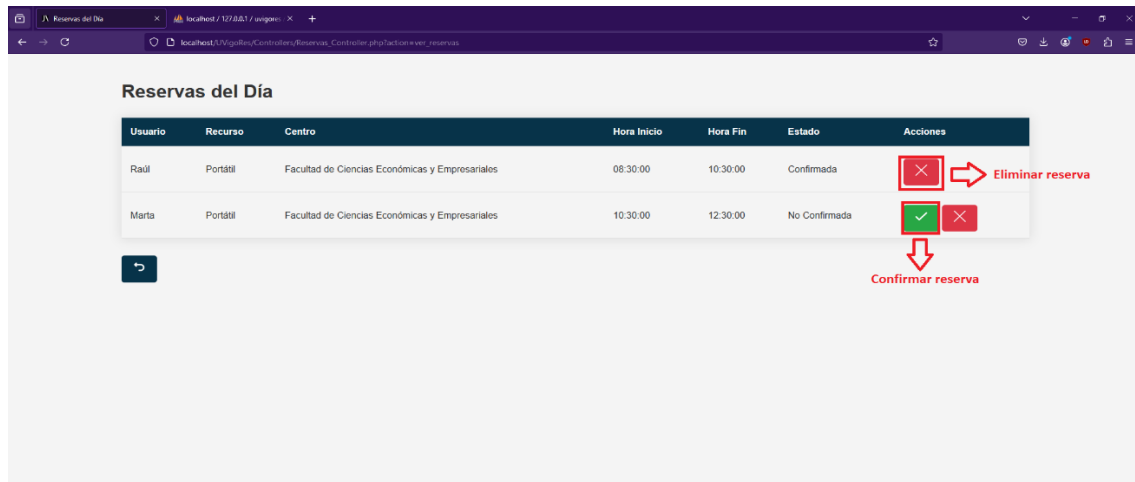


Ilustración 30. Vista de reservas del día

Se han añadido iconos y colores en los botones para que las acciones sean intuitivas, además de añadir un título si el usuario pone el cursor encima de los botones donde indica la opción del botón.

Histórico de reservas

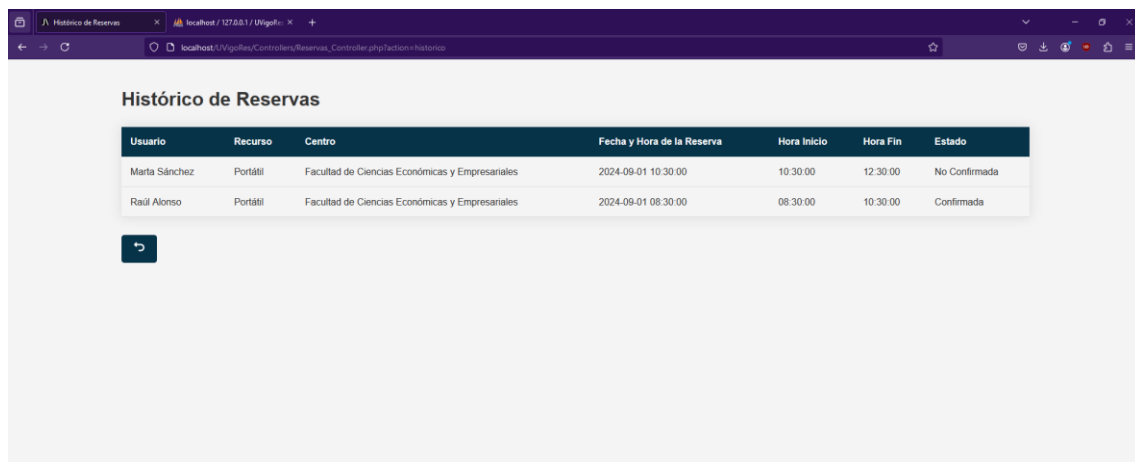


Ilustración 31. Vista del histórico de reservas

En esta vista se muestran todas las reservas realizadas, así como sus datos.

Ver todas las incidencias

UVigoRes: Sistema de reserva de recursos

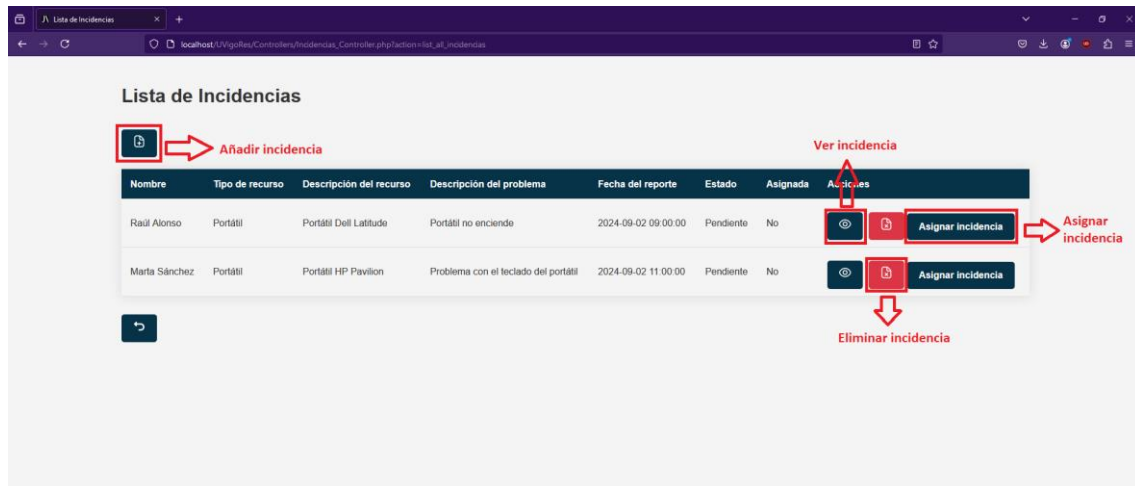


Ilustración 32. Vista de todas las incidencias

En esta vista se muestran las incidencias del sistema, además de las siguientes opciones:

- **Añadir incidencia:** nos mostrará un formulario donde una vez cubiertos los datos de la incidencia se añade al sistema
- **Volver al menú:** este botón nos devuelve a la vista del menú principal
- **Ver incidencia:** se nos muestra la información detallada de la incidencia
- **Asignar incidencia:** se nos mostrará un formulario para asignar la incidencia
- **Eliminar centro:** con este botón podremos eliminar una incidencia, siempre con previa confirmación

Se ha optado por poner iconos y colores en los botones para que las acciones sean intuitivas, además de añadir un título si el usuario pone el cursor encima de los botones donde indica la opción del botón.

Añadir incidencia

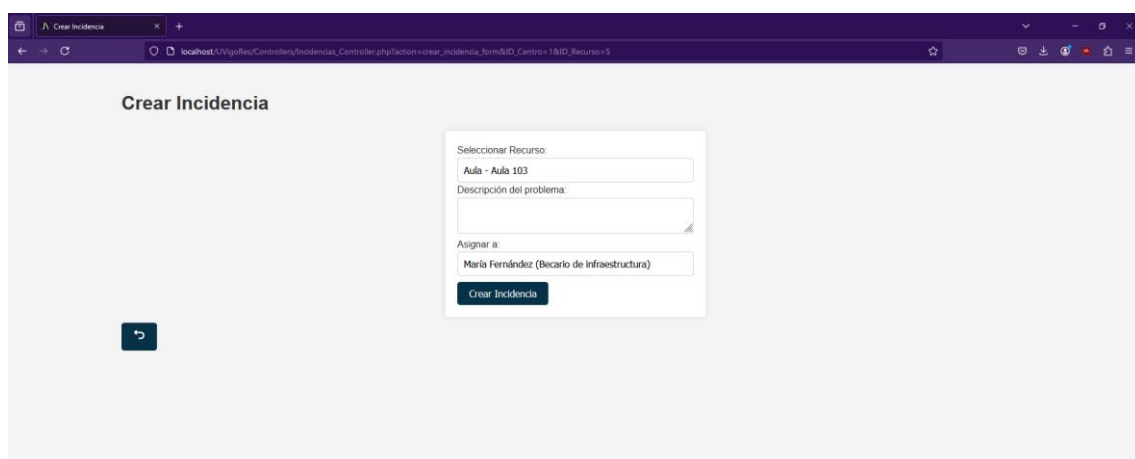


Ilustración 33. Vista del formulario de añadir incidencia

UVigoRes: Sistema de reserva de recursos

En esta vista se selecciona un recurso, se describe el problema de dicho recurso y se selecciona el personal de conserjería o becario de infraestructura al que se le asignará esa incidencia.

Una vez añadida, la incidencia se crea en el sistema.

Ver incidencia

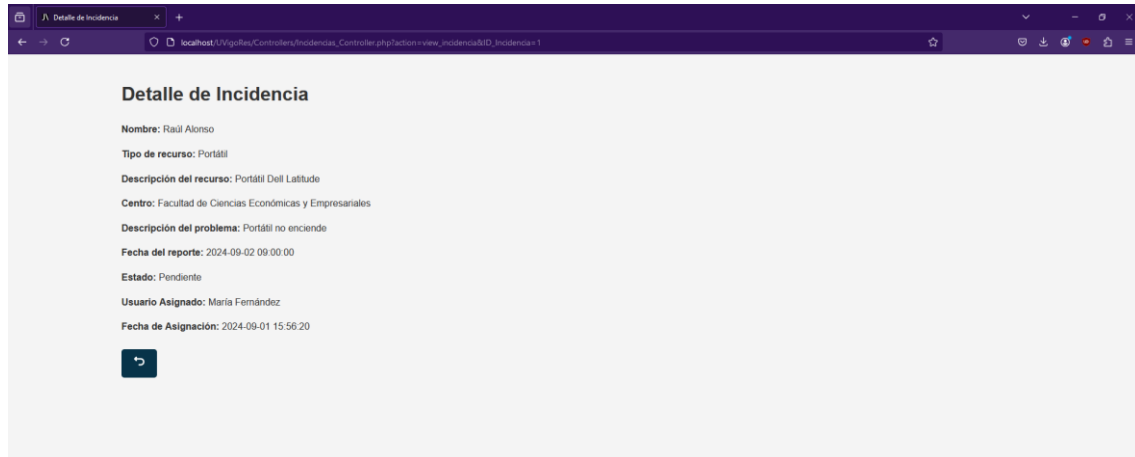


Ilustración 34. Vista donde se ven los datos de la incidencia

Aquí se ven los todos los datos de la incidencia.

Asignar incidencia

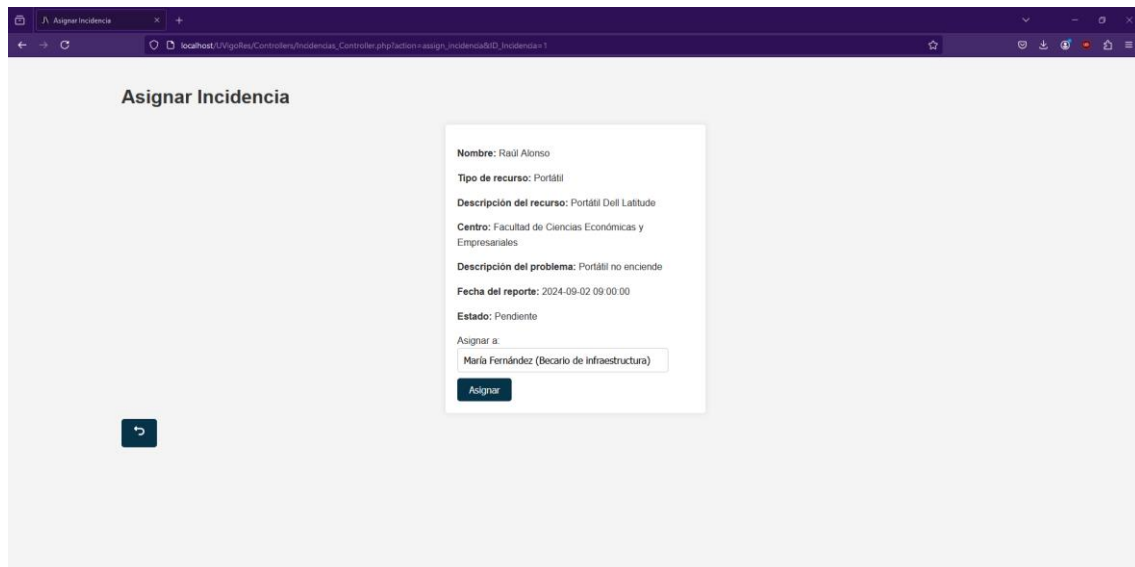


Ilustración 35. Vista de asignar incidencia

En esta vista se asigna una incidencia a un becario de infraestructura o personal de conserjería. Para asignarla solo hay que pulsar el botón de “Asignar”.

Reservar recurso

Esto se explica con los usuarios de rol estudiante y docente.

Cerrar sesión

Pulsando en esta opción se cerrará la sesión y se volverá al menú de login.

11.3.2 Vistas del Personal de conserjería y Becario de infraestructura

Menú de personal de conserjería y becario de infraestructura

La vista del usuario con el rol de personal de conserjería y becario de infraestructura es que los últimos no pueden consultar las reservas del día y el histórico de las mismas.

Empezando de arriba abajo, la aplicación nos da la bienvenida con un mensaje en donde se nos muestra el nombre y apellidos del usuario.



Ilustración 36. Vista del menú de personal de conserjería



Ilustración 37. Vista del menú de becario de infraestructura

En ambos usuarios, si pulsamos en el menú de “Ver todas las incidencias” se nos mostrará la misma vista y opciones que en el apartado **Vistas del administrador-Ver todas las incidencias**, explicada anteriormente.

Ver mis incidencias

UVigoRes: Sistema de reserva de recursos

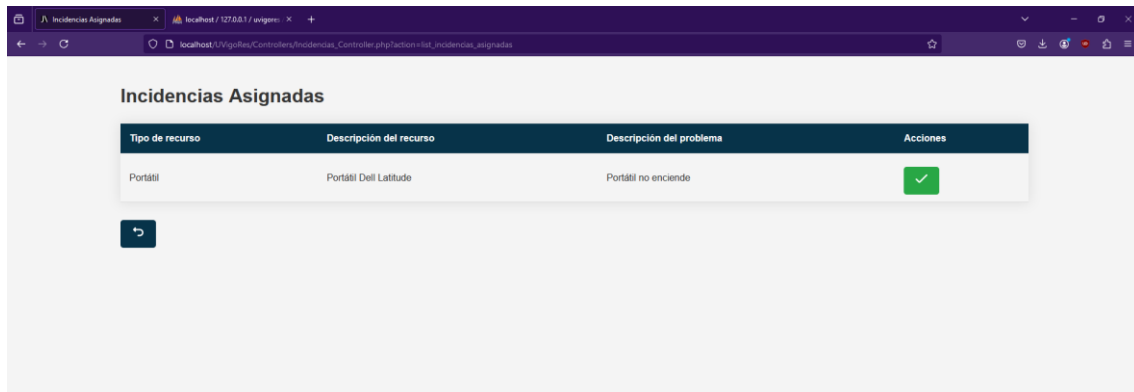


Ilustración 38. Vista de mis incidencias asignadas

En esta vista, los usuarios ven las incidencias y sus datos que tienen asignadas para su resolución. Tienen un botón para que, una vez resuelta, la incidencia pasa a estar resuelta.

Reservar recurso

Esto se explica con los usuarios de rol estudiante y docente.

Cerrar sesión

Pulsando en esta opción se cerrará la sesión y se volverá al menú de login.

11.3.3 Vistas de docente y estudiante

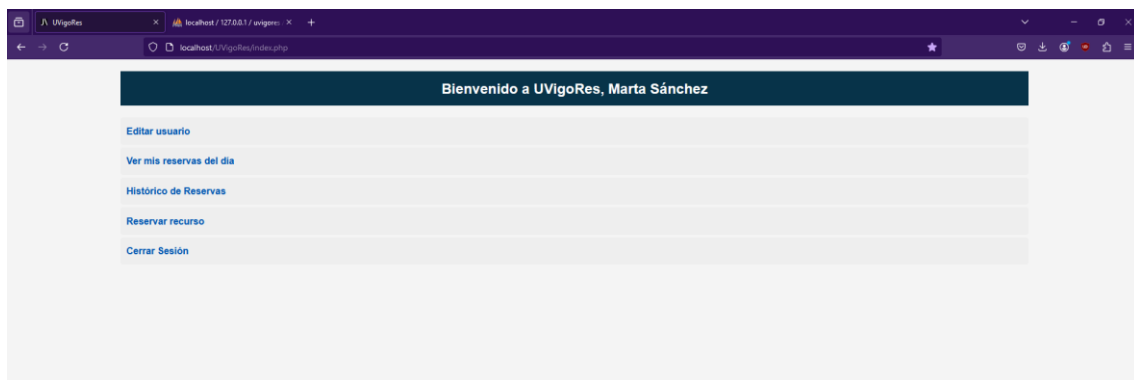


Ilustración 39. Vista de docente y estudiante

La vista principal del usuario docente y estudiante ofrece un menú donde se puede utilizar la aplicación.

Empezando de arriba abajo, la aplicación nos da la bienvenida con un mensaje en donde se nos muestra el nombre y apellidos del usuario.

Más abajo tenemos los menús que haciendo clic en ellos nos dirigen a las distintas funcionalidades que explicaremos con detalle más adelante, que son:

- **Editar usuario:** se nos mostrará un menú para elegir qué queremos editar del usuario
- **Ver reservas mis reservas día:** nos dirige a la vista donde se muestran mis reservas de ese día

UVigoRes: Sistema de reserva de recursos

- **Histórico de reservas:** nos dirige a la vista donde se muestran un histórico de reservas
- **Reservas recurso:** nos dirige a la vista donde se reservan los recursos
- **Cerrar Sesión:** cierra la sesión del usuario y nos redirige a la página del login

Editar usuario

Al igual que en la vista del administrador, podremos elegir si queremos editar nuestra información o cambiar la contraseña.

Si elegimos la primera opción, el único dato que podremos modificar en el formulario es el correo electrónico, puesto que no tiene sentido poder modificar el resto de los datos.

Si queremos modificar ese dato o la contraseña el formulario valida los datos de la misma forma que se explicó con anterioridad con el administrador.

Ver mis reservas del día

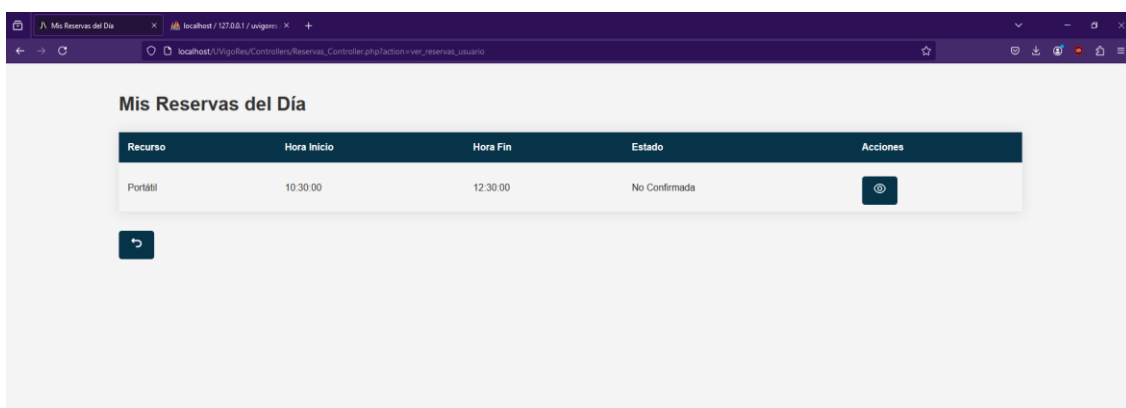


Ilustración 40. Vista donde se muestran las reservas del día

En esta vista se muestran todas las reservas de ese usuario para el día que estemos, así como todos los datos. También se puede consultar los detalles del mismo modo que lo vería el administrador (acción explicada con anterioridad).

Histórico de reservas

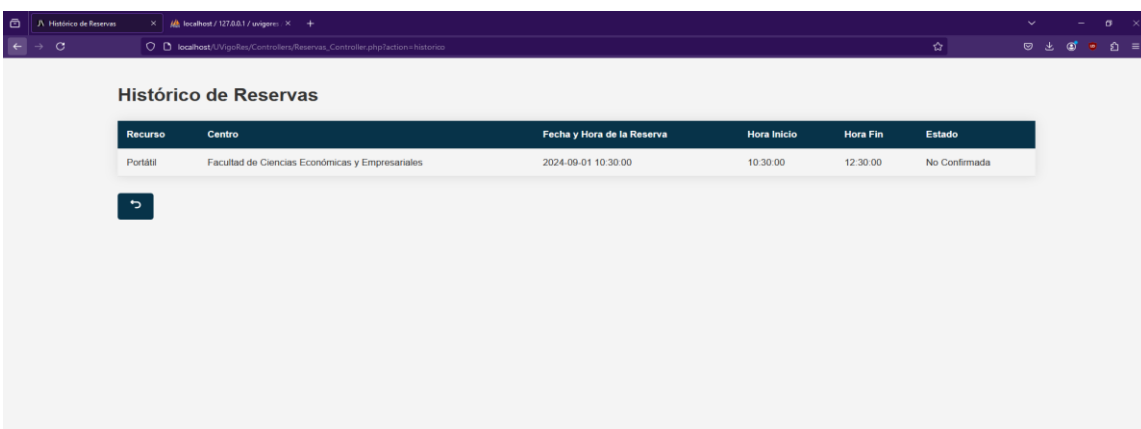


Ilustración 41. Vista del histórico de reservas

UVigoRes: Sistema de reserva de recursos

En esta vista se muestran todas las reservas, así como sus datos, que el usuario ha hecho.

Reservar recurso

En esta vista el usuario es guiado por una serie de vistas donde podrá seleccionar el recurso que desea reservar.

La primera de ella es seleccionar el centro donde desea reservar el recurso.

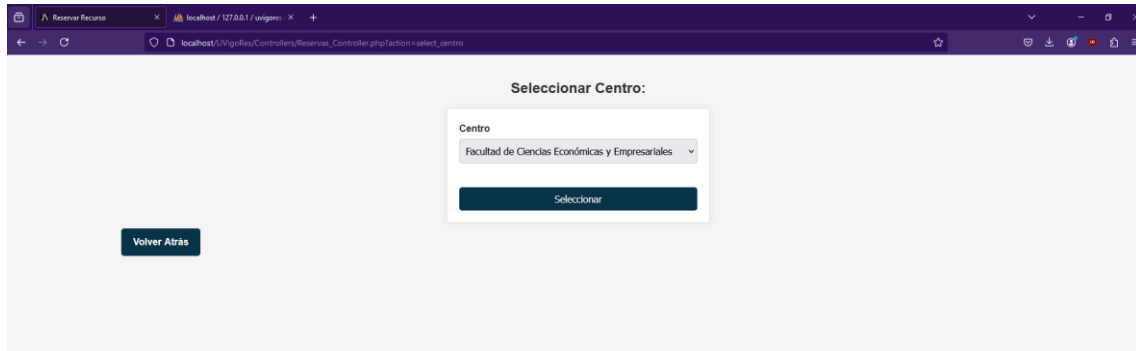


Ilustración 42. Vista donde se selecciona el centro

La siguiente vista debe seleccionar el tipo de recurso: los recursos de tipo aula, sala de juntas, salón de actos o laboratorios no se muestran a los estudiantes.

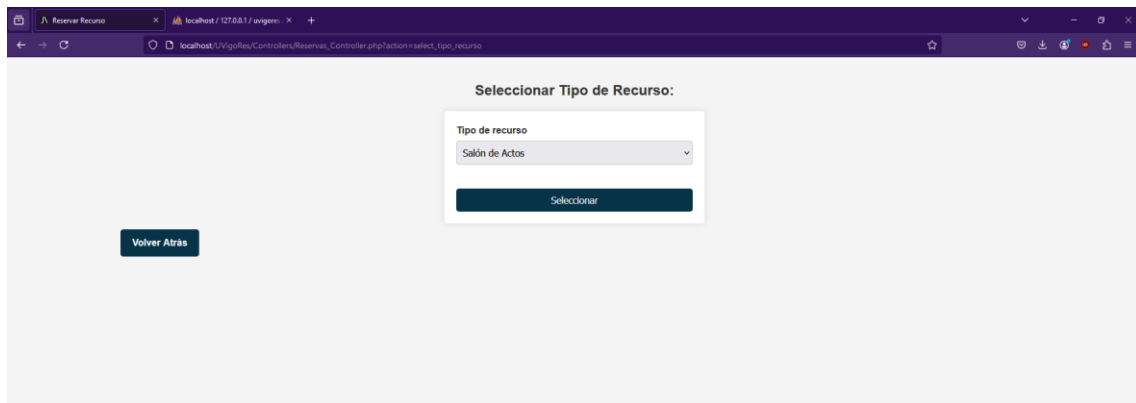


Ilustración 43. Vista donde se selecciona el tipo de recurso

Una vez seleccionado el tipo de recurso debemos seleccionar el recurso.

UVigoRes: Sistema de reserva de recursos

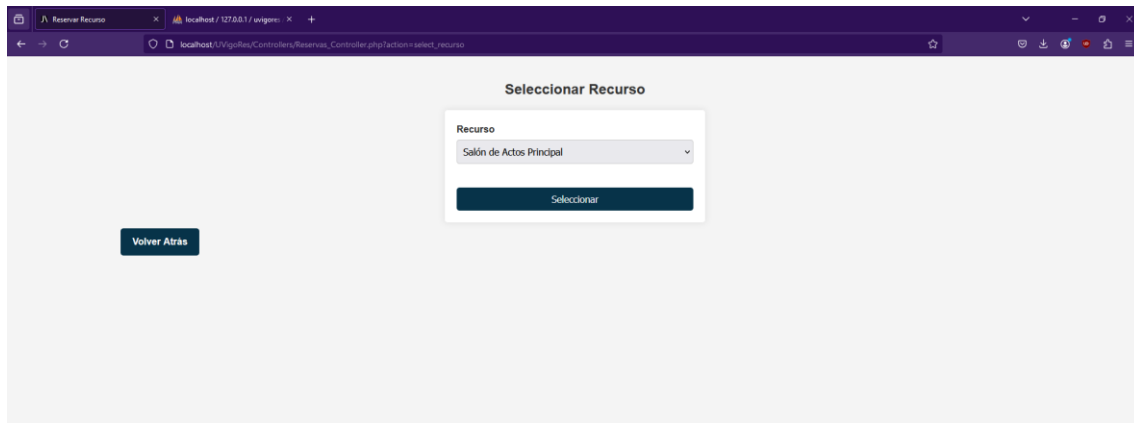


Ilustración 44. Vista donde se selecciona el recurso

Cuando ya elegimos el recurso seremos dirigidos a la vista donde se selecciona la franja horaria deseada.

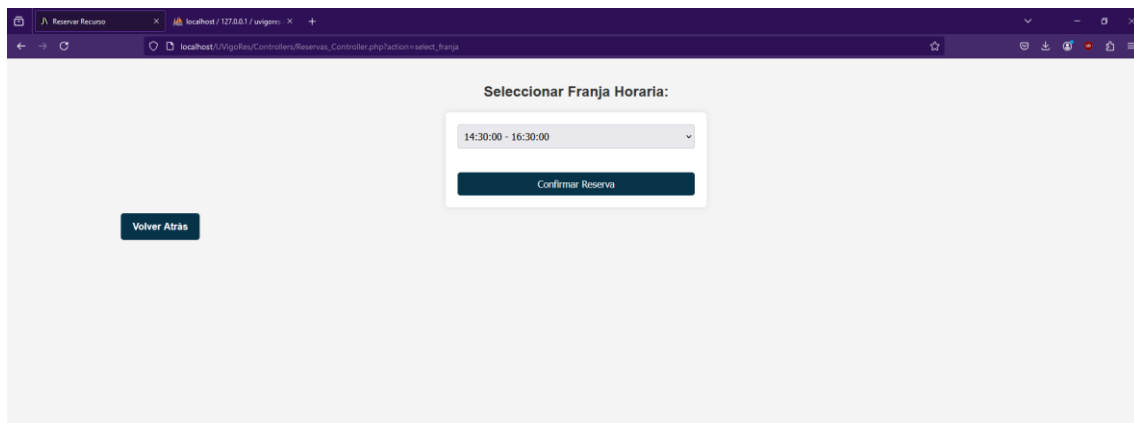


Ilustración 45. Vista donde se selecciona la franja horaria

Si la reserva se ha realizado se nos mostrará el siguiente mensaje:

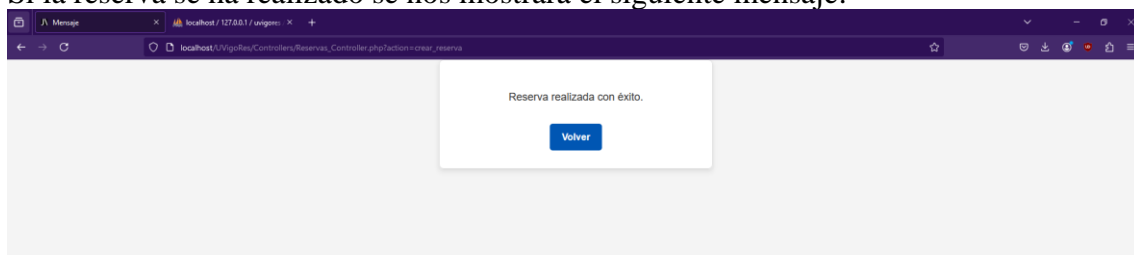


Ilustración 46. vista de mensaje de confirmación de reserva

Una vez le demos a “Volver” seremos redirigidos al menú principal.

Cerrar sesión

Pulsando en esta opción se cerrará la sesión y se volverá al menú de login.

12 Principales aportaciones

La aplicación desarrollada para la gestión de recursos en un entorno universitario ofrece varias aportaciones significativas:

- Aplicación personalizada para la gestión de recursos: facilita la administración eficiente de aulas, laboratorios y equipos tecnológicos, ahorrando tiempo y reduciendo la carga administrativa.
- Adaptabilidad a diferentes centros: diseñada para ser flexible y ajustarse a las necesidades específicas de diversas facultades, maximizando su potencial de uso.
- Facilitación de reservas y uso de recursos: mejora la accesibilidad y la organización en la consulta y reserva de recursos, promoviendo un uso más equitativo y eficiente.
- Gestión de incidencias: incluye un sistema para reportar y gestionar problemas, mejorando la experiencia del usuario y el seguimiento del uso de los recursos.
- Uso de tecnologías gratuitas: la aplicación es económica y accesible, además de estar adaptada para dispositivos móviles, lo que promueve su uso inclusivo.
- Seguridad y control de acceso: implementa medidas para asegurar el uso adecuado y seguro de los recursos, garantizando su correcta utilización.
- Digitalización de procesos universitarios: contribuye a la modernización y digitalización de la administración universitaria, mejorando las operaciones y la experiencia educativa.

13 Conclusiones

13.1 Conclusiones técnicas

Se ha utilizado PHP como lenguaje web para desarrollar la aplicación UVigoRes de manera rápida y eficiente. PHP fue elegido principalmente por su capacidad de integrarse bien con el patrón de arquitectura MVC, lo que facilita su uso junto a HTML y CSS para la creación de vistas, así como con SQL para gestionar la base de datos.

Además, PHP cuenta con una extensa comunidad en línea que proporciona un gran soporte y recursos, como su completo manual online.

Los otros lenguajes, como HTML y CSS, también fueron seleccionados por ser los estándares más comunes en su respectivo ámbito, especialmente para el diseño y desarrollo de interfaces web.

AÑADIR VENTAJAS

13.2 Conclusiones personales

Participar en este proyecto me ha permitido adquirir y consolidar conocimientos en desarrollo web, sobre todo PHP, brindándome una base sólida para futuros trabajos similares. Las actividades se planificaron en ciclos de 2 semanas, y cada 14 días tenía reuniones con los tutores, estableciendo fechas límite para la entrega de tareas. Este enfoque me ayudó a aprender a trabajar bajo presión y a gestionar los plazos de manera efectiva.

Además, utilizar GitHub para la versión de proyectos facilitó la organización del proyecto, permitiendo un desarrollo más eficiente, ordenado y estructurado.

Otras de las cosas que han añadido conocimientos, es la utilización de las técnicas de patronaje y UML aprendidas tanto en Ingeniería de Software I como en Ingeniería de Software II, así como la gestión de un proyecto que se enseñó y llevó a cabo en la asignatura de Aprendizaje Basado en Proyectos.

14 Vías de trabajo futuro

Esta aplicación tiene varios puntos de mejora que serían interesantes añadir en futuras versiones.

Uno de los aspectos más relevantes es la adaptación del sistema para dispositivos iOS y/o Android, ya que son los dos sistemas operativos más utilizados en el mercado, superando por mucho a sus competidores. Esto facilitaría a los usuarios poder reservar un recurso desde cualquiera parte.

También incluir algún tipo de búsquedas, por ejemplo, para filtrar los recursos por centro, por tipo... También sería interesante poder buscar a los usuarios con algún tipo de barra de búsqueda.

Una mejora interesante sería la opción de añadir penalizaciones a los usuarios que hayan hecho una reserva de un recurso, pero no la hayan disfrutado. De esta manera los usuarios serían más empáticos con las demás personas, ya que al reservar ese recurso no queda disponible para el resto y si no lo utilizas otro usuario pierde la opción de hacerlo.

Otra mejora sustancial sería la opción de hacer el sistema multilingüaje. Actualmente solo está disponible en castellano, pero sería interesante hacer que el sistema esté en gallego, inglés, portugués, etc.

Menos importante sería incorporar mejoras en el diseño, como un modo nocturno u oscuro, que permita mostrar las interfaces con colores y tonos más oscuros, haciendo que sea más cómodo para la vista en ambientes con poca luz. Actualmente, son pocas las aplicaciones o sitios web que no ofrecen esta opción.

15 Referencias