



UNIVERSIDAD DE JAÉN
Nombre del Centro

Trabajo Fin de Grado

APLICACIÓN WEB PARA EL MANTENIMIENTO DEL SOFTWARE DE LOS LABORATORIOS DEL DPTO. DE INFORMÁTICA

Alumno: Tamara Morales Comino

Tutor: Prof. D. Manuel García Vega
Dpto: Informática

Septiembre, 2018

Índice

1. Introducción	4
1.1. Propósito	4
1.2. Objetivos	4
1.3. Metodología	5
2. Análisis	6
2.1. Análisis preliminar	6
2.2. Requisitos iniciales	6
2.3. Planificación del Proyecto	7
2.3.1. Estructura del plan de trabajo	7
2.4. Estudio de viabilidad	10
2.5. Análisis de la interfaz de usuario	11
2.5.1. Introducción a los prototipos en papel	11
2.5.2. Prototipo en papel 1	12
2.5.3. Prototipo en papel 2	25
2.5.4. Prototipo en papel 3	45
2.5.5. Recorridos cognitivos	72
2.6. Requisitos del sistema finales	77
2.7. Propuesta de solución	79
2.8. Modelo de dominio	80
3. Diseño	82
3.1. Diagrama Entidad-Relación	82
3.1.1. Definición de las tablas	84
3.2. Diagrama de clases	87
3.3. Modelo de Casos de Uso	89
3.3.1. Diagramas de caso de uso	91
3.3.2. Especificaciones de los casos de uso	96
3.4. Diagramas de secuencia	119

4. Implementación	130
4.1. Arquitectura	130
4.1.1. Arquitectura Cliente-Servidor	130
4.1.2. Arquitectura Modelo-Vista-Controlador	131
4.2. Detalles sobre la implementación	131
4.2.1. WAMPServer	132
4.2.2. CodeIgniter	133
4.3. Interfaz de usuario	135
5. Conclusiones	138
5.1. Mejoras y trabajos futuros	139
Bibliografía	140
I. Manual de instalación	142
II. Manual de Usuario	144

Capítulo 1

Introducción

1.1. Propósito

El propósito de este Trabajo Fin de Grado es analizar, diseñar y desarrollar una aplicación web para gestionar el mantenimiento del software de los laboratorios del Dpto. de Informática.

Dicha aplicación nos permitirá una gestión del software que hay instalado en dichos laboratorios además de llevar un control del software que falta por instalar a petición de los profesores mediante la misma aplicación.

1.2. Objetivos

Realizar un estudio sobre el procedimiento existente en la Universidad de Jaén para el mantenimiento del software de los laboratorios del Dpto. de Informática. Este estudio se realizará mediante una reunión con los técnicos de laboratorio del Dpto. de Informática además de con profesores de dicho departamento.

Diseñar una aplicación web para la gestión del software de los laboratorios del Dpto. de Informática. Este proceso consistirá en realizar un análisis de los requisitos de la aplicación mediante el uso de prototipos en papel.

Implementar un prototipo de aplicación web para el proyecto, para ello se estudiarán las posibles herramientas y técnicas a seguir.

1.3. Metodología

Para el desarrollo de este TFG se va a seguir técnicas usadas en la disciplina de la IPO (Interacción Persona-Ordenador). La IPO se define como:

“... Disciplina relacionada con el diseño, evaluación e implementación de sistemas informáticos interactivos para el uso de seres humanos, y con el estudio de los fenómenos más importantes con los que está relacionado [1].”

Para la evaluación del proyecto haremos uso del prototipado. Los prototipos se definen como:

... Documentos, diseños o sistemas que simulan o tienen implementadas partes del sistema final. El prototipo es una herramienta muy útil para hacer participar al usuario en el desarrollo y poder evaluar el producto ya en las primeras fases del diseño (modelo del ciclo de vida centrado en el usuario) [1].

En nuestro caso utilizaremos la técnica de prototipado conocida como prototipo en papel que se explicará más adelante.

La figura 1.1 representa dicho modelo de vida basado en prototipos. Como podemos ver en todo el proceso el usuario está presente, desde el análisis de requisitos hasta implementación, para recoger sus necesidades y mejorar su utilización. De esta forma, el software final obtenido debe de satisfacer completamente al usuario.

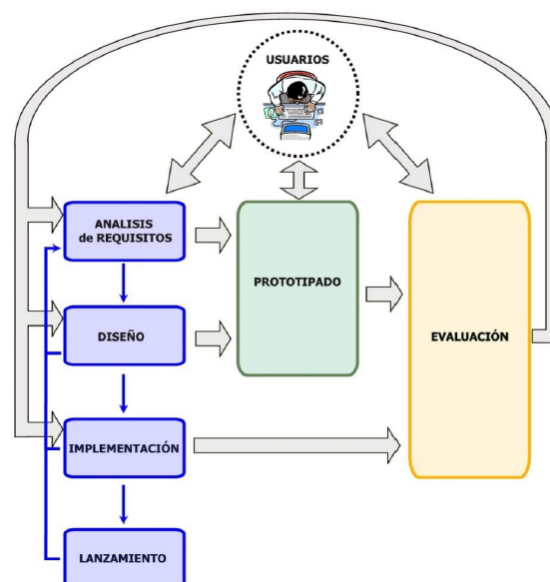


Figura 1.1: Modelo del ciclo de vida centrado en el usuario.

Capítulo 2

Análisis

2.1. Análisis preliminar

Partimos del Dpto. de Informática de la Universidad de Jaén que realiza un mantenimiento y gestión del software que se usa en los laboratorios de dicho departamento y debemos de proporcionar una solución al gusto de los diferentes usuarios.

La gestión del software la realizan los técnicos de laboratorio, que necesitan saber qué software está instalado o falta por instalar en los laboratorios. La petición de qué software se debe instalar la realizan los profesores del departamento, que solicitan para las asignaturas en las que imparten clase el software necesario.

2.2. Requisitos iniciales

Los requisitos obtenidos del análisis preliminar son:

1. Tres tipos de usuarios: Administrador (para mantener el sistema), Técnico (permite ver el software instalado o que falta por instalar) y Profesor (permite asignar software a diferentes asignaturas para que sea instalado).
2. Tres vistas diferentes adecuadas a las operaciones que puede realizar cada usuario.
3. El sistema debe tener un acceso mediante una cuenta con contraseña para evitar accesos no autorizados.
4. Mantener una lista de asignaturas, laboratorios y software en la base de datos.

2.3. Planificación del Proyecto

Una vez establecidos unos requisitos iniciales para enfocar mejor nuestro proyecto es necesario identificar el conjunto de tareas para organizar mejor nuestro proyecto. Para ello debemos identificar dichas tareas y subtareas, asignar al personal responsable de cada tarea y estimar la duración de cada una de estas tareas.

Para que la planificación del proyecto sea eficiente, se deben cumplir una serie de requisitos:

1. Todas las tareas identificadas deben aparecer en el diagrama.
2. El reparto de esfuerzo para realizar cada tarea debe de ser equitativo.
3. El diagrama debe de reflejar la relación existente entre las diferentes tareas, de forma que consultando el diagrama se pueda saber cuándo empieza y finaliza cada tarea.
4. Se deben establecer “hitos”, que marcarán el desarrollo del proyecto.

2.3.1. Estructura del plan de trabajo

La jerarquía resultante sería la siguiente:

1. Puesta al día bibliográfica.
 - a) Estudio de las Tecnologías disponibles para Desarrollo Web.
2. Análisis de la Interfaz de Usuario.
 - a) Realización de Prototipos en papel.
 - b) Evaluación de Prototipos en papel.
 - 1) Recorridos cognitivos.
 - c) Análisis de Requisitos.

HITO 1: Fase de Análisis.

3. Desarrollo del prototipo de la Aplicación Web.
 - a) Diseño del prototipo de Aplicación Web.
 - 1) Diseño de la Arquitectura del prototipo.

HITO 2: Diseño del prototipo de la Aplicación Web.

b) Implementación del prototipo de la Aplicación Web.

- 1) Implementación de la Interfaz de Usuario
- 2) Implementación de la Arquitectura del prototipo.

HITO 3: Implementación del prototipo de la Aplicación Web.

4. Evaluación del prototipo de la Aplicación Web.

HITO 4: Evaluación del prototipo de la Aplicación Web.

5. Elaboración de la Documentación.

- a) Completar la memoria.
- b) Realizar manuales de instalación y de uso.

HITO 5: Elaboración de la Documentación.

La figura 2.1 muestra la división del plan de trabajo en tareas. Estas a su vez también están divididas en diferentes subtareas.

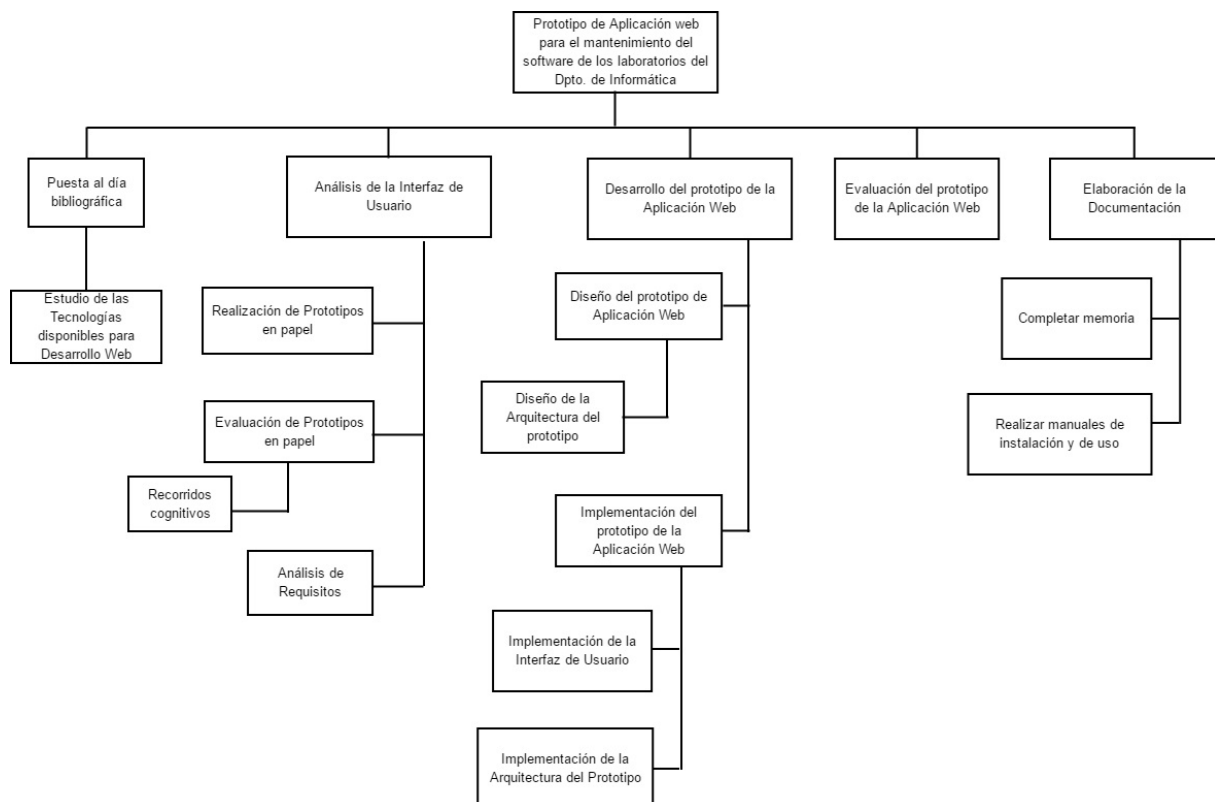


Figura 2.1: Estructura del plan de trabajo

A continuación se asignará el esfuerzo a cada una de las tareas y subtareas anteriores.

Tarea	Duración (en días)	Encargado
Puesta al día bibliográfica		
Estudio de las Tecnologías disponibles para Desarrollo Web	5	Analista
Análisis de la Interfaz de Usuario		
Realización de Prototipos en papel	10	Analista
<i>Evaluación de Prototipos en papel</i>		
Recorridos cognitivos	5	Analista
Análisis de Requisitos	2	Analista/ Programador
Desarrollo del prototipo de la Aplicación Web		
<i>Diseño del prototipo de Aplicación Web</i>		
Diseño de la Arquitectura del prototipo	10	Analista
<i>Implementación del prototipo de la Aplicación Web</i>		
Implementación de la Interfaz de Usuario	25	Programador
Implementación de la Arquitectura del prototipo	40	Programador
Evaluación del prototipo de la Aplicación Web	10	Programador
Elaboración de la Documentación		
Completar la memoria	30	Analista
Realizar manuales de instalación y de uso	10	Analista
Esfuerzo total	142	

Tabla 2.1: Estimación del esfuerzo

Por último, sólo nos queda realizar el Diagrama de Gantt para reflejar dicha planificación del proyecto. Este diagrama es una forma fácil y rápida de visualizar las tareas, además, nos permite realizar un seguimiento del progreso de cada una de los hitos del proyecto.

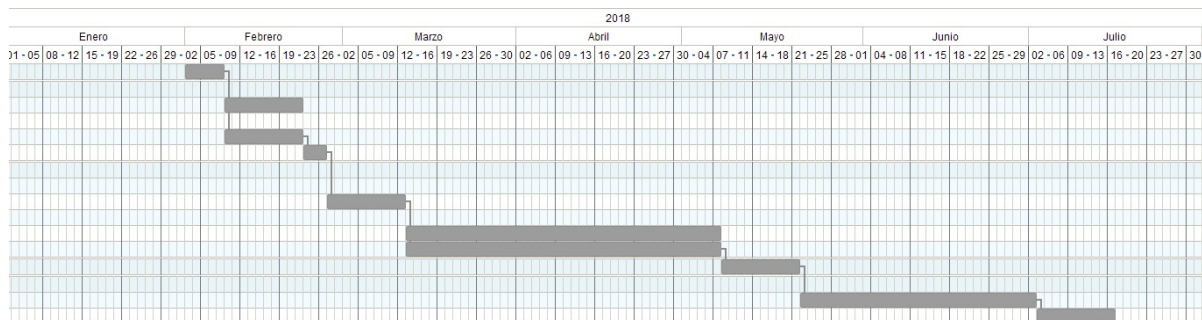


Figura 2.2: Diagrama de Gantt

2.4. Estudio de viabilidad

El siguiente paso que debemos dar es realizar un estudio de viabilidad para realizar una estimación de los costes de desarrollo y explotación (si los hubiese) del proyecto. Estos gastos nos ayudarán a conocer si el proyecto que queremos llevar a cabo es realista y se puede desarrollar.

A continuación, se mostrarán una serie de tablas dónde se detallarán los diferentes tipos de gastos necesarios para el proyecto.

Herramienta o Licencia Software	Precio	Duración del proyecto
Microsoft Windows 7 Professional 64bits Service Pack 1	149 €	6 meses
WampServer	Gratuito	6 meses
CodeIgniter	Gratuito	6 meses
PencilProject	Gratuito	6 meses

Tabla 2.2: Estudio de las herramientas y licencias software utilizadas

La tabla 2.2 representa los gastos relacionados con herramientas y licencias software. Estos gastos no son elevados dado que casi todos poseen licencias gratuitas. Dichas herramientas han sido escogidas porque nos permitirán desarrollar cualquier aplicación web.

Equipo informático	Precio	Vida estimada	Precio ponderado
MSI GE62 6QF	1100€	7 años	157€

Tabla 2.3: Estudio del equipo informático utilizado

La tabla 2.3 muestra el equipo informático utilizado para realizar el proyecto.

Personal	Salario/semana	Salario/mes	Salario/hora	Días trabajo	Horas/Día	Total horas	Total estimado
Analista	357,80€	1.229,95€	6,38€	70	8	560	3.578€
Programador	337,30€	1.159,49€	6€	77	8	616	3.696€

Tabla 2.4: Estudio salarios personal

Para el desarrollo del proyecto se necesitan dos tipos de personal, un analista y un programador. En la tabla 2.4 aparece el estudio realizado para estos tipos de personal. El salario/mes y salario/semana para cada tipo de personal han sido obtenidos del BOE [2].

Finalmente podemos resumir el total de los gastos en la tabla 2.5. Como podemos observar en dicha tabla, el proyecto supondría un desembolso final por parte de la empresa que lo contratase de 8.717€.

Tipo de gasto	Precio
Software	149€
Hardware	157€
Nóminas	7.274€
Beneficios(15 % de los gastos)	1.137€
TOTAL	8.717€

Tabla 2.5: Gastos totales

2.5. Análisis de la interfaz de usuario

Como ya se ha detallado antes en el apartado “Planificación del Proyecto” comenzaremos con la etapa de análisis de la interfaz de usuario.

La interfaz es un apartado complicado en cualquier aplicación. En nuestro caso, al seguir los principios de la IPO, el análisis de la interfaz se realizará mediante el uso del prototipado. Gracias a esta herramienta podemos facilitar este proceso, de forma que podemos identificar los requisitos del sistema a la vez que definimos y evaluamos el diseño de la interfaz. Ahorrando así tiempo y ajustándose mejor a los requisitos de los usuarios que usarán finalmente sistema.

En los siguientes apartados se explicará en qué consiste un prototipo en papel y se mostrarán los diferentes prototipos en papel realizados. Por último, se mostrarán todos los requisitos finalmente obtenidos tras el análisis de los diferentes prototipos.

2.5.1. Introducción a los prototipos en papel

El tipo de prototipo que vamos a utilizar es el prototipo en papel. Antes de comenzar con las diferentes propuestas realizadas es necesario saber qué es un prototipo en papel.

“... Este tipo de prototipo se basa en la utilización de papel, tijeras, lápiz o instrumentos que se puedan utilizar para describir un diseño en un papel. Este sistema nos permite una gran velocidad y flexibilidad.” [1]

Para poder simular las diferentes interacciones que vamos a realizar con el sistema, en cada página aparecerán anotaciones en azul y cursiva para cada una de las posibles interacciones.

En este caso el prototipo no se ha realizado en papel directamente, se ha utilizado una herramienta software denominada *Pencil Project* [3]. Este software es una herramienta de prototipado de código abierto basada en *drag and drop* (arrastrar y soltar).

Se ha optado por esta herramienta porque es intuitiva y muy fácil de usar. Además, incluye diferentes métodos de exportación para nuestros proyectos como exportar a un archivo pdf o incluso crear una página web que simula el prototipo.

2.5.2. Prototipo en papel 1

El prototipo en papel inicial contiene una primera propuesta para la aplicación web que contiene diferentes funcionalidades a incluir en dicha aplicación. Para comenzar tenemos diversos roles (tipos de usuarios) que pueden existir, cada uno de ellos accederá a una página diferente:

- **Administrador:** permite la gestión (creación, modificación y eliminación) de los datos almacenados (usuarios, programas, laboratorios y asignaturas).
- **Profesor:** permite asignar programas a distintas asignaturas.
- **Técnico:** muestra una lista de los programas a instalar.

A continuación, se mostrará el primer prototipo en papel, cada figura representa una “página” del prototipo.

1

Correo electrónico

usuario@ujaen.es

Contraseña

Entrar

Si usuario = administrador ir a 2
Si usuario = profesor ir a 3
Si usuario = técnico ir a 4

Página 1 - Prototipo en papel 1

2

Usuarios 2

Programas 5

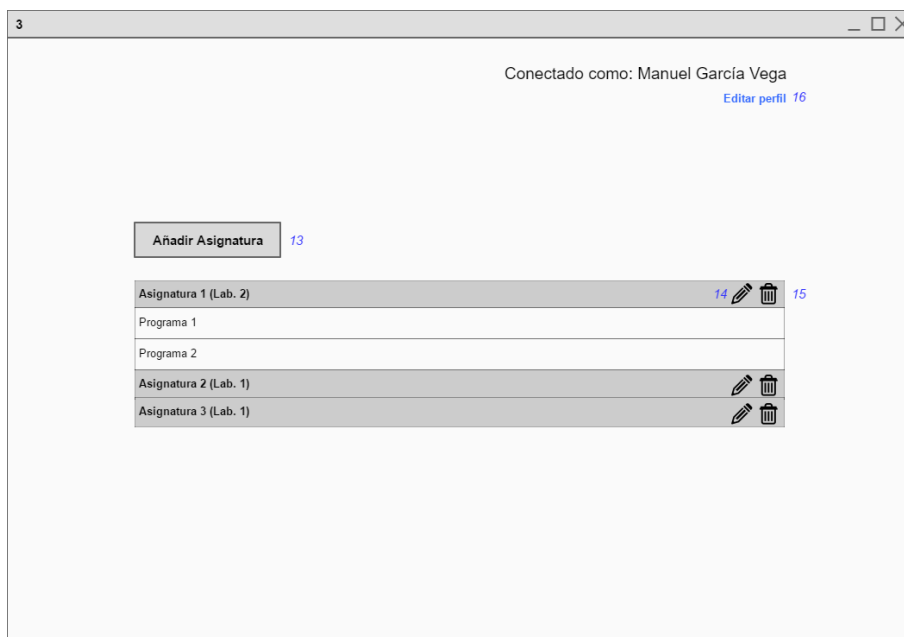
Laboratorios 6

Asignaturas 7

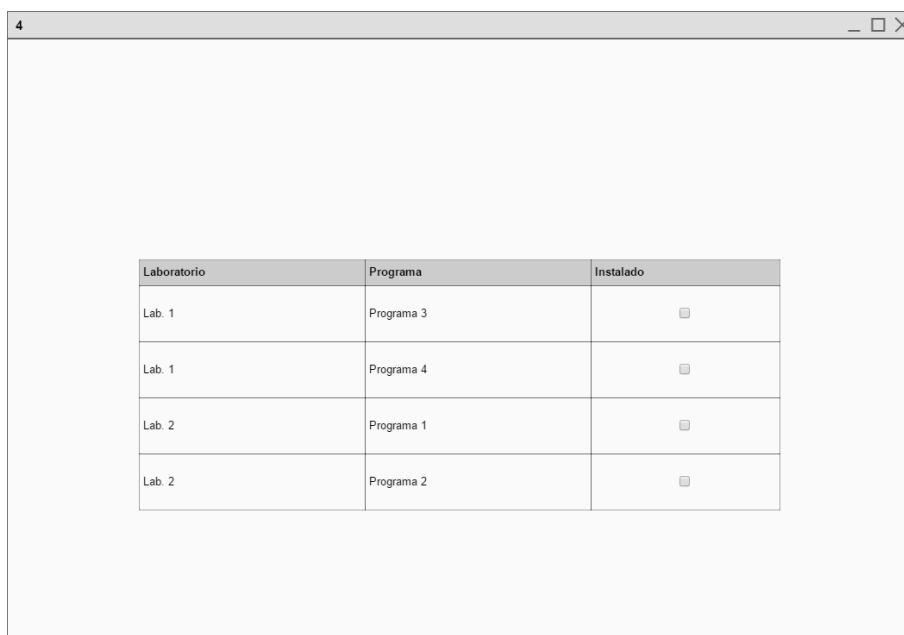
Añadir usuario 8

Email	Nombre	Apellidos	Telefono	Rol
tmc00009@red.ujaen.es	Tamara	Morales Comino	123456789	admin
mgarcia@ujaen.es	Manuel	García Vega	953212465	prof

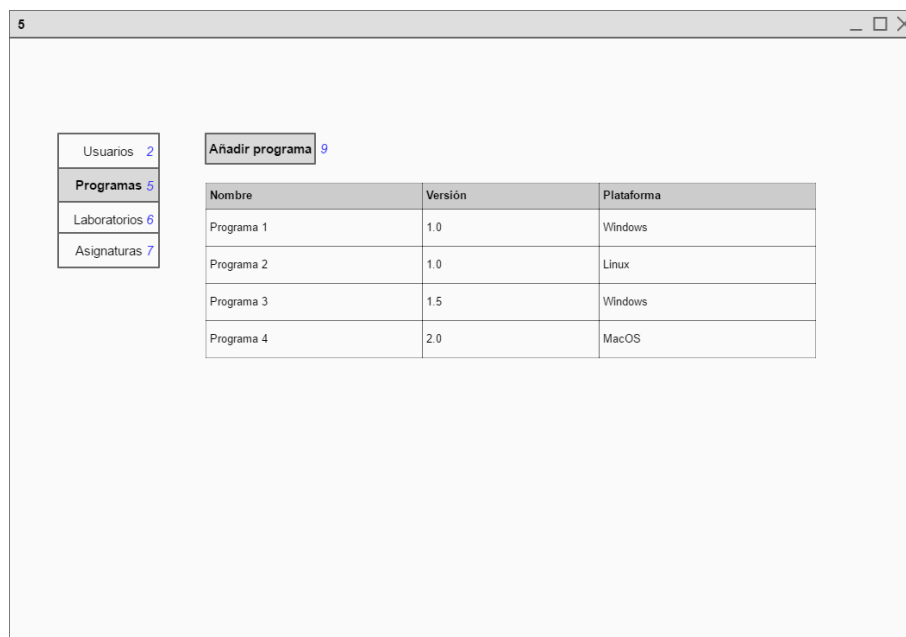
Página 2 - Prototipo en papel 1



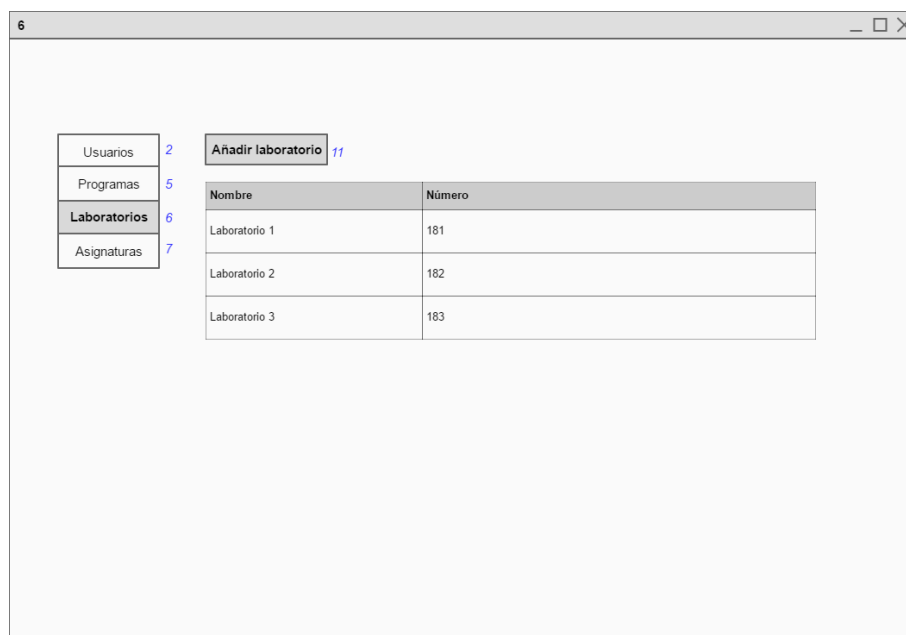
Página 3 - Prototipo en papel 1



Página 4 - Prototipo en papel 1



Página 5 - Prototipo en papel 1



Página 6 - Prototipo en papel 1

7

Usuarios	2
Programas	5
Laboratorios	6
Asignaturas	7

Añadir asignatura 12

Nombre
Asignatura 1
Asignatura 2
Asignatura 3

Página 7 - Prototipo en papel 1

8

Usuarios
Programas
Laboratorios
Asignaturas

Añadir usuario

Email
tmc00009@red.ujae
mgarcia@ujaen.es

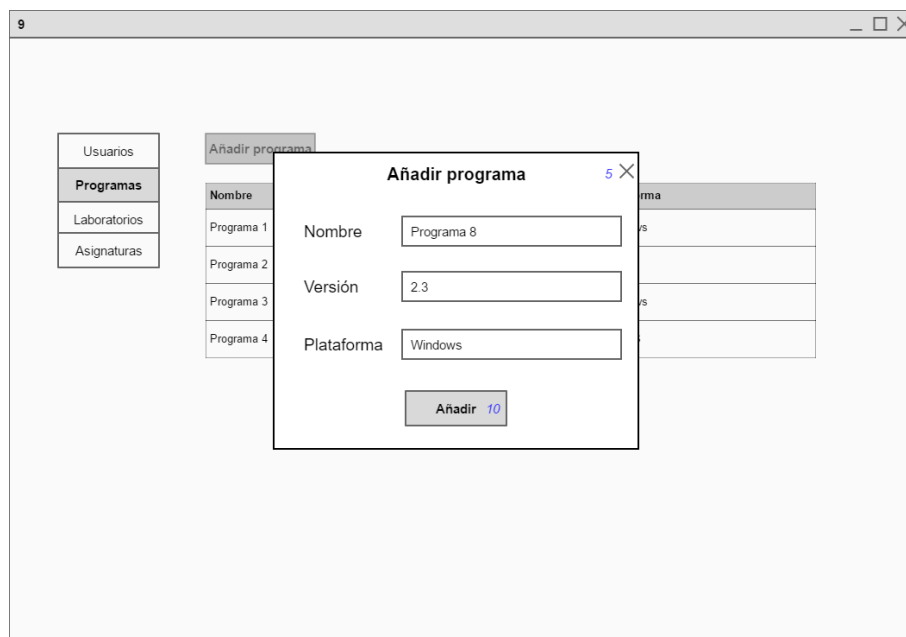
Añadir usuario 2 X

Correo electrónico

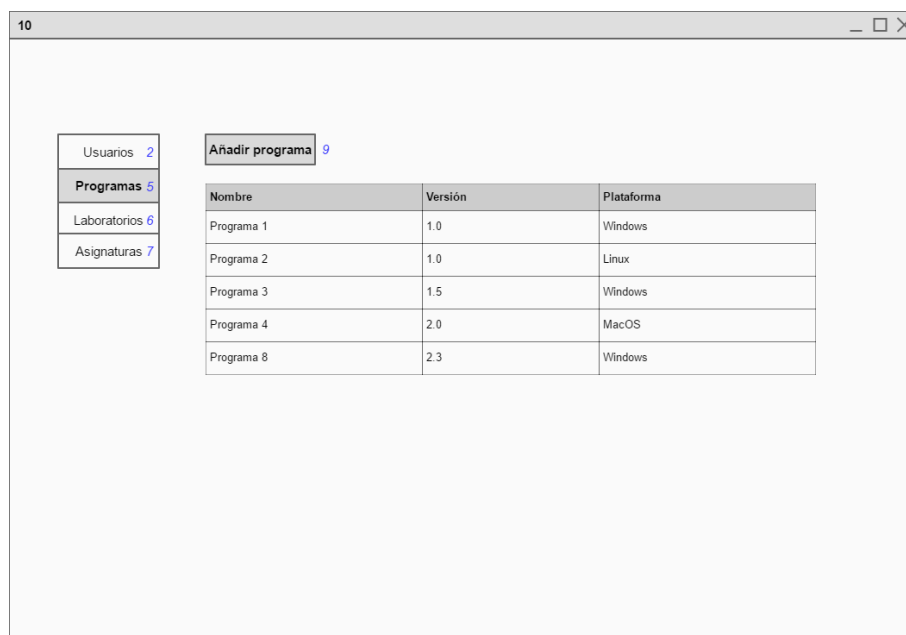
Añadir

Telefono	Rol
123456789	admin
953212465	prof

Página 8 - Prototipo en papel 1



Página 9 - Prototipo en papel 1



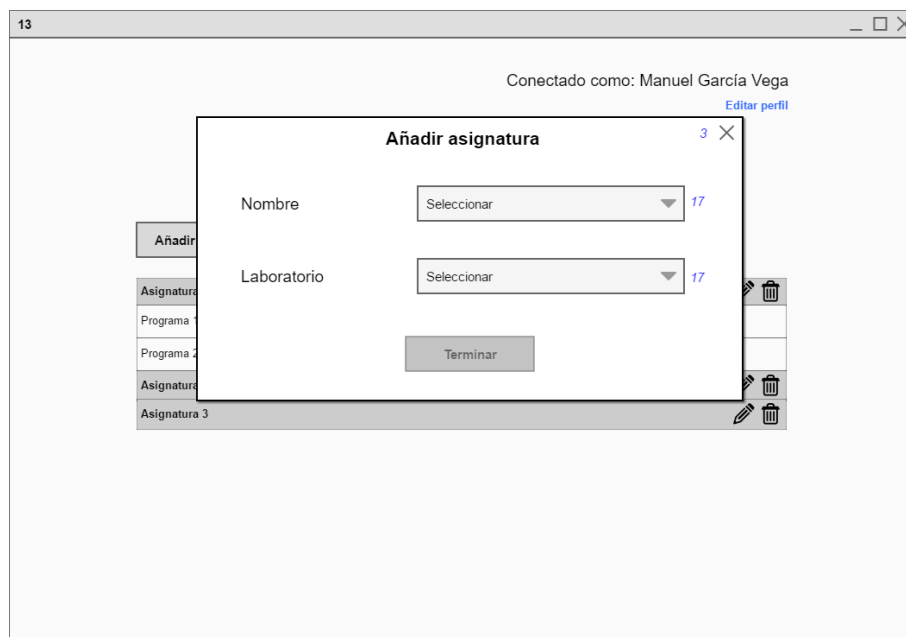
Página 10 - Prototipo en papel 1

The screenshot shows a web application interface. On the left is a sidebar menu with four items: 'Usuarios', 'Programas', 'Laboratorios' (which is highlighted), and 'Asignaturas'. In the center, a modal window titled 'Añadir laboratorio' is open, featuring two input fields labeled 'Nombre' and 'Número', and an 'Añadir' button at the bottom. To the right of the modal is a table with three rows, each labeled 'Laboratorio 1', 'Laboratorio 2', and 'Laboratorio 3' respectively. The table has several empty columns for data entry.

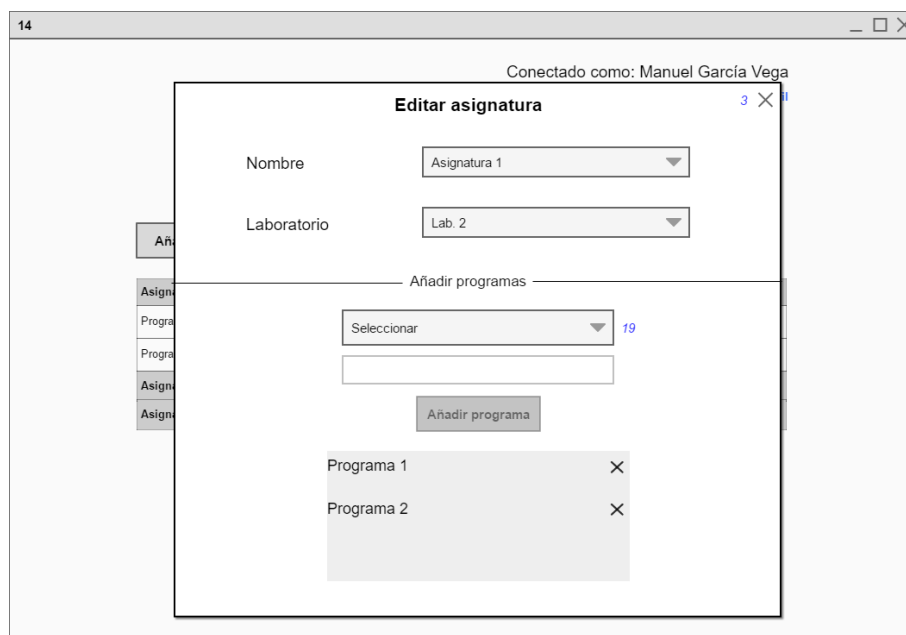
Página 11 - Prototipo en papel 1

The screenshot shows a web application interface. On the left is a sidebar menu with four items: 'Usuarios', 'Programas', 'Laboratorios', and 'Asignaturas' (which is highlighted). In the center, a modal window titled 'Añadir asignatura' is open, featuring one input field labeled 'Nombre' and an 'Añadir' button at the bottom. To the right of the modal is a table with three rows, each labeled 'Asignatura 1', 'Asignatura 2', and 'Asignatura 3' respectively. The table has several empty columns for data entry.

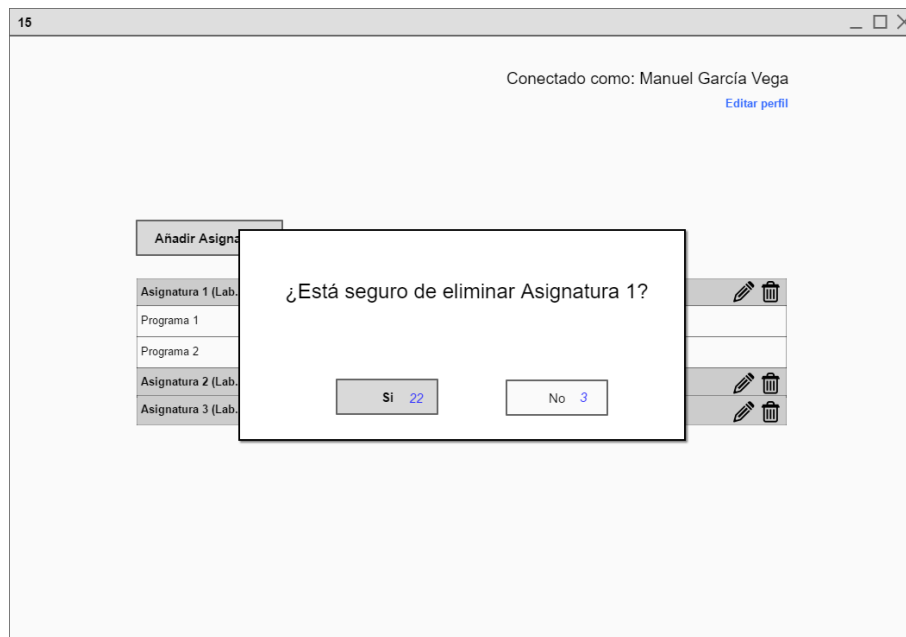
Página 12 - Prototipo en papel 1



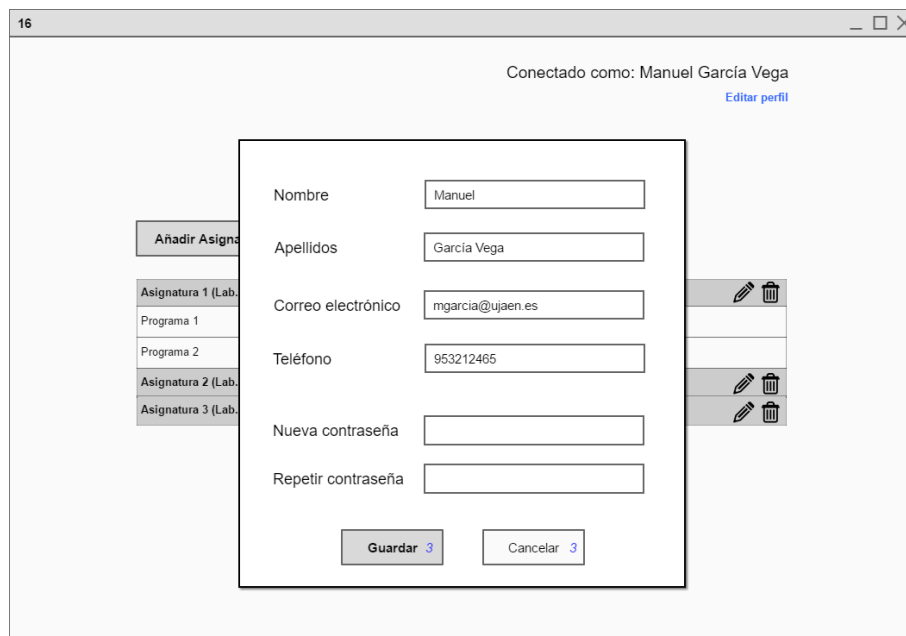
Página 13 - Prototipo en papel 1



Página 14 - Prototipo en papel 1



Página 15 - Prototipo en papel 1



Página 16 - Prototipo en papel 1

The screenshot shows a window titled 'Añadir asignatura' with a close button (X) and a tab indicator '3'. Inside the window, there are two dropdown menus: 'Nombre' with 'Asignatura 4' selected, and 'Laboratorio' with 'Lab. 2' selected. Below these is a 'Terminar' button with a blue '18' next to it. To the left of the window is a vertical sidebar with a list of items: 'Añadir', 'Asignatura', 'Programa', 'Programa 2', 'Asignatura', and 'Asignatura 3'. To the right of the window is a vertical toolbar with icons for editing and deleting.

Página 17 - Prototipo en papel 1

The screenshot shows a window titled 'Asignatura' with a close button (X) and a tab indicator '3'. Inside the window, there are two dropdown menus: 'Nombre' with 'Asignatura 4' selected, and 'Laboratorio' with 'Lab. 2' selected. Below these is a section titled 'Añadir programas' which contains a 'Seleccionar' dropdown menu, an empty text input field, and an 'Añadir programa' button. At the bottom of the window is a large, empty rectangular box. To the left of the window is a vertical sidebar with a list of items: 'Añ', 'Asign', 'Progra', 'Progra', 'Progra', 'Asign', and 'Asign'. To the right of the window is a vertical toolbar with icons for editing and deleting.

Página 18 - Prototipo en papel 1

19

Editar asignatura 3 X

Nombre

Laboratorio

Añadir programas

Programa 4

Otro 20

Programa 1 X

Programa 2 X

Página 19 - Prototipo en papel 1

20

Editar asignatura 3 X

Nombre

Laboratorio

Añadir programas

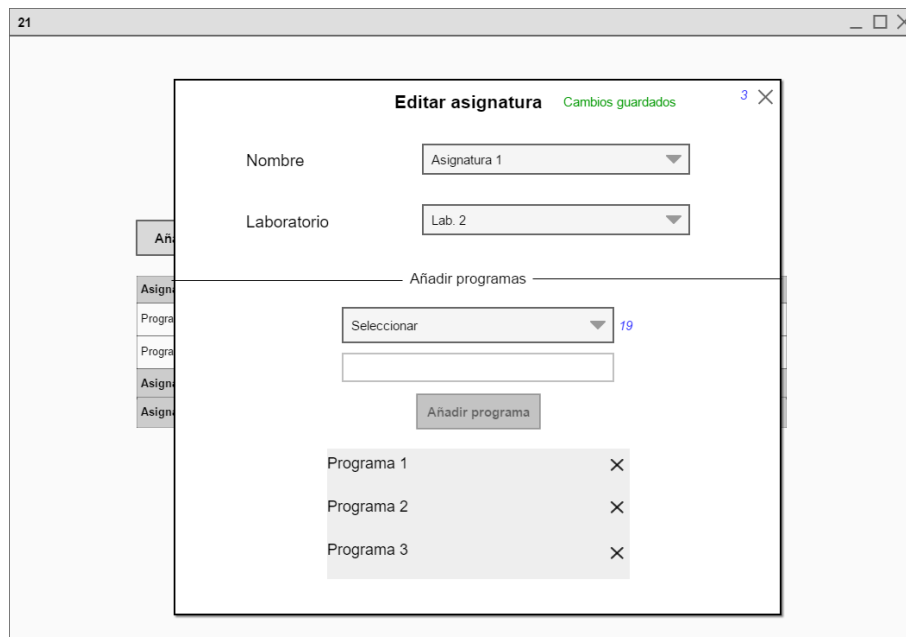
Programa 3

Añadir programa 21

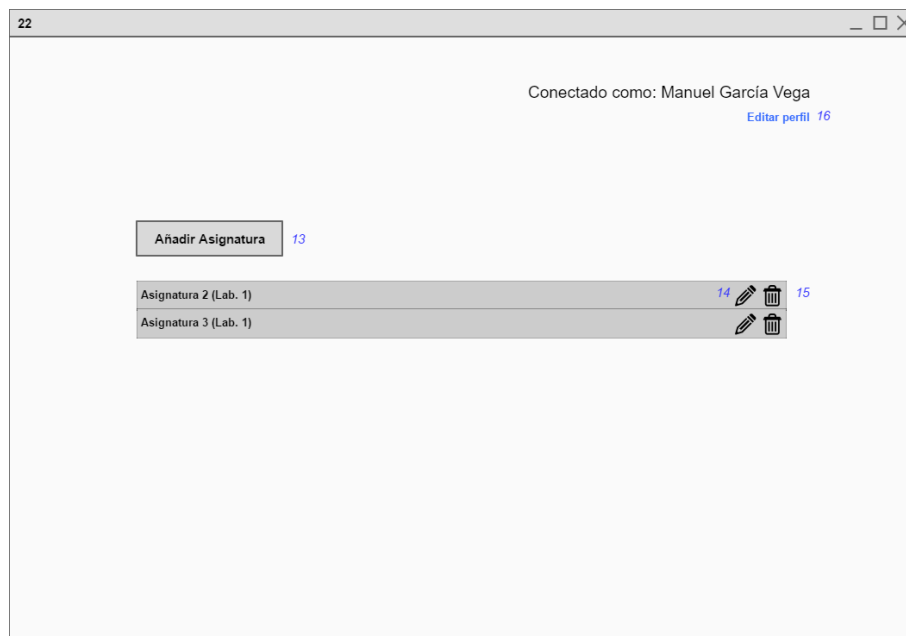
Programa 1 X

Programa 2 X

Página 20 - Prototipo en papel 1



Página 21 - Prototipo en papel 1



Página 22 - Prototipo en papel 1

Tras un análisis del prototipo con los usuarios se decidió por aplicar los siguientes cambios en él:

1. De los tres tipos de roles de usuario que hay (Administrador, Profesor y Técnico) se pasará a tener dos (Administrador y Profesor) combinando los roles de Administrador y Técnico.
2. El software (en el prototipo "Programas") pasará a nombrarse "Software" y debe contener los siguientes atributos:
 - Nombre.
 - Versión.
 - Plataforma.
 - Idioma.
 - Licencia.
 - Campo de observaciones, para escribir aspectos más concretos del programa, por ejemplo, los módulos a instalar de dicho programa.
 - Especificar si el software es básico o no. El software marcado como básico será el que se instale siempre en todos los laboratorios, por ejemplo, navegadores o software de edición de texto.
3. Se debe de controlar la fecha de instalación máxima para el software y la fecha en la que se crea la solicitud para la instalación.
4. Las asignaturas deben contener los siguientes atributos:
 - Curso y cuatrimestre al que pertenecen.
 - El laboratorio no debe ser elegido por el profesor, debe ser un atributo de la asignatura, modificable, que asignen los administradores.
5. Incluir otro método para pedir software denominado "Petición puntual" para poder realizar peticiones de software para distintos eventos que no se identifican como asignaturas. En este método se deben especificar:
 - Nombre para identificar la petición.
 - Fecha límite para instalar el software.
 - Laboratorio. En este caso el laboratorio si debe ser especificado (no se debe especificar en las asignaturas) ya que sí es responsabilidad del usuario que realiza la petición indicar dónde instalar dicho software.

2.5.3. Prototipo en papel 2

1

Correo electrónico

usuario@ujaen.es

Contraseña

Entrar

Si usuario = administrador ir a 2
Si usuario = profesor ir a 25

Página 1 - Prototipo en papel 2

2

Conectado como: Tamara Morales Comino

[Editar perfil 8](#)

Orden: Asig. A-Z 9

Tareas pendientes	2
Tareas finalizadas	3
Usuarios	4
Software	5
Laboratorios	6
Asignaturas	7

Asignatura 1 Lab. 2

Solicitante: Manuel García Vega (mgarcia@ujaen.es)
Fecha de solicitud: 4/12/2017
Fecha límite: 10/1/2018

Programa 1

- Versión: 1.0
- Idioma: Español
- Plataforma: Windows
- Licencia: Gratuita
- Observaciones: Descarga en la dirección [www.programa1/download.com](#)

Programa 2

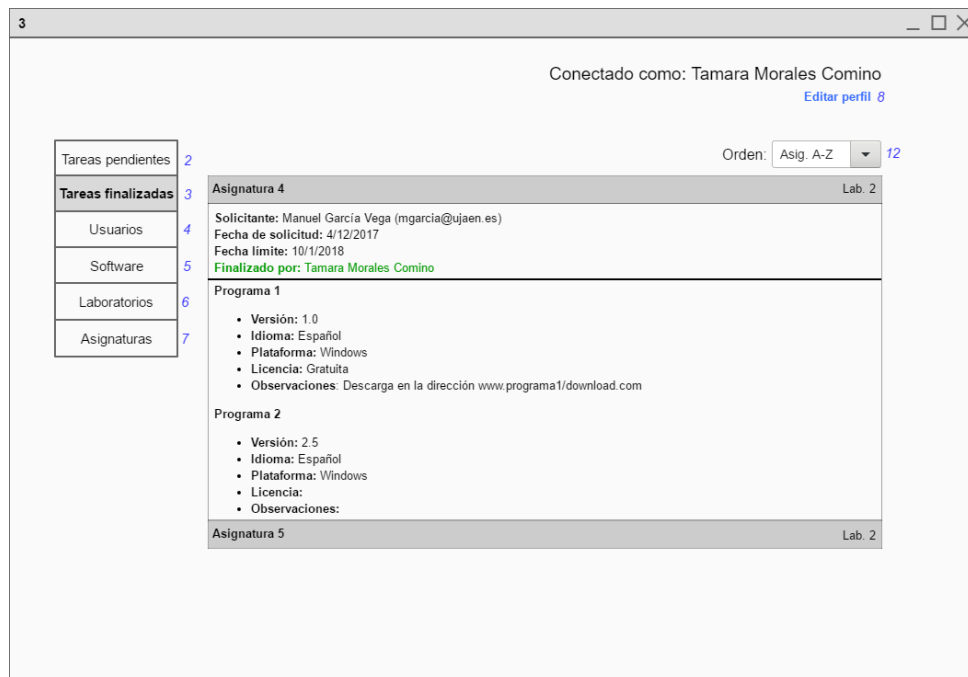
- Versión: 2.5
- Idioma: Español
- Plataforma: Windows
- Licencia:
- Observaciones:

Asignatura 2 Lab. 1

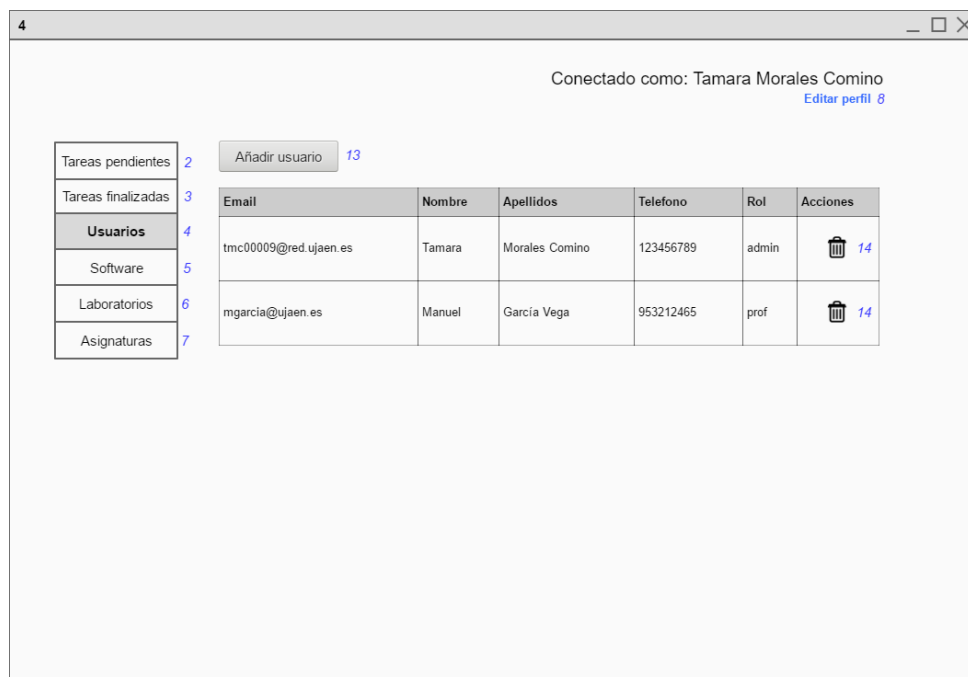
Asignatura 3 Lab. 1

Finalizar 10

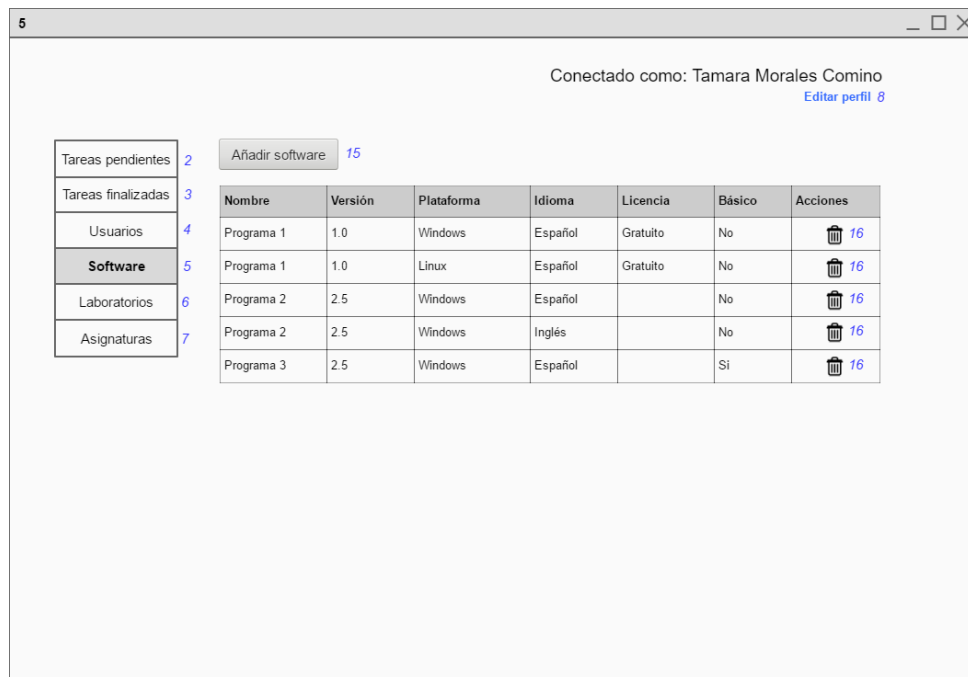
Página 2 - Prototipo en papel 2



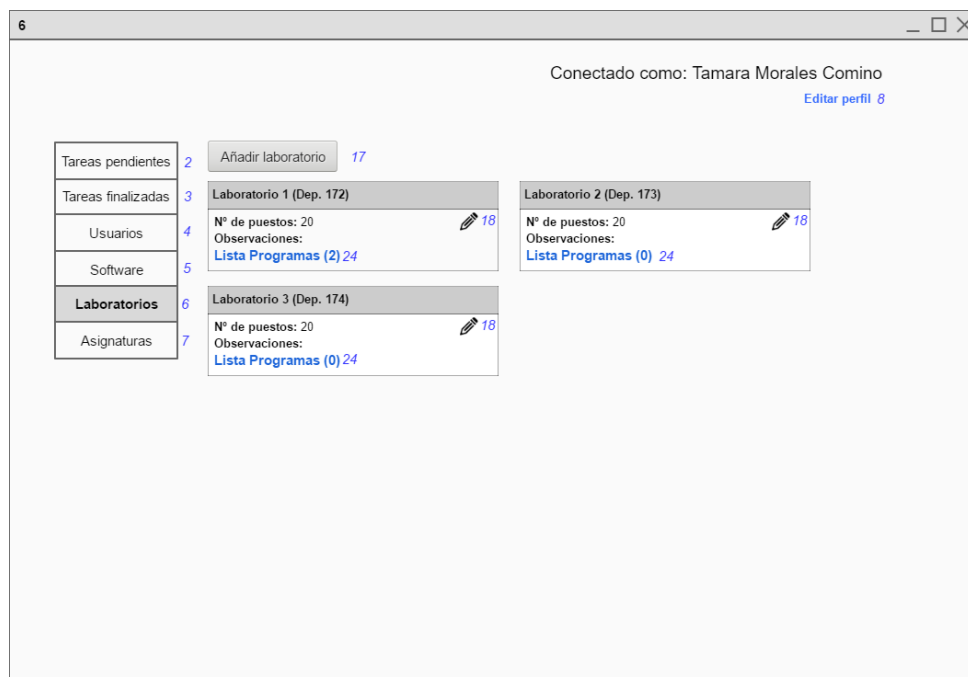
Página 3 - Prototipo en papel 2



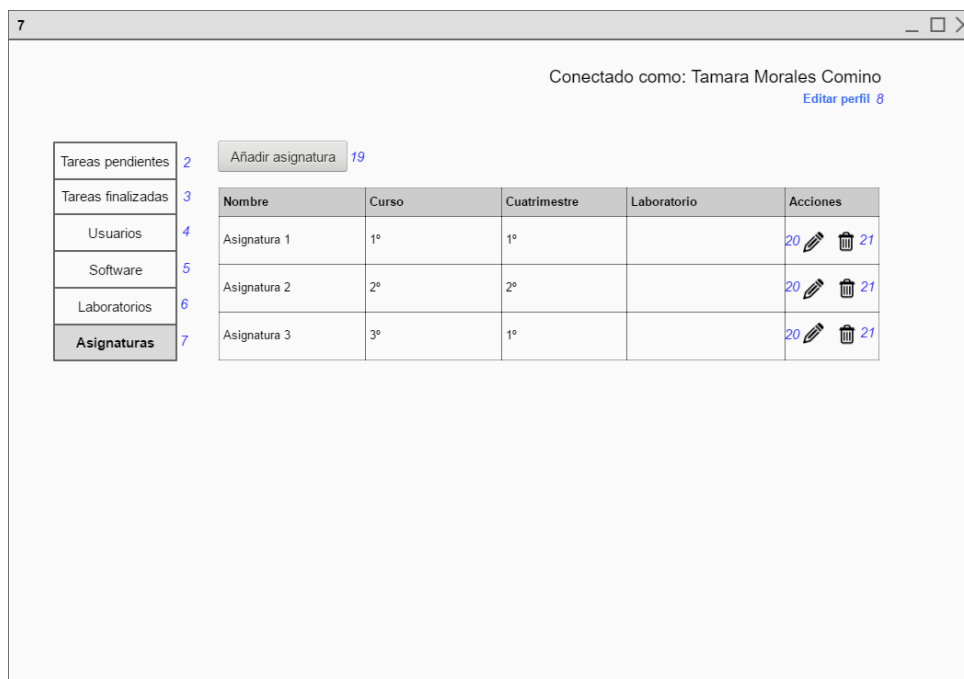
Página 4 - Prototipo en papel 2



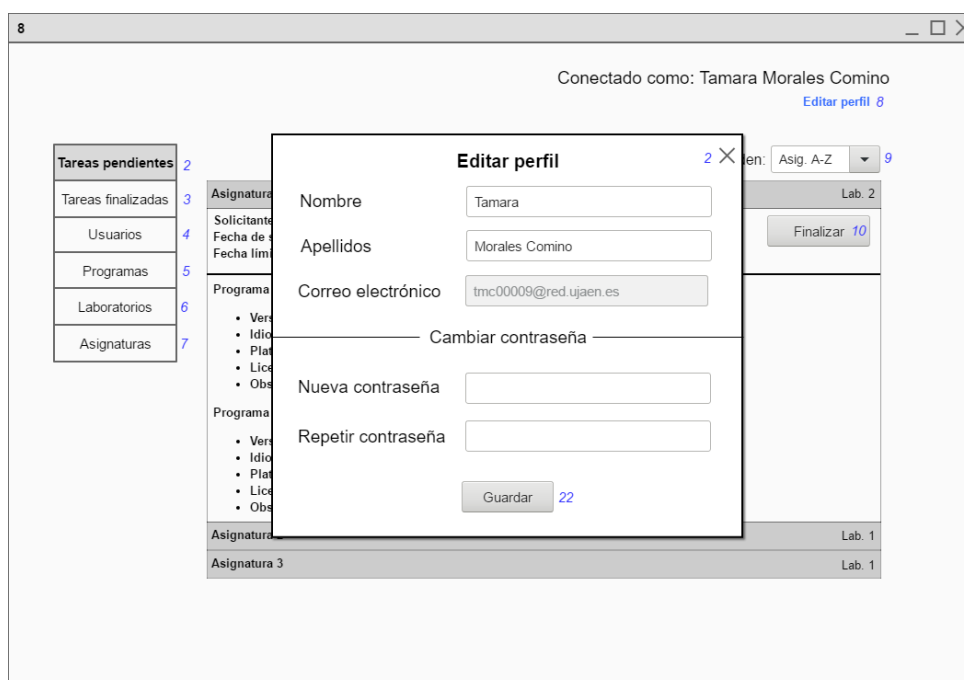
Página 5 - Prototipo en papel 2



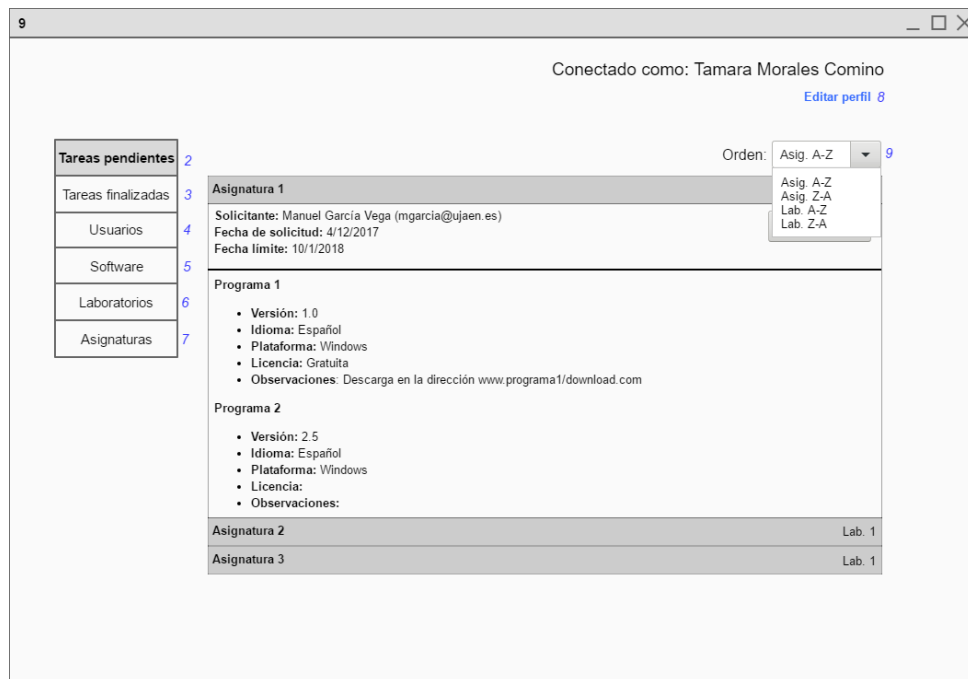
Página 6 - Prototipo en papel 2



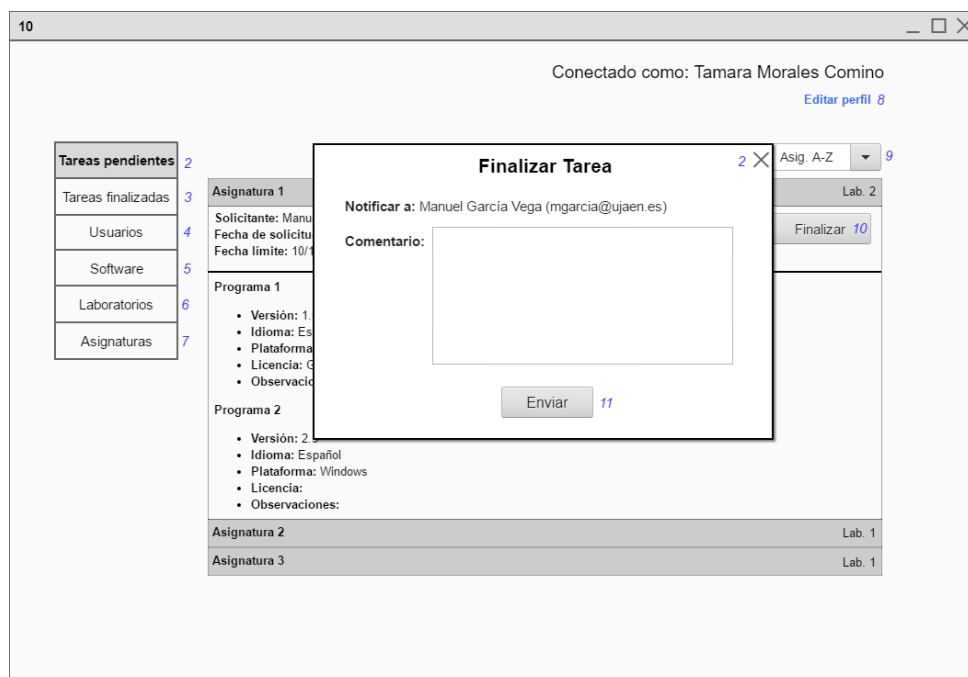
Página 7 - Prototipo en papel 2



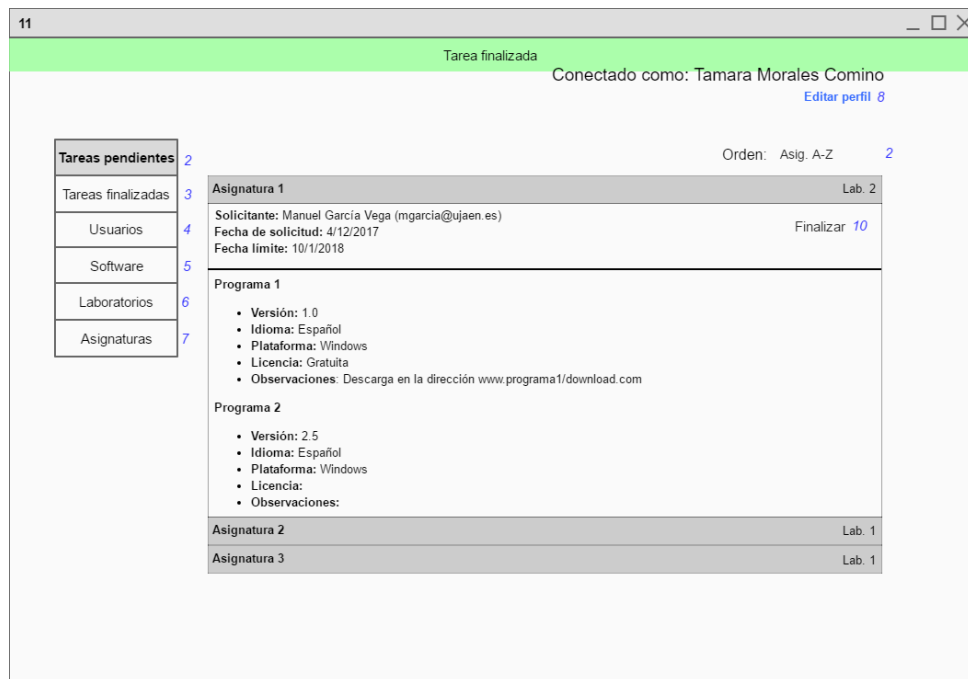
Página 8 - Prototipo en papel 2



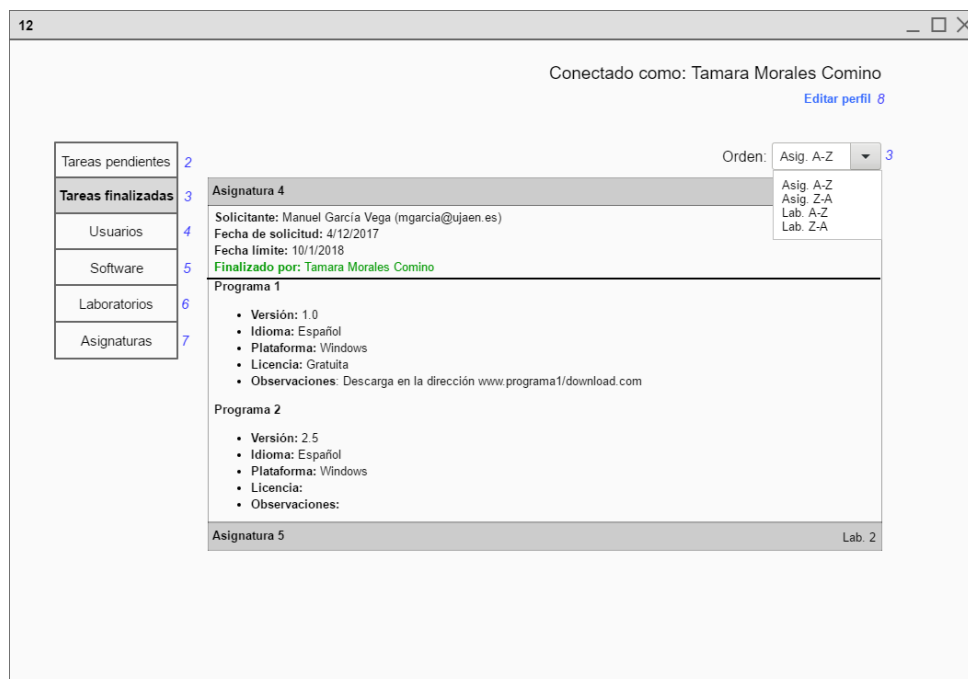
Página 9 - Prototipo en papel 2



Página 10 - Prototipo en papel 2



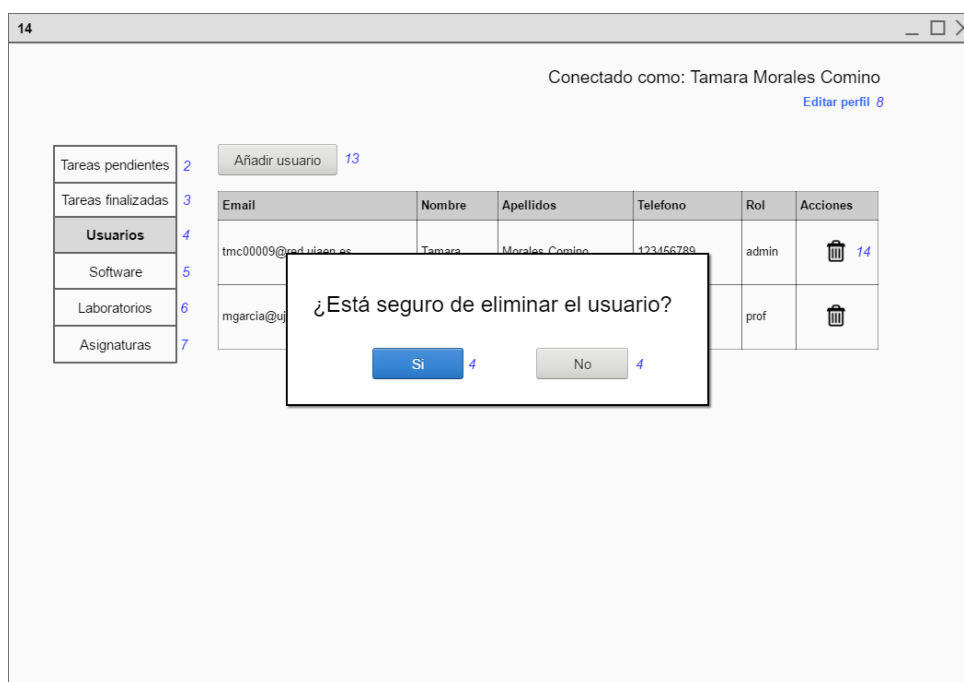
Página 11 - Prototipo en papel 2



Página 12 - Prototipo en papel 2



Página 13 - Prototipo en papel 2



Página 14 - Prototipo en papel 2

15

Conectado como: Tamara Morales Comino

[Editar perfil](#) 8

Tareas pendientes 2

Tareas finalizadas 3

Usuarios 4

Software 5

Laboratorios 6

Asignaturas 7

Añadir software 15

Nombre

Programa 1

Programa 2

Programa 3

Programa 4

Programa 5

Nombre

Programa 1

Programa 2

Programa 3

Programa 4

Programa 5

Versión

Programa 1

Programa 2

Programa 3

Programa 4

Programa 5

Plataforma

Programa 1

Programa 2

Programa 3

Programa 4

Programa 5

Idioma

Programa 1

Programa 2

Programa 3

Programa 4

Programa 5

Licencia

Programa 1

Programa 2

Programa 3

Programa 4

Programa 5

Añadir 5

Programa 1

Programa 2

Programa 3

Programa 4

Programa 5

Acciones

Programa 1

Programa 2

Programa 3

Programa 4

Programa 5

Página 15 - Prototipo en papel 2

16

Conectado como: Tamara Morales Comino

[Editar perfil](#) 8

Tareas pendientes 2

Tareas finalizadas 3

Usuarios 4

Software 5

Laboratorios 6

Asignaturas 7

Añadir software 15

Nombre

Programa 1

Programa 2

Programa 3

Programa 4

Programa 5

Versión

Programa 1

Programa 2

Programa 3

Programa 4

Programa 5

Plataforma

Programa 1

Programa 2

Programa 3

Programa 4

Programa 5

Idioma

Programa 1

Programa 2

Programa 3

Programa 4

Programa 5

Licencia

Programa 1

Programa 2

Programa 3

Programa 4

Programa 5

Básico

Programa 1

Programa 2

Programa 3

Programa 4

Programa 5

Acciones

Programa 1

Programa 2

Programa 3

Programa 4

Programa 5

¿Está seguro de eliminar el software?

Si 5

No 5

Página 16 - Prototipo en papel 2

17

Conectado como: Tamara Morales Comino
[Editar perfil 8](#)

Tareas pendientes	2
Tareas finalizadas	3
Usuarios	4
Software	5
Laboratorios	6
Asignaturas	7

Añadir laboratorio 17

Nombre	Acciones
Laboratorio	 18
Laboratorio	 18
Laboratorio	 18

Añadir laboratorio

Nombre

Dependencia

Nº de puestos

Observaciones

Añadir 6

Página 17 - Prototipo en papel 2

18

Conectado como: Tamara Morales Comino
[Editar perfil 8](#)

Tareas pendientes	2
Tareas finalizadas	3
Usuarios	4
Software	5
Laboratorios	6
Asignaturas	7

Añadir laboratorio 17

Nombre	Acciones
Laboratorio	 18
Laboratorio	 18
Laboratorio	 18

Editar laboratorio

Nombre

Dependencia

Nº de puestos

Observaciones

Eliminar 23

Guardar 6

Página 18 - Prototipo en papel 2

19
Conectado como: Tamara Morales Comino
[Editar perfil 8](#)

Tareas pendientes	2
Tareas finalizadas	3
Usuarios	4
Software	5
Laboratorios	6
Asignaturas	7

Añadir asignatura 19

Nombre	Acciones
Asignatu	20 21
Asignatu	20 21
Asignatu	20 21

Añadir asignatura 7 X

Nombre
Curso
Cuatrimestre
Laboratorio
Añadir 7

Página 19 - Prototipo en papel 2

20
Conectado como: Tamara Morales Comino
[Editar perfil 8](#)

Tareas pendientes	2
Tareas finalizadas	3
Usuarios	4
Software	5
Laboratorios	6
Asignaturas	7

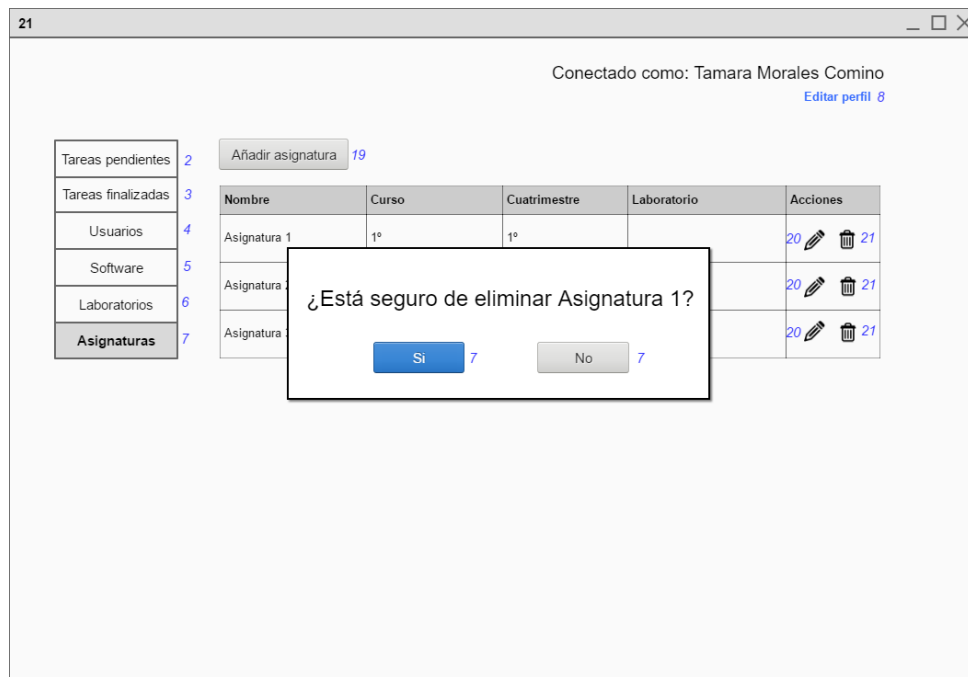
Añadir asignatura 19

Nombre	Acciones
Asignatu	20 21
Asignatu	20 21
Asignatu	20 21

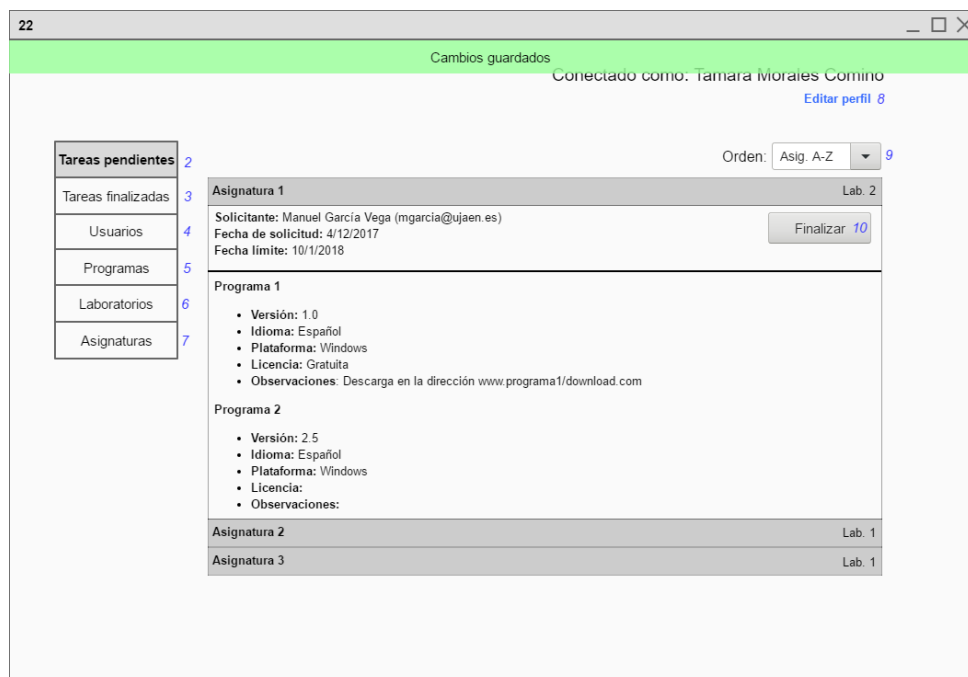
Editar asignatura 7 X

Nombre
Curso
Cuatrimestre
Laboratorio
Guardar 7

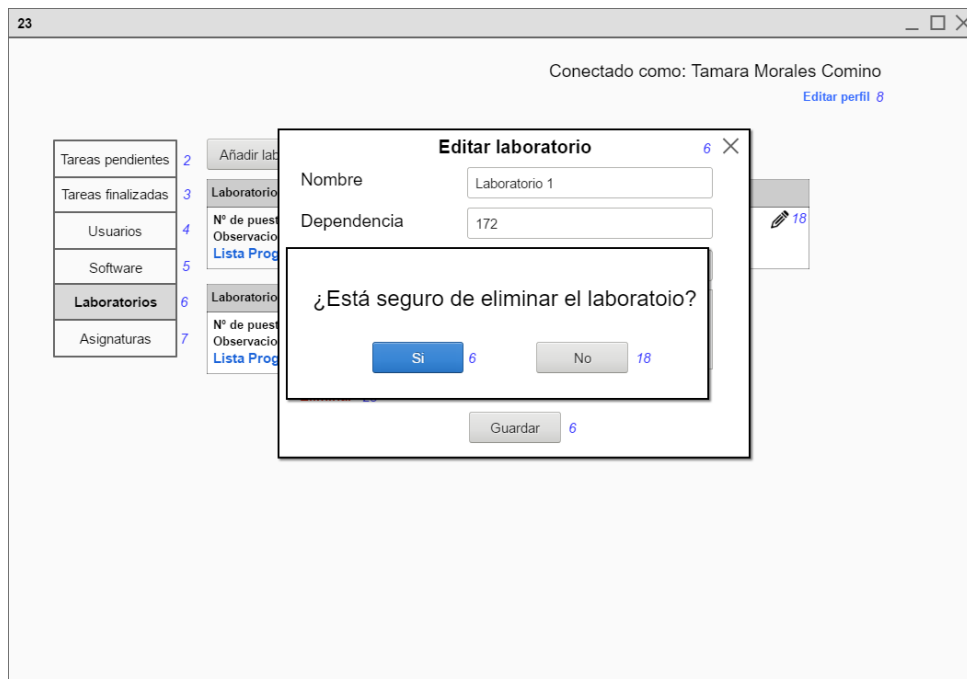
Página 20 - Prototipo en papel 2



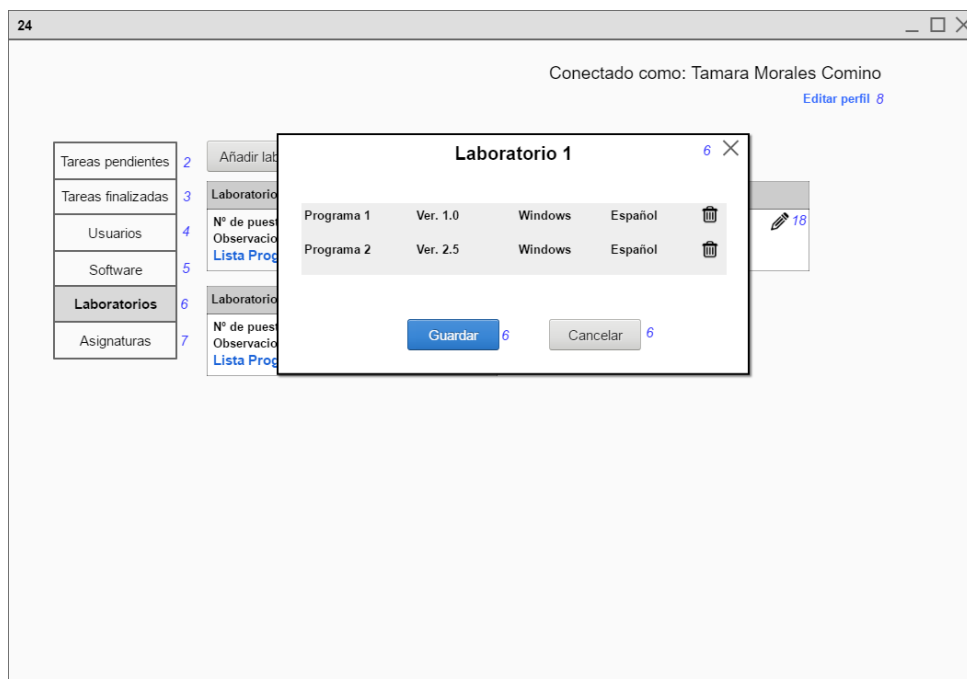
Página 21 - Prototipo en papel 2



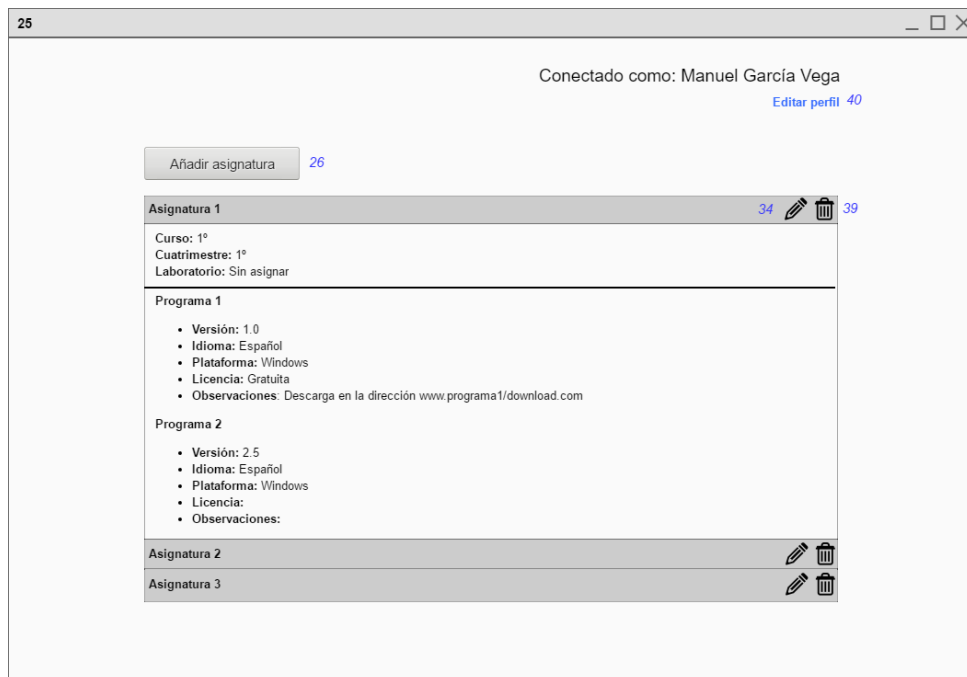
Página 22 - Prototipo en papel 2



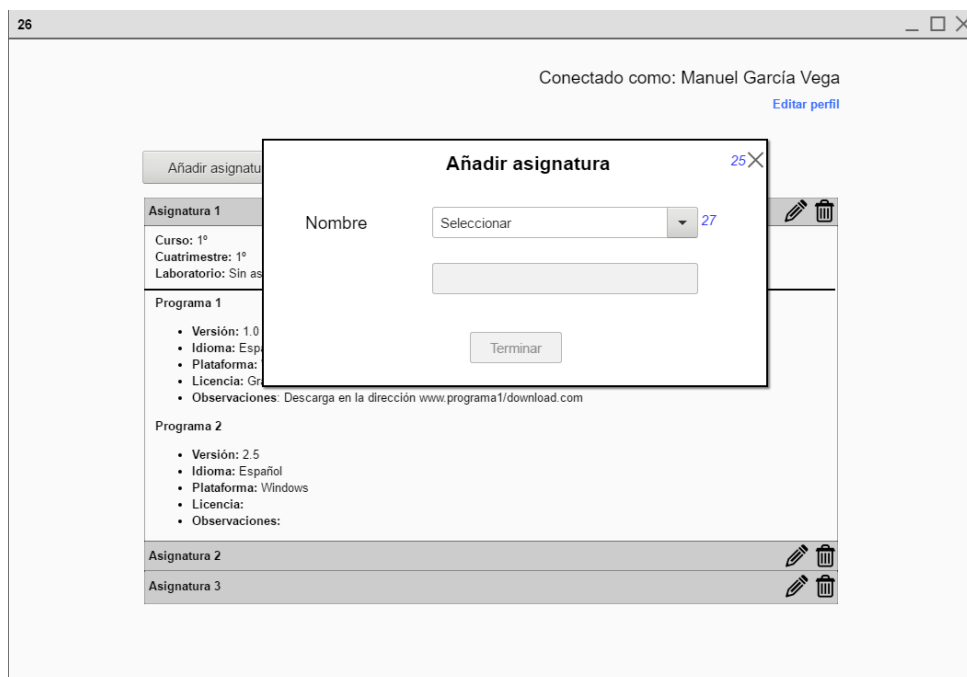
Página 23 - Prototipo en papel 2



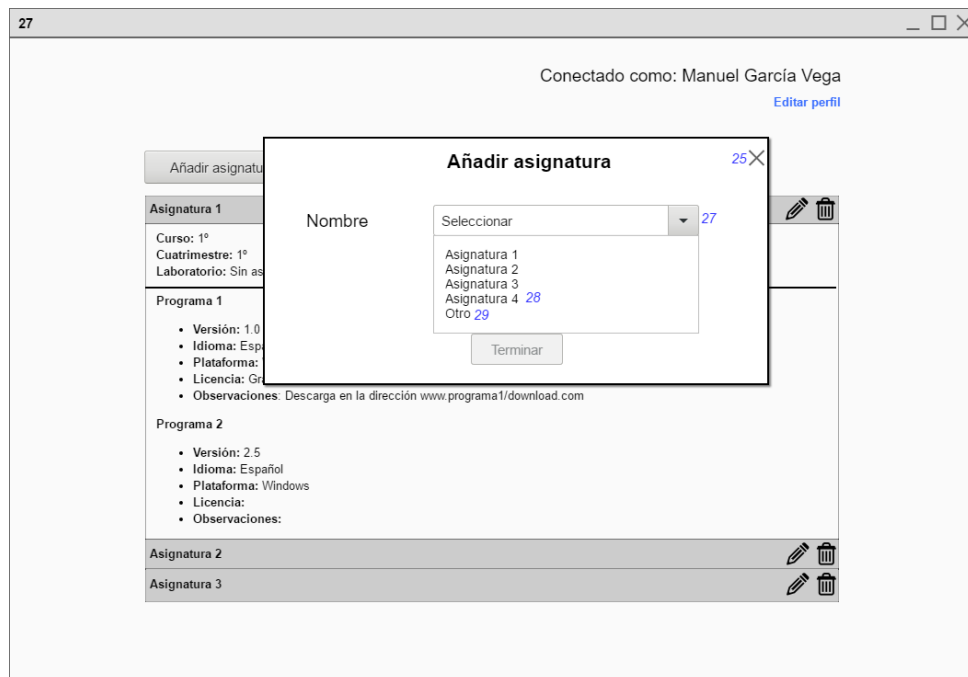
Página 24 - Prototipo en papel 2



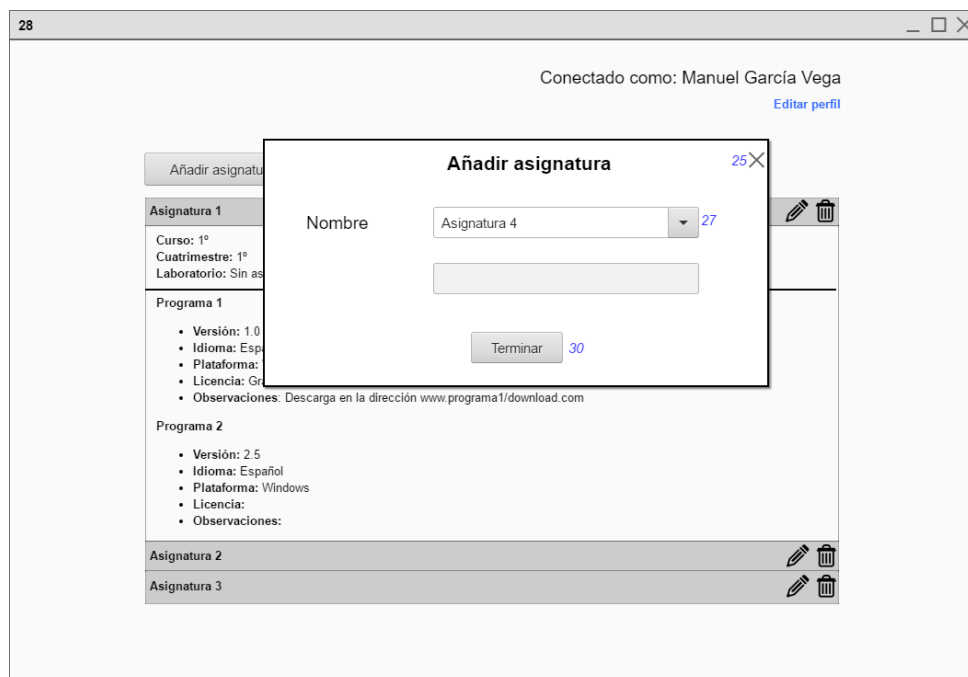
Página 25 - Prototipo en papel 2



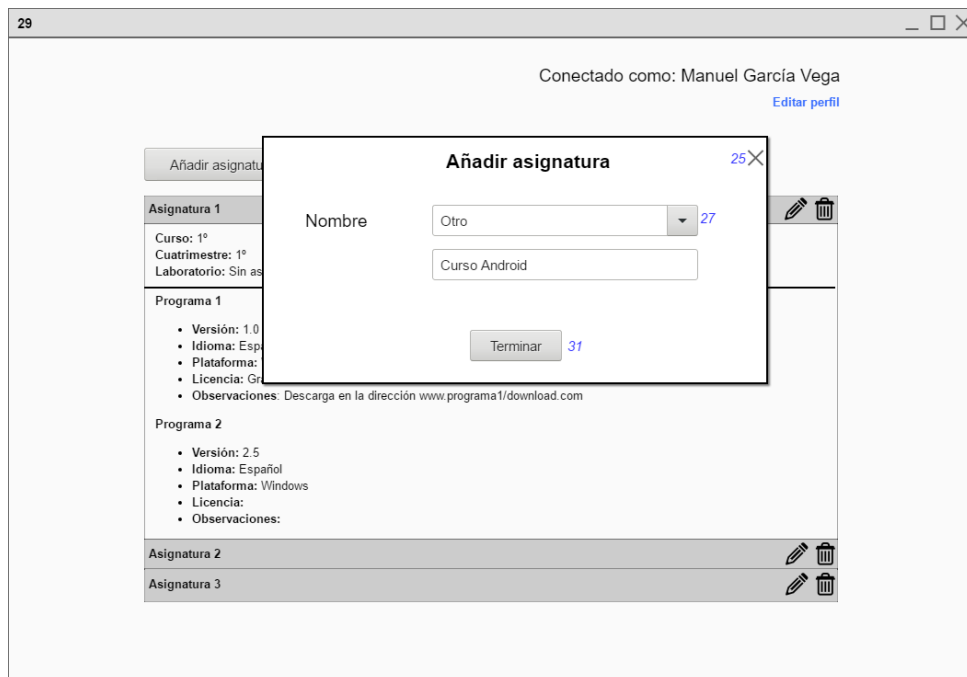
Página 26 - Prototipo en papel 2



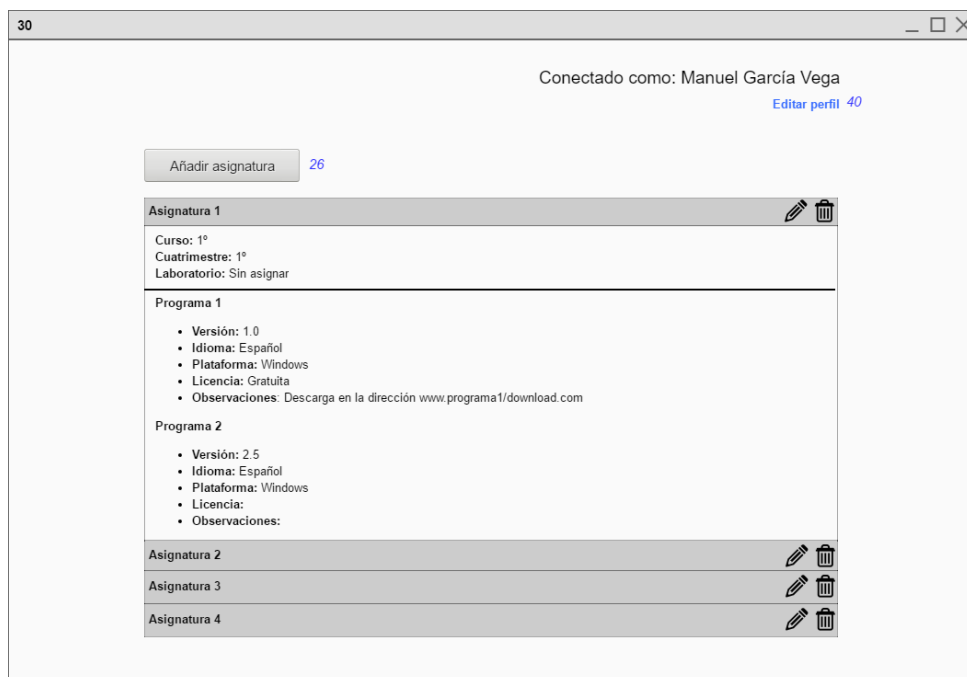
Página 27 - Prototipo en papel 2



Página 28 - Prototipo en papel 2



Página 29 - Prototipo en papel 2



Página 30 - Prototipo en papel 2

31

Conectado como: Manuel García Vega
[Editar perfil](#)

Añadir asignatura

Añadir asignatura 25 X

Nombre: Otro 27

Curso: Curso Android

Fecha: dd/mm/aaaa

Laboratorio: Seleccionar

Terminar

Asignatura 1

Curso: 1º
Cuatrimestre: 1º
Laboratorio: Sin as

Programa 1

- Versión: 1.0
- Idioma: Esp
- Plataforma:
- Licencia: Gr
- Observacion

Programa 2

- Versión: 2.5
- Idioma: Esp
- Plataforma: Windows
- Licencia:
- Observaciones:

Asignatura 2

Asignatura 3

Página 31 - Prototipo en papel 2

32

Conectado como: Manuel García Vega
[Editar perfil](#)

Añadir asignatura

Añadir asignatura 25 X

Nombre: Otro 27

Curso: Curso Android

Fecha: 18/03/2018

Laboratorio: Lab. 1

Terminar 33

Asignatura 1

Curso: 1º
Cuatrimestre: 1º
Laboratorio: Sin as

Programa 1

- Versión: 1.0
- Idioma: Esp
- Plataforma:
- Licencia: Gr
- Observacion

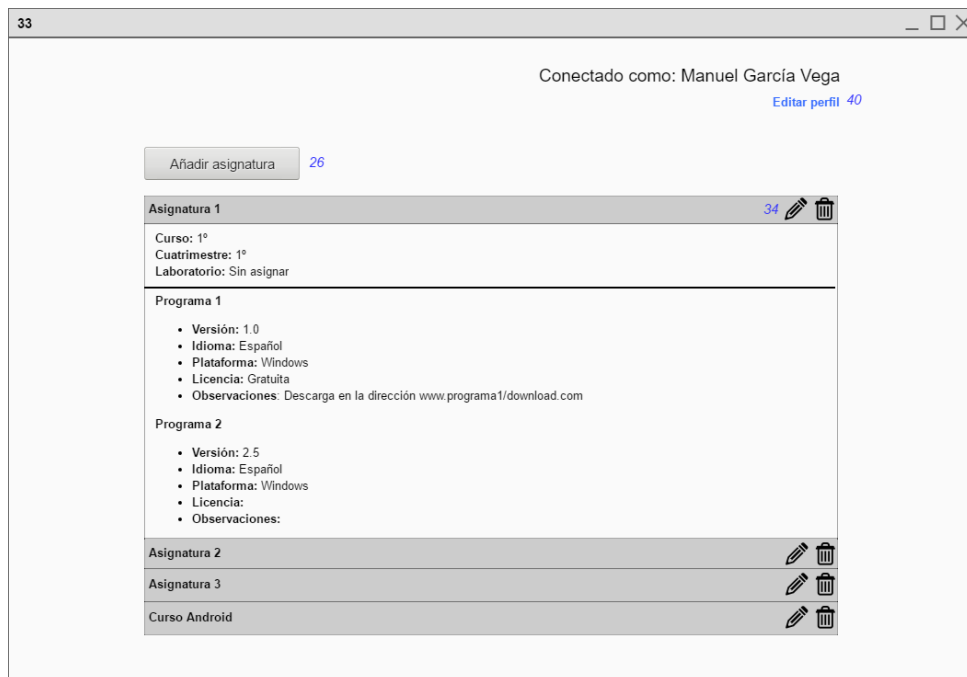
Programa 2

- Versión: 2.5
- Idioma: Esp
- Plataforma: Windows
- Licencia:
- Observaciones:

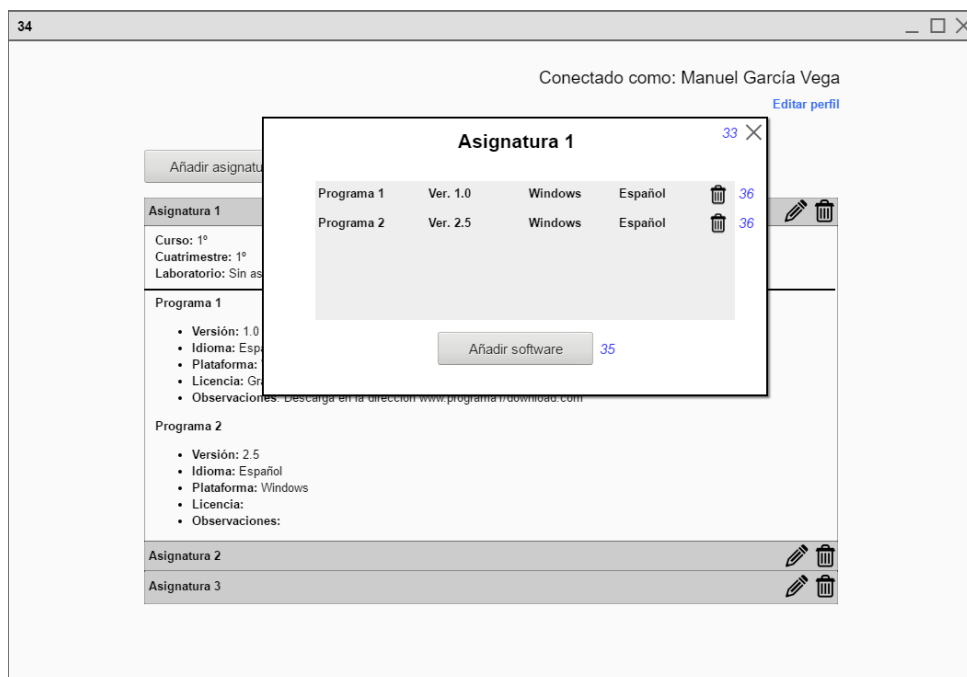
Asignatura 2

Asignatura 3

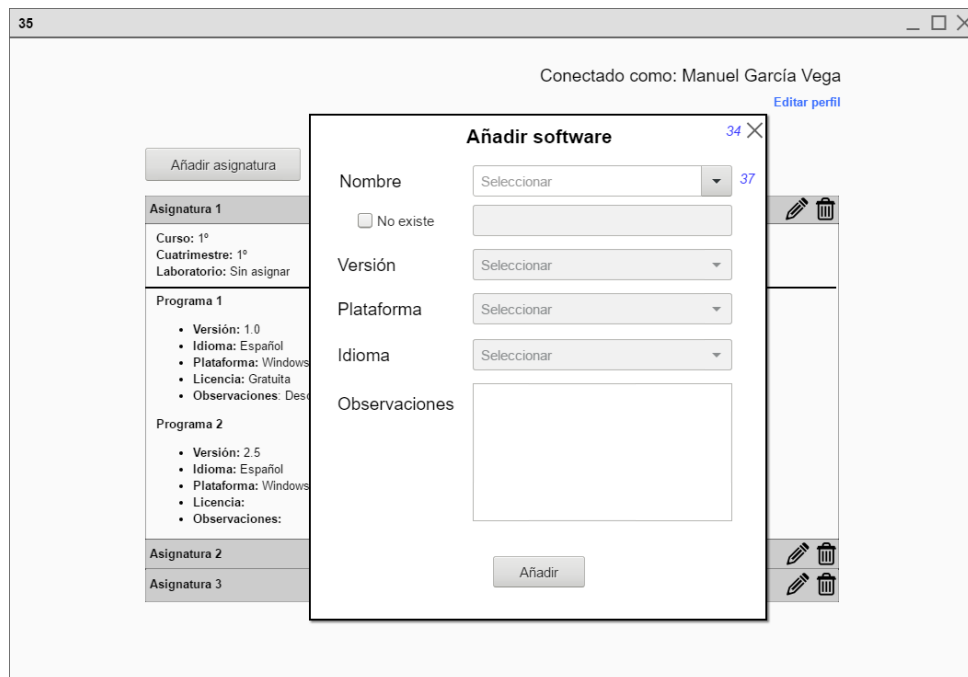
Página 32 - Prototipo en papel 2



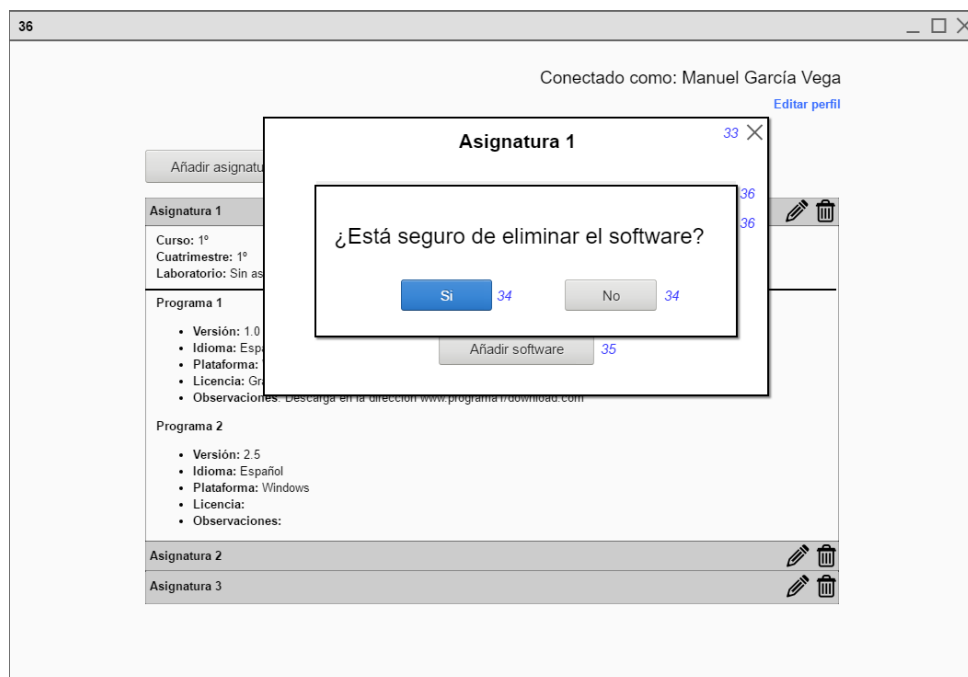
Página 33 - Prototipo en papel 2



Página 34 - Prototipo en papel 2



Página 35 - Prototipo en papel 2



Página 36 - Prototipo en papel 2

37

Conectado como: Manuel García Vega
[Editar perfil](#)

Añadir asignatura

Asignatura 1

Curso: 1º
Cuatrimestre: 1º
Laboratorio: Sin asignar

Programa 1

- Versión: 1.0
- Idioma: Español
- Plataforma: Windows
- Licencia: Gratuita
- Observaciones: Descarga en la dirección [www.programa1download.com](#)

Programa 2

- Versión: 2.5
- Idioma: Español
- Plataforma: Windows
- Licencia: Gratuita
- Observaciones:

Asignatura 2

Asignatura 3

Añadir software 34

Nombre: Programa 3

☐ No existe

Versión: 1.5

Plataforma: Windows

Idioma: Inglés

Observaciones:

Añadir 38

Página 37 - Prototipo en papel 2

38

Conectado como: Manuel García Vega
[Editar perfil](#)

Añadir asignatura

Asignatura 1

Curso: 1º
Cuatrimestre: 1º
Laboratorio: Sin asignar

Programa 1

- Versión: 1.0
- Idioma: Español
- Plataforma: Windows
- Licencia: Gratuita
- Observaciones: Descarga en la dirección [www.programa1download.com](#)

Programa 2

- Versión: 2.5
- Idioma: Español
- Plataforma: Windows
- Licencia: Gratuita
- Observaciones:

Asignatura 2

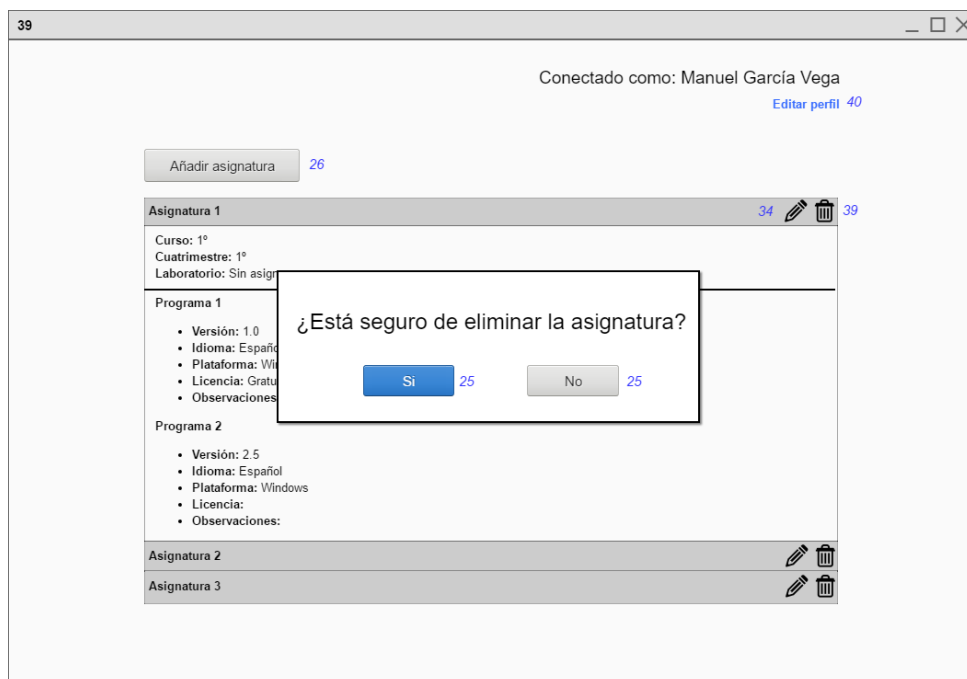
Asignatura 3

Asignatura 1 33

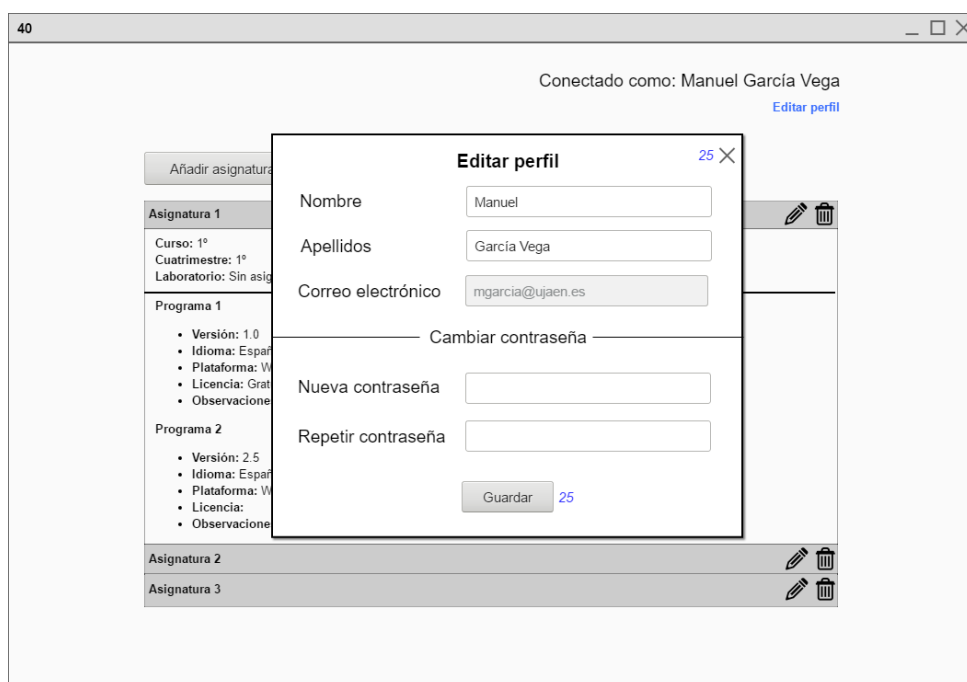
Programa 1	Ver. 1.0	Windows	Español		36
Programa 2	Ver. 2.5	Windows	Español		36
Programa 3	Ver. 1.5	Windows	Inglés		

Añadir software 35

Página 38 - Prototipo en papel 2



Página 39 - Prototipo en papel 2



Página 40 - Prototipo en papel 2

Tras la evaluación de este segundo prototipo en papel aún se encontraron diferentes fallos en el diseño y detalles que faltaban por especificar, los cuales se mencionan a continuación:

1. En las vistas de administrador:

- En el menú de tareas pendientes y finalizadas faltaría añadir un menú de filtros para poder filtrar las tareas más específicamente y facilitar su uso y búsqueda. Además, los usuarios no encontraron útil la información que mostraban las tareas cuando se encontraban sin desplegar (sólo se muestra el nombre de la asignatura y el laboratorio) por lo que se debe de modificar para mostrar más información, en este caso, laboratorio, solicitante, fecha de solicitud y fecha límite para realizar la tarea.
- En el apartado usuarios faltaría un campo de búsqueda.
- En el apartado software se añadirá un método para poder editar el software y se cambiará la opción de eliminar por ocultar. También se debe de añadir un método para validar el software que los profesores pueden solicitar como no existente y de este modo mantener la base de datos consistente.
- En el apartado de laboratorios se debe añadir la opción para eliminar todos los programas que estén instalados en un laboratorio.
- En el apartado asignaturas se debe de añadir un mecanismo para asignar los laboratorios a las distintas asignaturas más fácilmente.

2. En las vistas del profesor:

- Para añadir una “Petición puntual” se realiza mediante el mismo método de añadir una asignatura, pero debe de estar en una opción a parte. Para diferenciar entre “Asignaturas” y “Peticiones puntuales”.
- El atributo “Nombre” de una “Petición puntual” no es el indicado por lo que se debe de cambiar a “Descripción”.
- Se debe añadir una nueva parte que muestre los distintos laboratorios, así como sus datos para que los profesores puedan consultarlo cuando deseen.

2.5.4. Prototipo en papel 3

Una vez analizados los dos prototipos en papel anteriores ya conocemos los requisitos necesarios para la aplicación web por lo que ya podemos realizar un prototipo y una evaluación de dicho prototipo más exhaustivos.

1

Correo electrónico

usuario@ujaen.es

Contraseña

Entrar

Si usuario = administrador ir a 2
Si usuario = profesor ir a 28

Página 1 - Prototipo en papel 3

2

Si no hay tareas por validar ir a 6
Si hay tareas por validar ir a 6.1

Tamara Morales Comino 8

2 3 4 5 7

Tareas Pendientes Tareas Finalizadas Laboratorios Asignaturas Software Usuarios

Lab. 2	Manuel García Vega	4/12/2017	10/01/2018
--------	--------------------	-----------	------------

Asignatura: Asignatura 1
Cuatrimestre: 1º Finalizar 10

Programa 1

- Versión: 1.0
- Idioma: Español
- Plataforma: Windows
- Licencia: Gratuita
- Observaciones: Descarga en la dirección www.programa1/download.com

Programa 2

- Versión: 2.5
- Idioma: Español
- Plataforma: Windows
- Licencia:
- Observaciones:

Lab. 2	Manuel García Vega	6/12/2017	10/01/2018
Lab. 1	Manuel García Vega	8/12/2017	10/01/2018

Filtros

Seleccionar Cuatrimestre

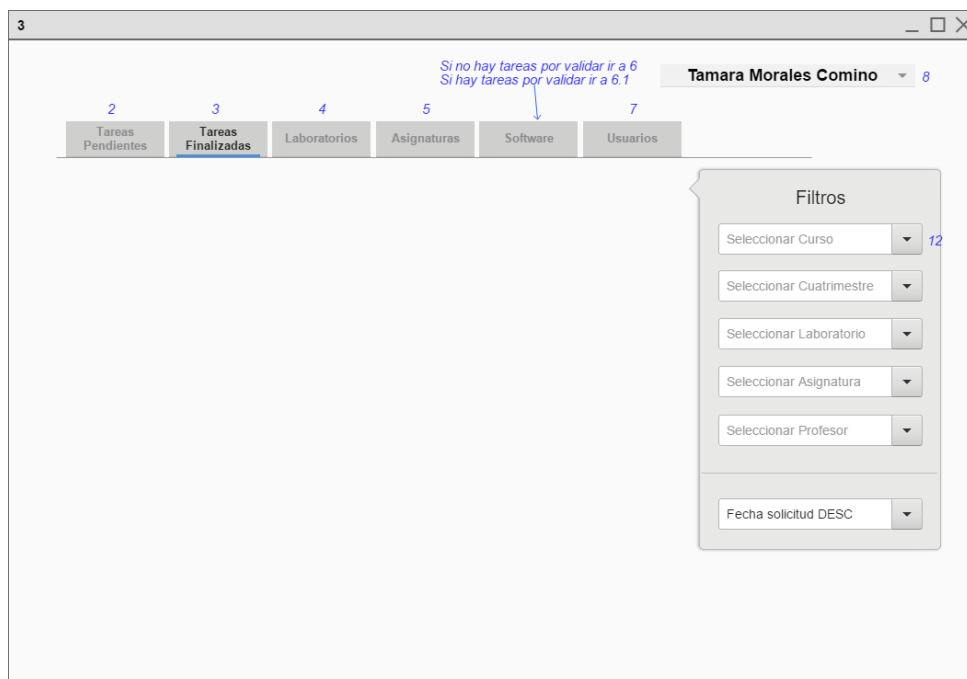
Seleccionar Laboratorio

Seleccionar Asignatura

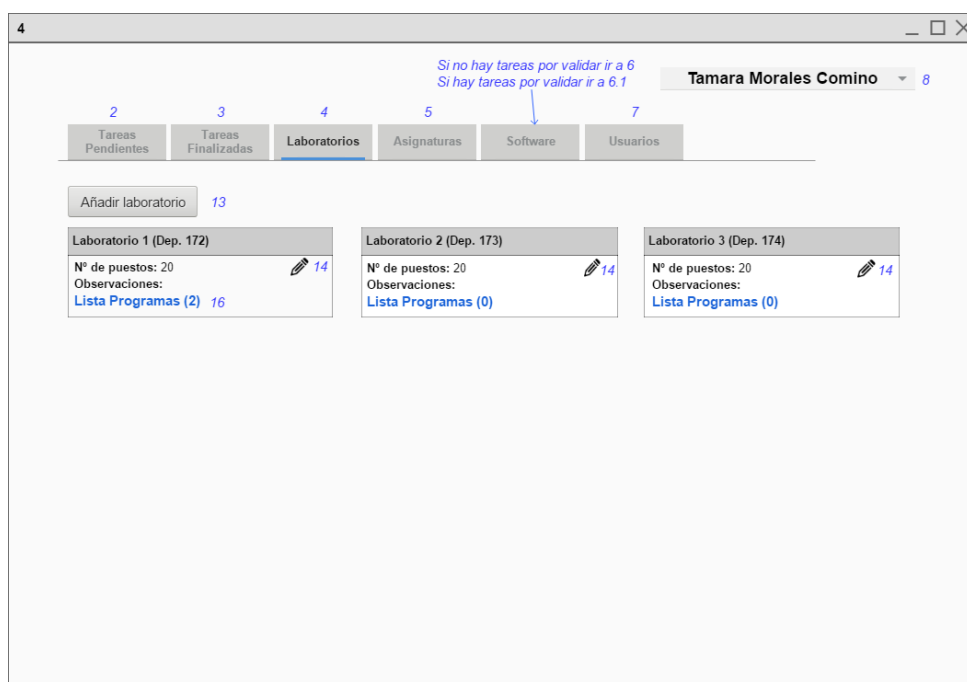
Manuel García Vega

Fecha solicitud Desc. 9

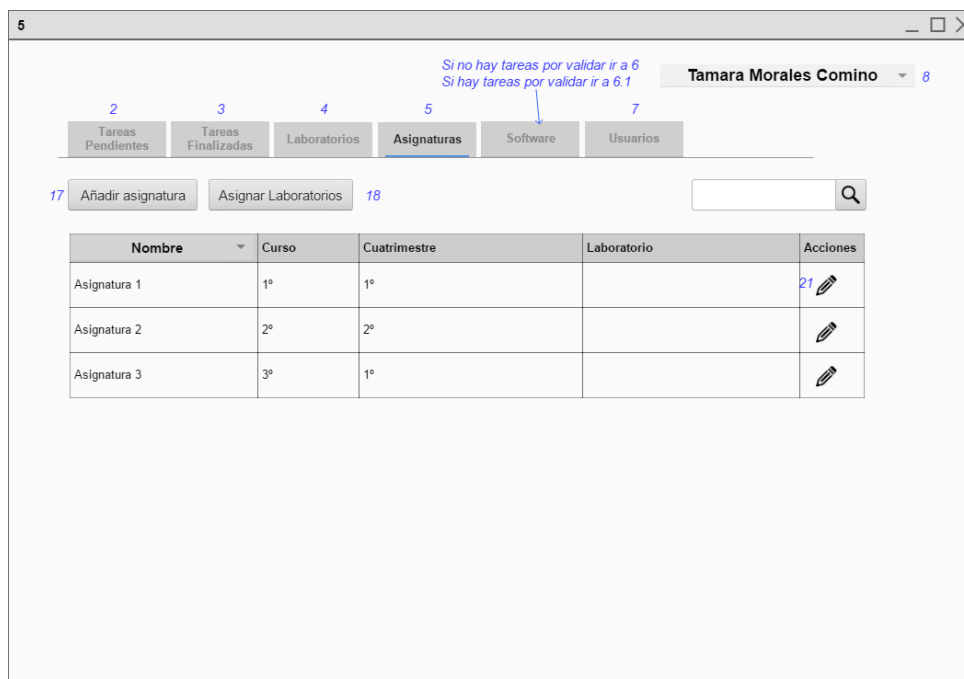
Página 2 - Prototipo en papel 3



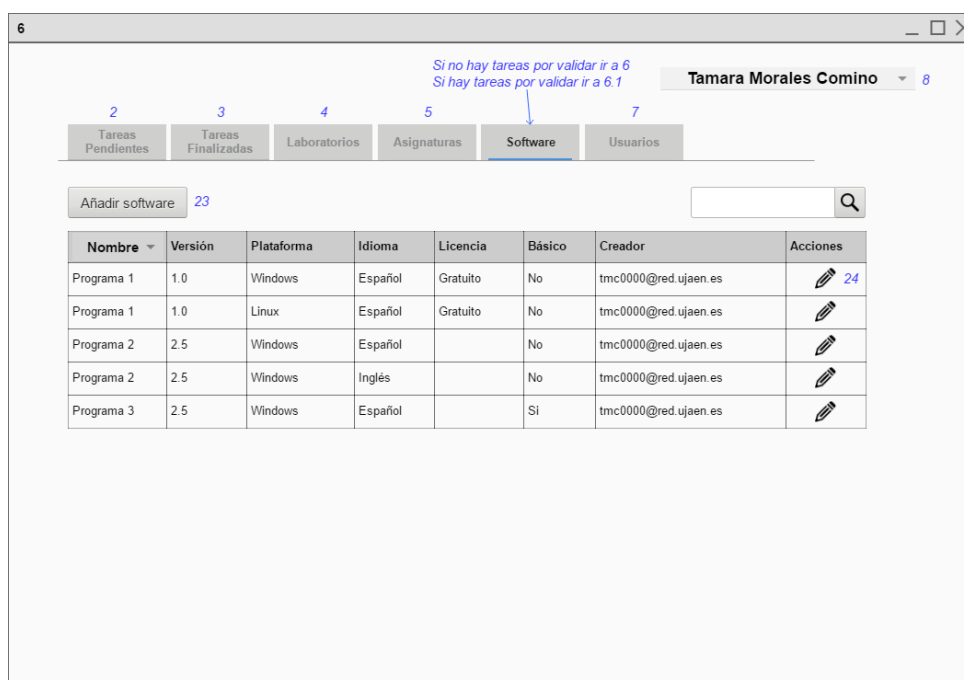
Página 3 - Prototipo en papel 3



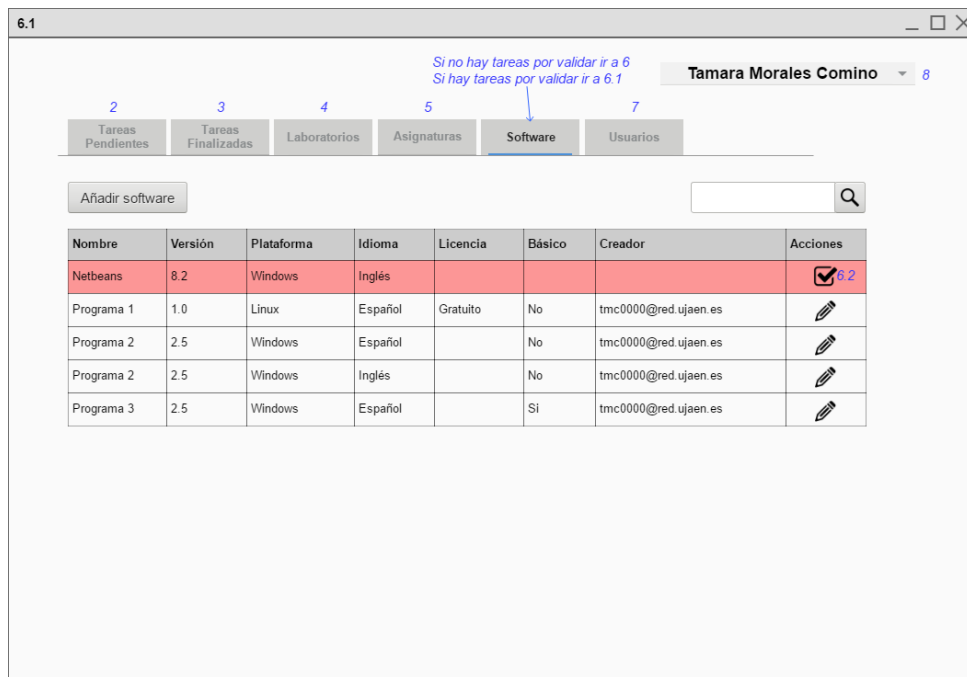
Página 4 - Prototipo en papel 3



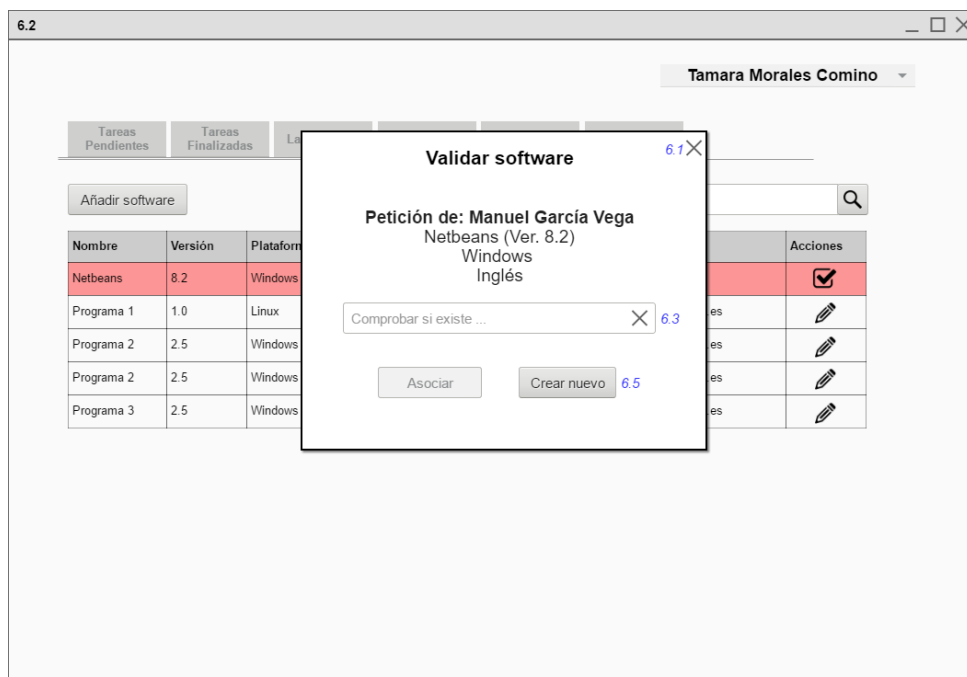
Página 5 - Prototipo en papel 3



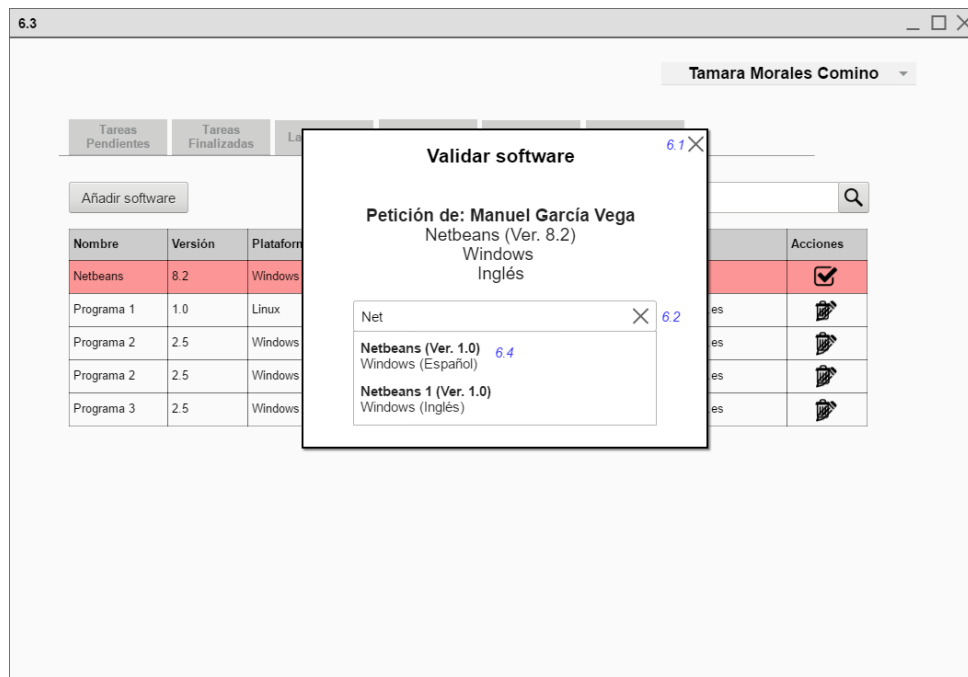
Página 6 - Prototipo en papel 3



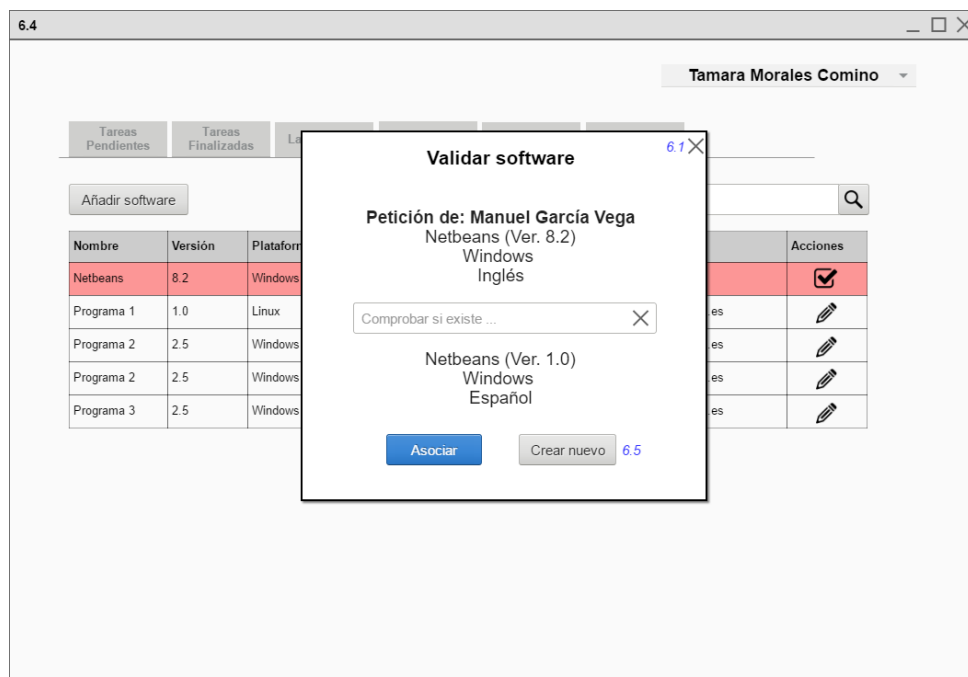
Página 6.1 - Prototipo en papel 3



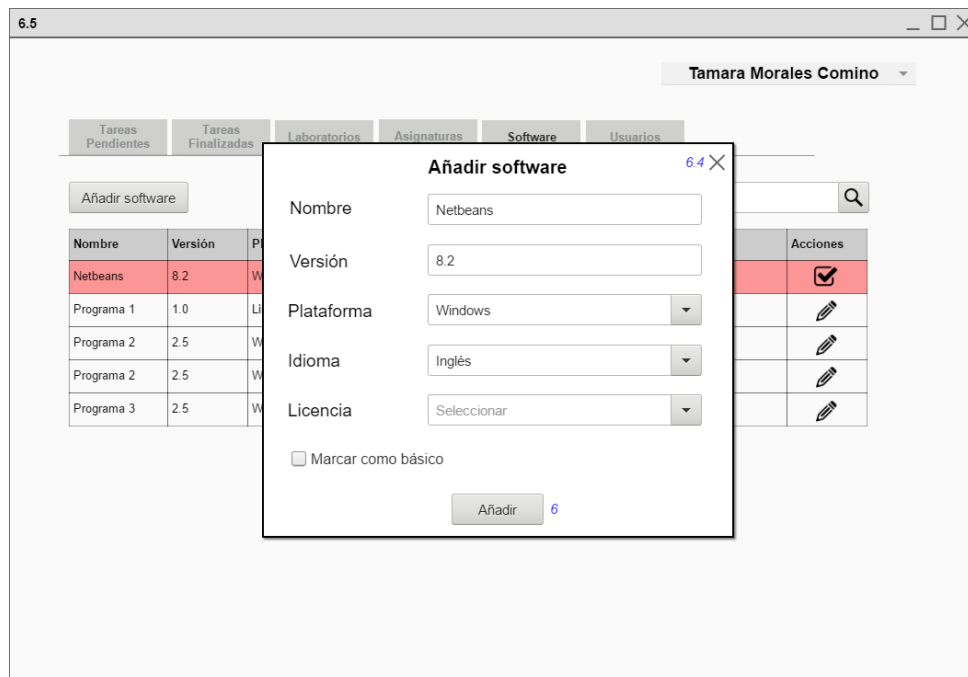
Página 6.2 - Prototipo en papel 3



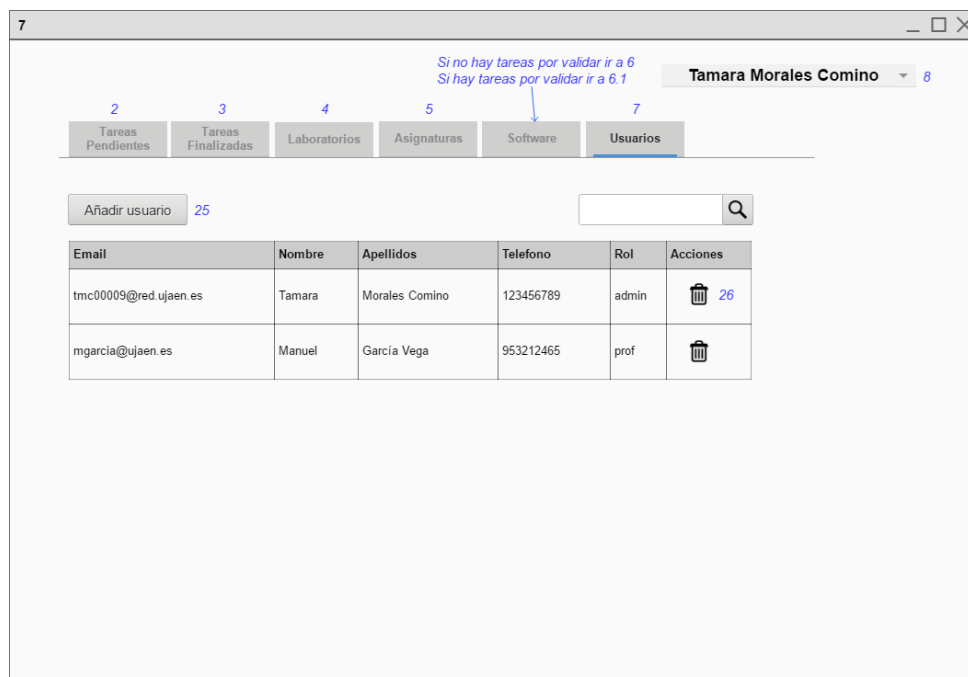
Página 6.3 - Prototipo en papel 3



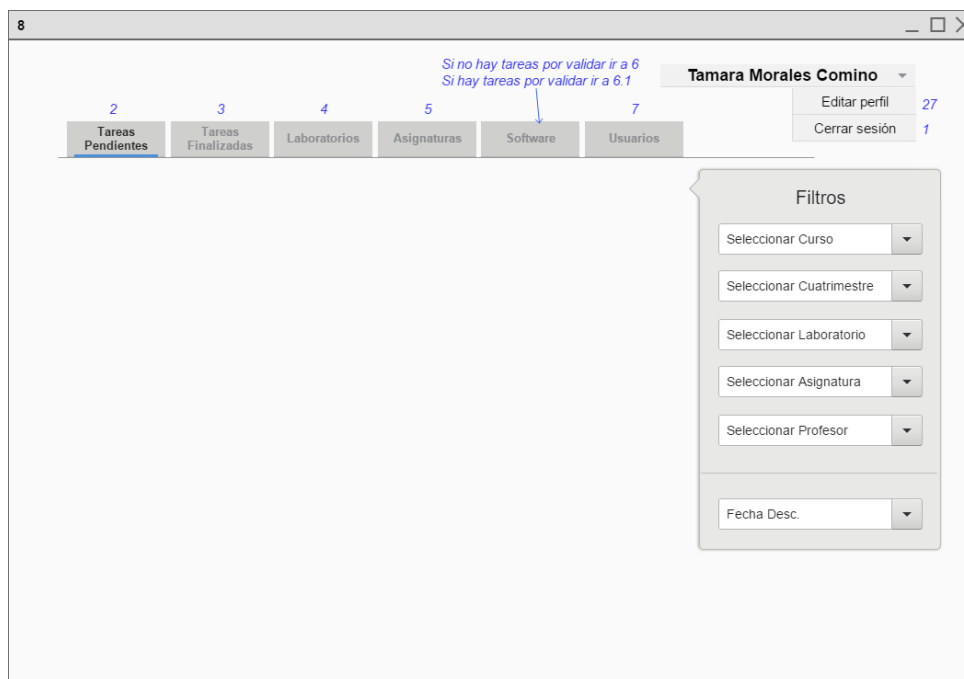
Página 6.4 - Prototipo en papel 3



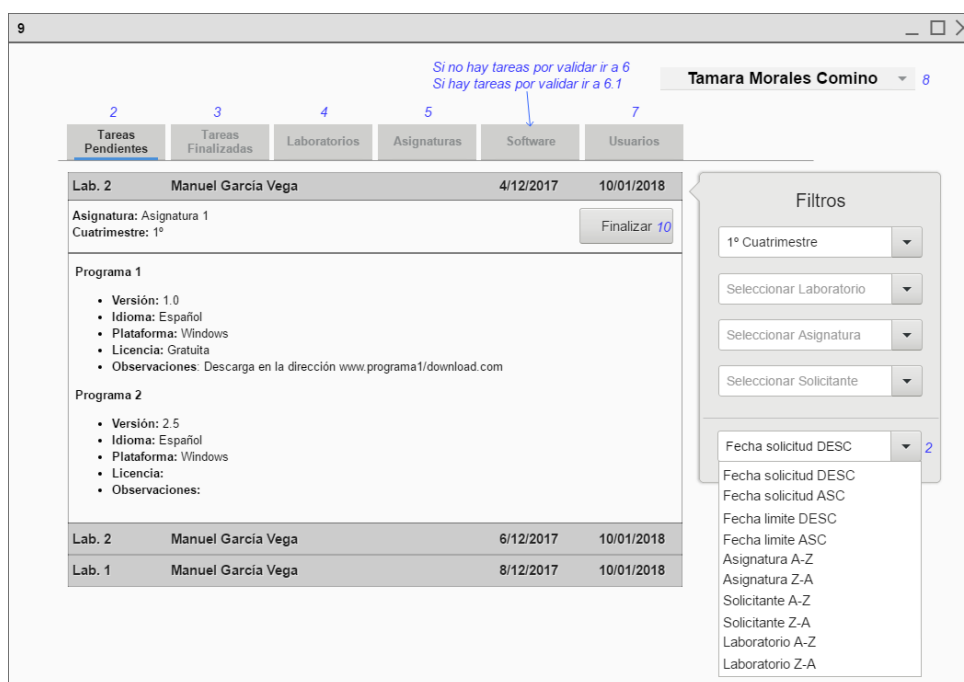
Página 6.5 - Prototipo en papel 3



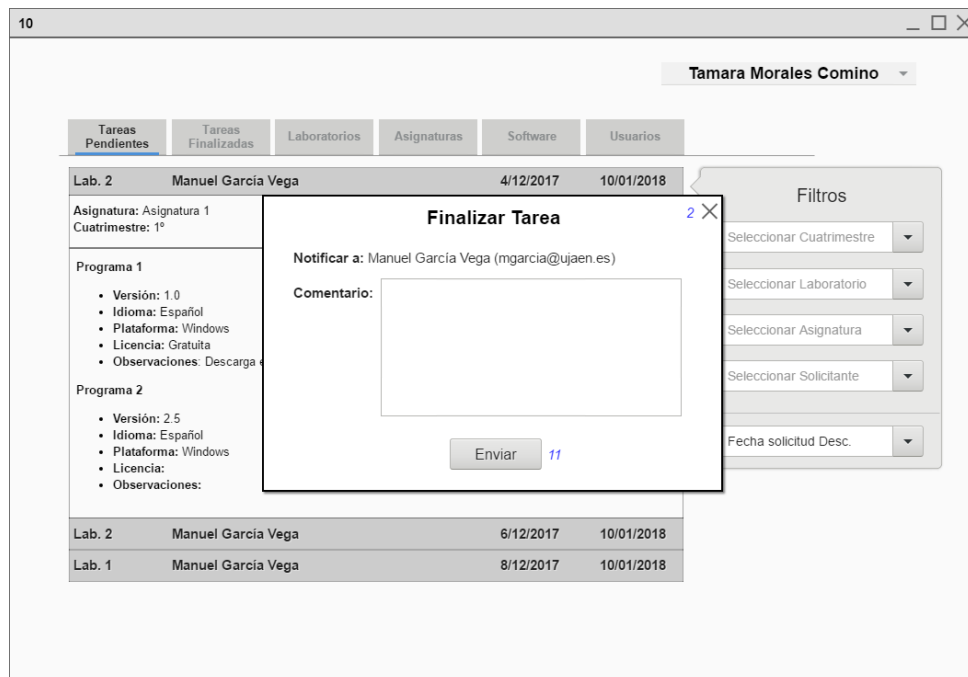
Página 7 - Prototipo en papel 3



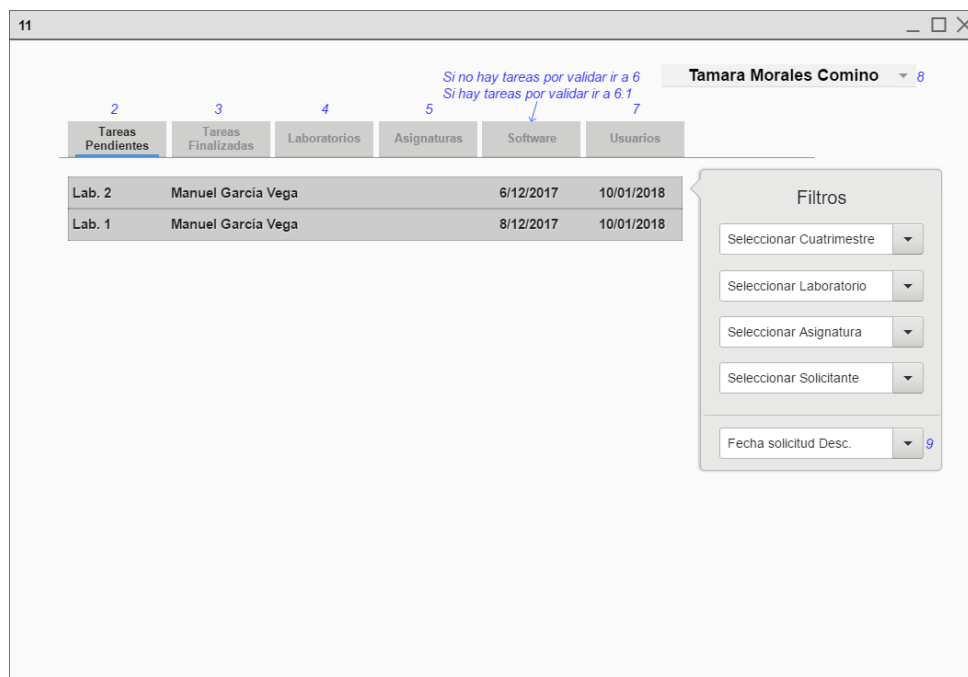
Página 8 - Prototipo en papel 3



Página 9 - Prototipo en papel 3



Página 10 - Prototipo en papel 3



Página 11 - Prototipo en papel 3

12

Tamara Morales Comino 8

2 Tareas Pendientes 3 Tareas Finalizadas 4 Laboratorios 5 Asignaturas 6 Software 7 Usuarios

Si no hay tareas por validar ir a 6
Si hay tareas por validar ir a 6.1

Lab. 2	Manuel Garcia Vega	4/12/2017	10/01/2018
Asignatura: Asignatura 4 Cuatrimestre: 2º Finalizado por: Tamara Morales Comino Fecha finalización: 6/01/2018			
Programa 1 - Versión: 1.0 - Idioma: Español - Plataforma: Windows - Licencia: Gratuita - Observaciones: Descarga en la dirección www.programa1/download.com			
Programa 2 - Versión: 2.5 - Idioma: Español - Plataforma: Windows - Licencia: - Observaciones:			
Lab. 2	Manuel Garcia Vega	6/12/2017	10/01/2018

Filtros

2017/18

2º Cuatrimestre

Laboratorio 2

Seleccionar Asignatura

Seleccionar Profesor

Fecha solicitud DESC

Página 12 - Prototipo en papel 3

13

Tamara Morales Comino

Tareas Pendientes Tareas Finalizadas Laboratorios Asignaturas Software Usuarios

Añadir laboratorio

Laboratorio 1 (Dep. 172)

Nº de puestos: 20
Observaciones:
[Lista Programas \(2\)](#)

(Dep. 174)

Nº de puestos:
Observaciones:
[Programas \(0\)](#)

Añadir laboratorio

Nombre

Dependencia

Nº de puestos

Observaciones

Añadir

Página 13 - Prototipo en papel 3

14

Tamara Morales Comino 8

Tareas Pendientes Tareas Finalizadas Laboratorios Asignaturas Software Usuarios

Añadir laboratorio

Laboratorio 1 (Dep. 172)

Nº de puestos: 20
Observaciones:
[Lista Programas \(2\)](#)

Editar laboratorio 4 X

Nombre: Laboratorio 1

Dependencia: 172

Nº de puestos: 20

Observaciones:

Eliminar 15

Guardar 4

Página 14 - Prototipo en papel 3

15

Tamara Morales Comino 8

Tareas Pendientes Tareas Finalizadas Laboratorios Asignaturas Software Usuarios

Añadir laboratorio

Laboratorio 1 (Dep. 172)

Nº de puestos: 20
Observaciones:
[Lista Programas \(2\)](#)

Editar laboratorio X

Nombre: Laboratorio 1

Dependencia: 172

Nº de puestos: 20

Observaciones:

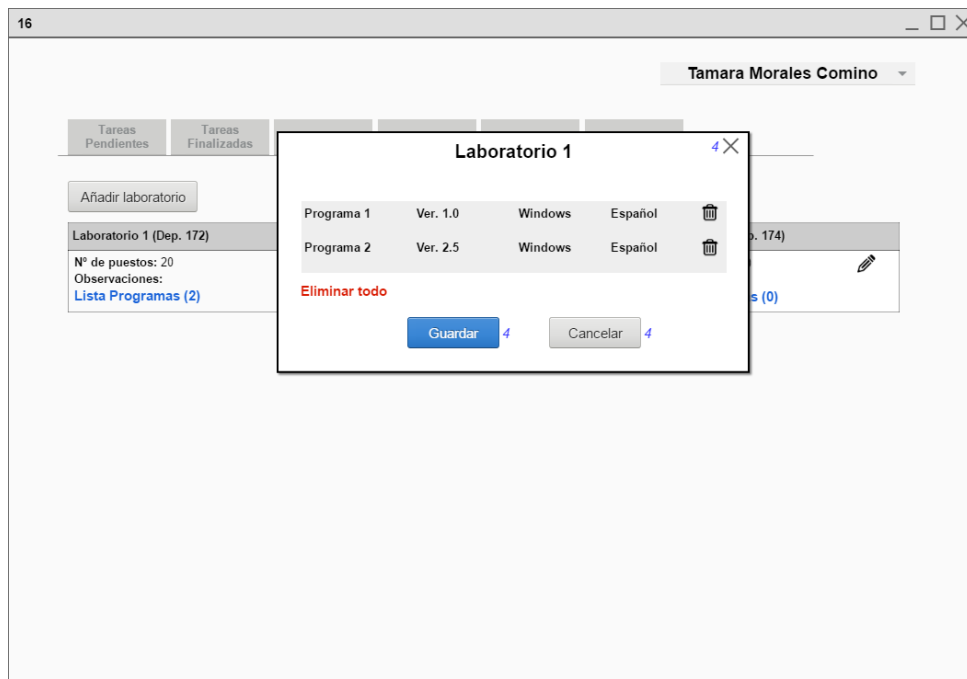
¿Está seguro de eliminar el laboratorio?

Si 4 No 14

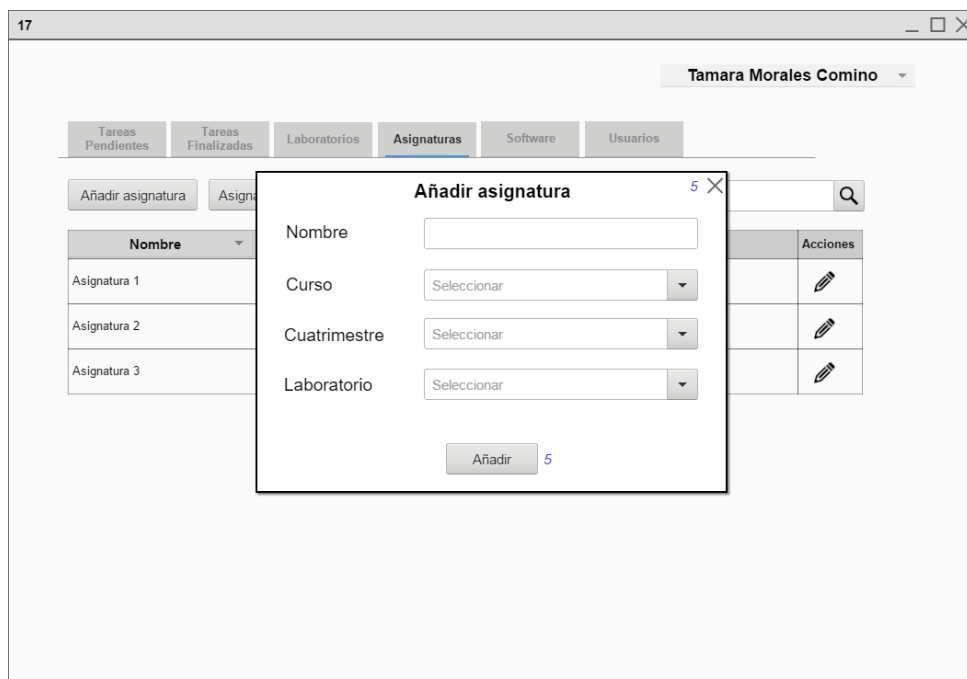
Eliminar

Guardar

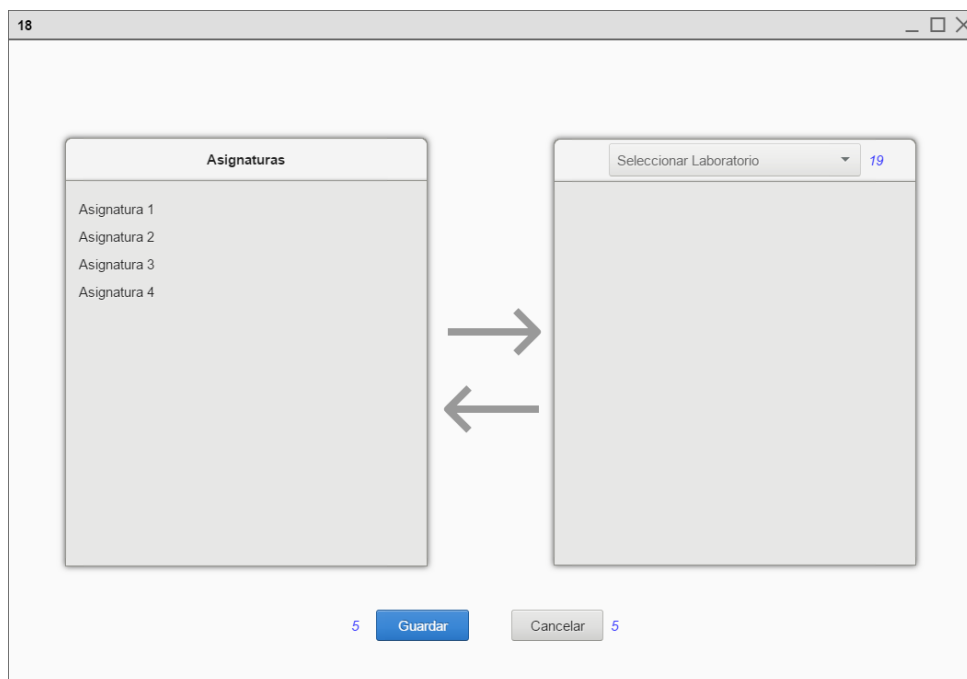
Página 15 - Prototipo en papel 3



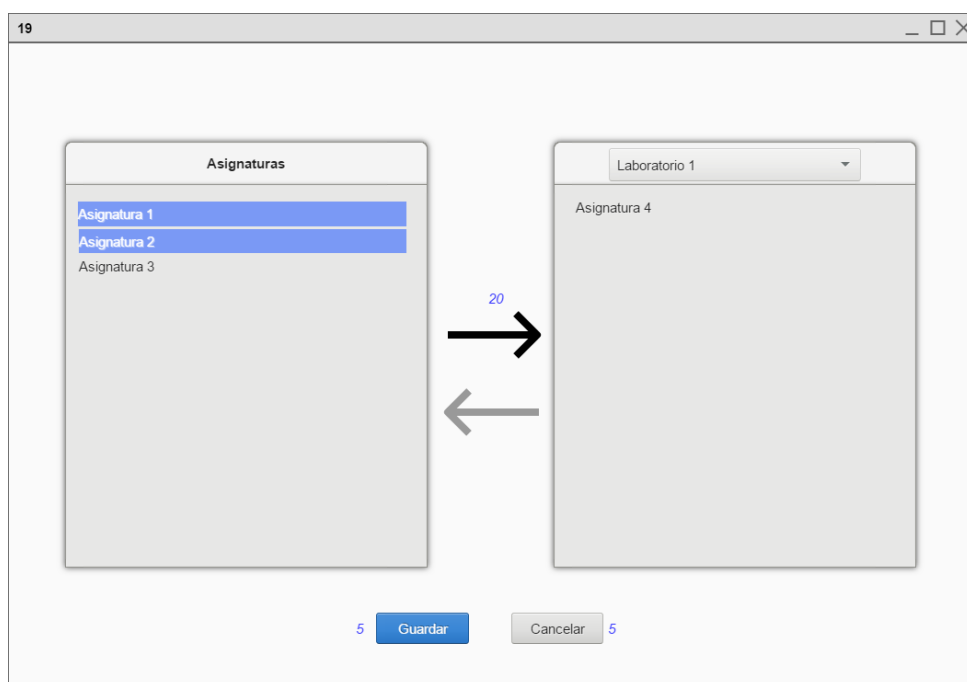
Página 16 - Prototipo en papel 3



Página 17 - Prototipo en papel 3



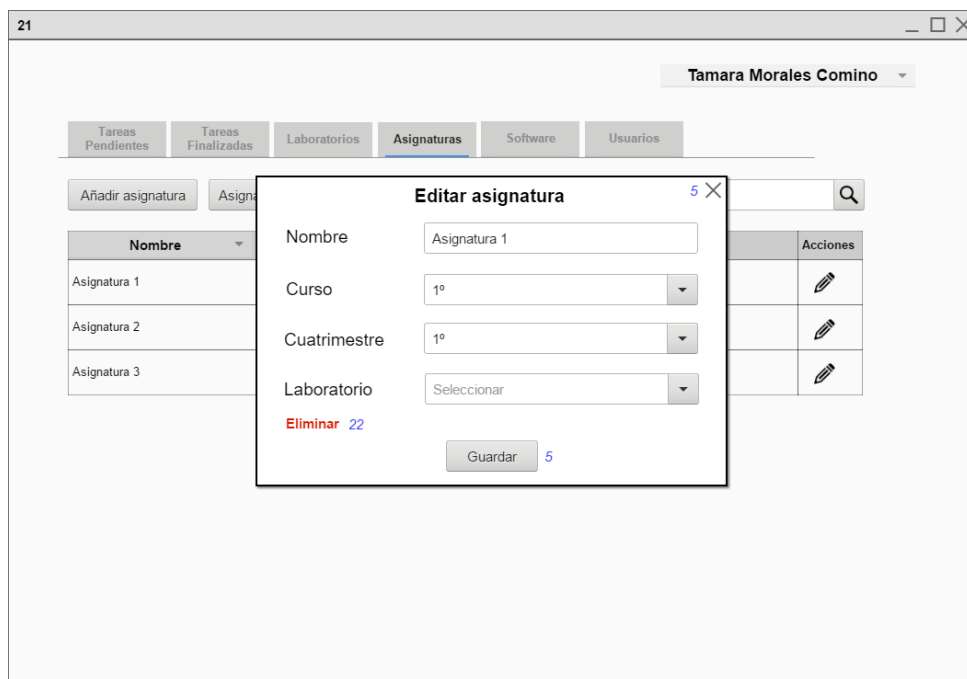
Página 18 - Prototipo en papel 3



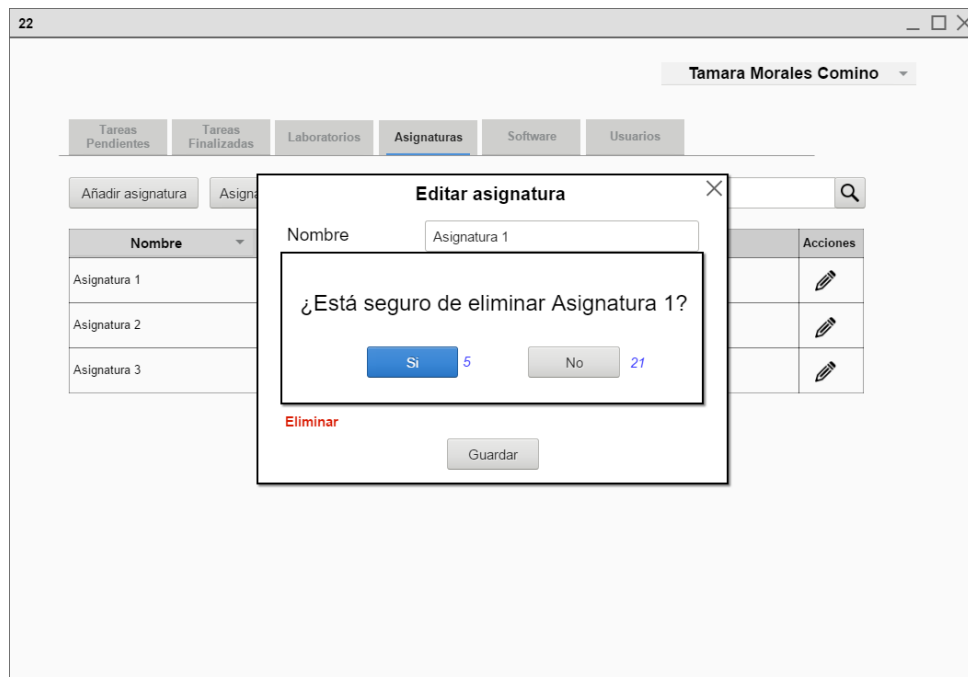
Página 19 - Prototipo en papel 3



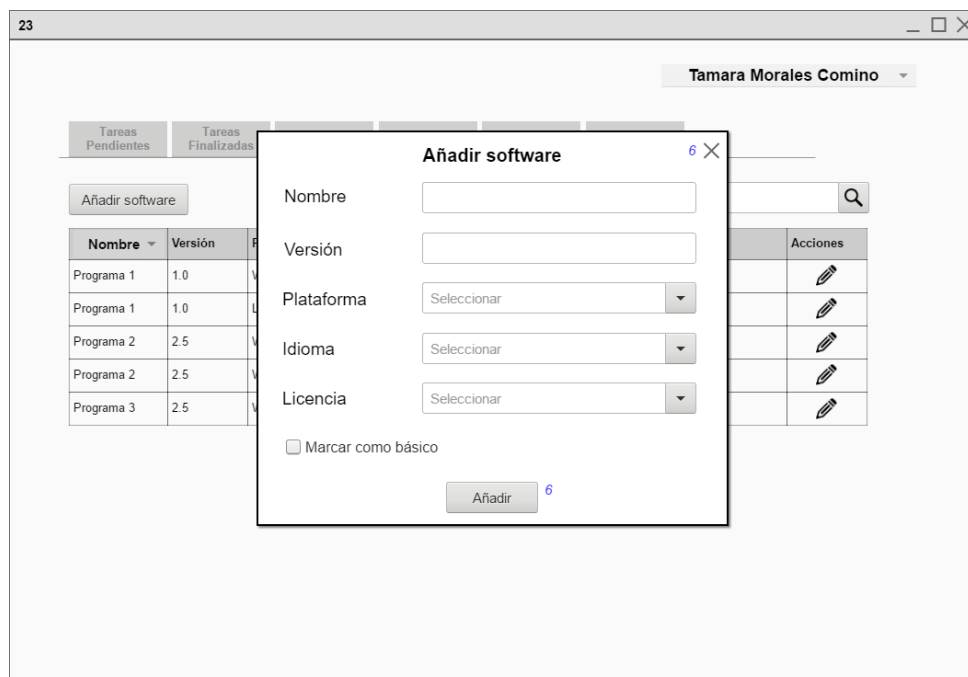
Página 20 - Prototipo en papel 3



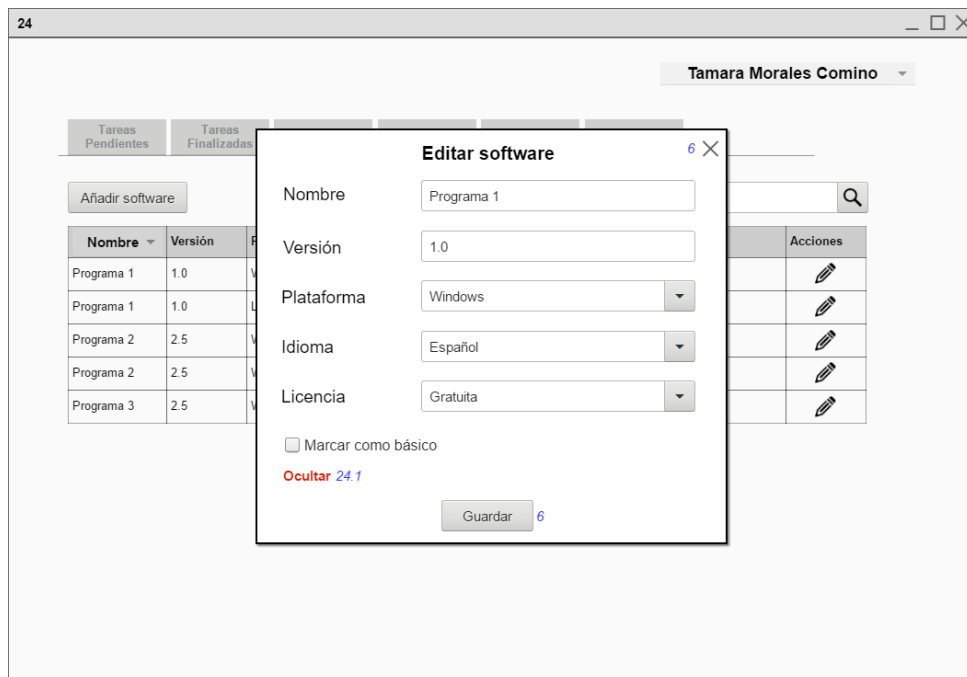
Página 21 - Prototipo en papel 3



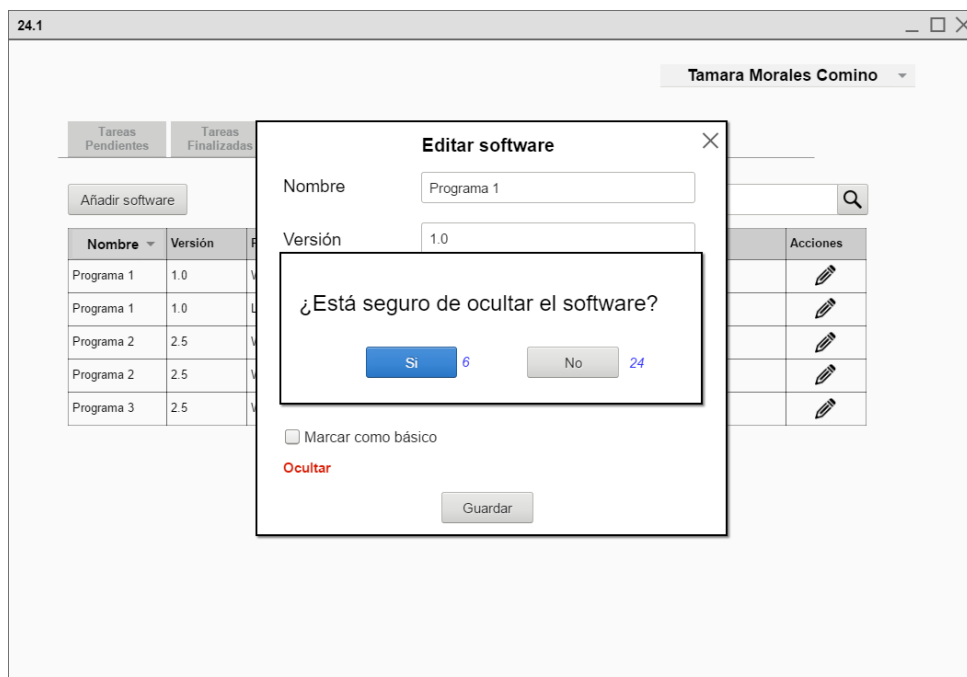
Página 22 - Prototipo en papel 3



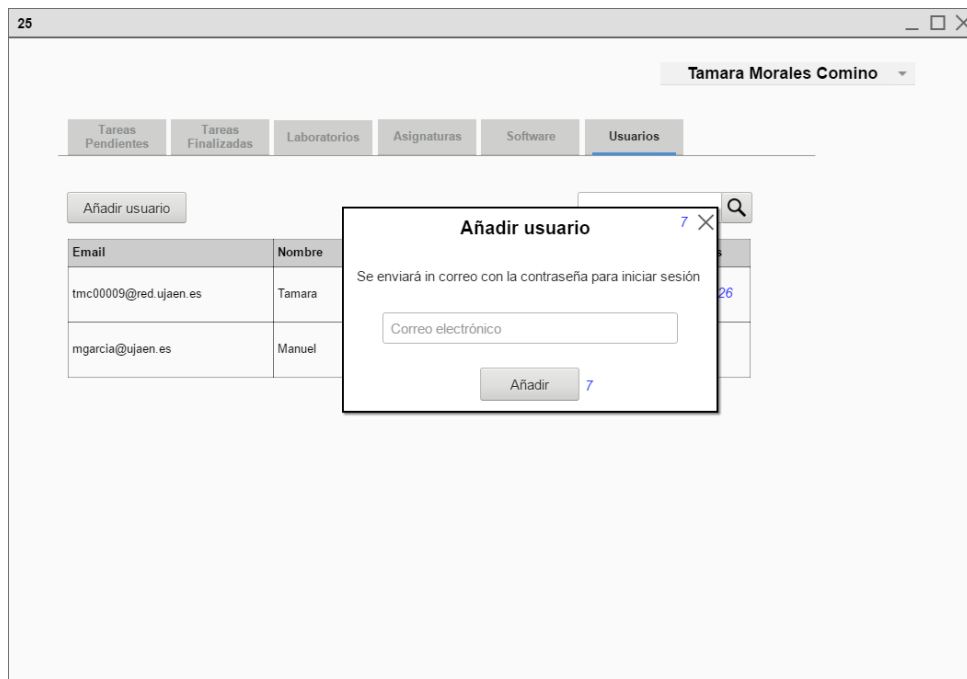
Página 23 - Prototipo en papel 3



Página 24 - Prototipo en papel 3



Página 24.1 - Prototipo en papel 3



Página 25 - Prototipo en papel 3



Página 26 - Prototipo en papel 3

27

Tamara Morales Comino

Tareas Pendientes Tareas Finalizadas Laboratorios Asignaturas Software Usuarios

Editar perfil

Nombre

Apellidos

Correo electrónico

Cambiar contraseña

Nueva contraseña

Repetir contraseña

Guardar

Filtros

Seleccionar Curso

Seleccionar Cuatrimestre

Seleccionar Laboratorio

Seleccionar Asignatura

Seleccionar Profesor

Fecha Desc.

Página 27 - Prototipo en papel 3

28

Manuel García Vega

Asignaturas Laboratorios

Añadir asignatura Petición puntual

Asig. A-Z

Asignatura 1

Curso: 1º
Cuatrimestre: 1º
Laboratorio: Sin asignar

Programa 1

- Versión: 1.0
- Idioma: Español
- Plataforma: Windows
- Licencia: Gratuita
- Observaciones: Descarga en la dirección www.programa1/download.com

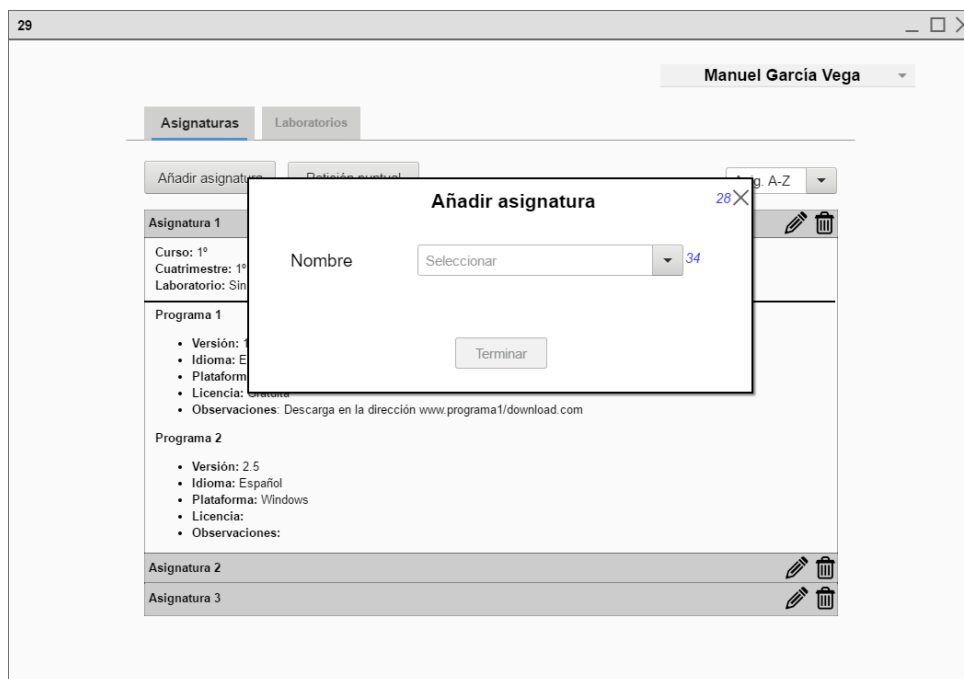
Programa 2

- Versión: 2.5
- Idioma: Español
- Plataforma: Windows
- Licencia:
- Observaciones:

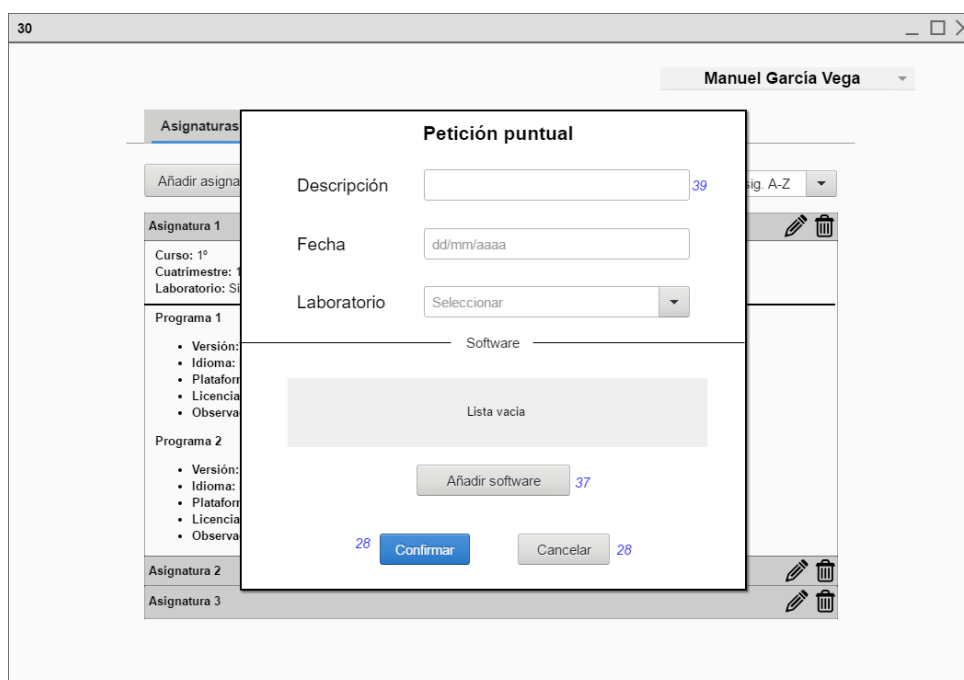
Asignatura 2

Asignatura 3

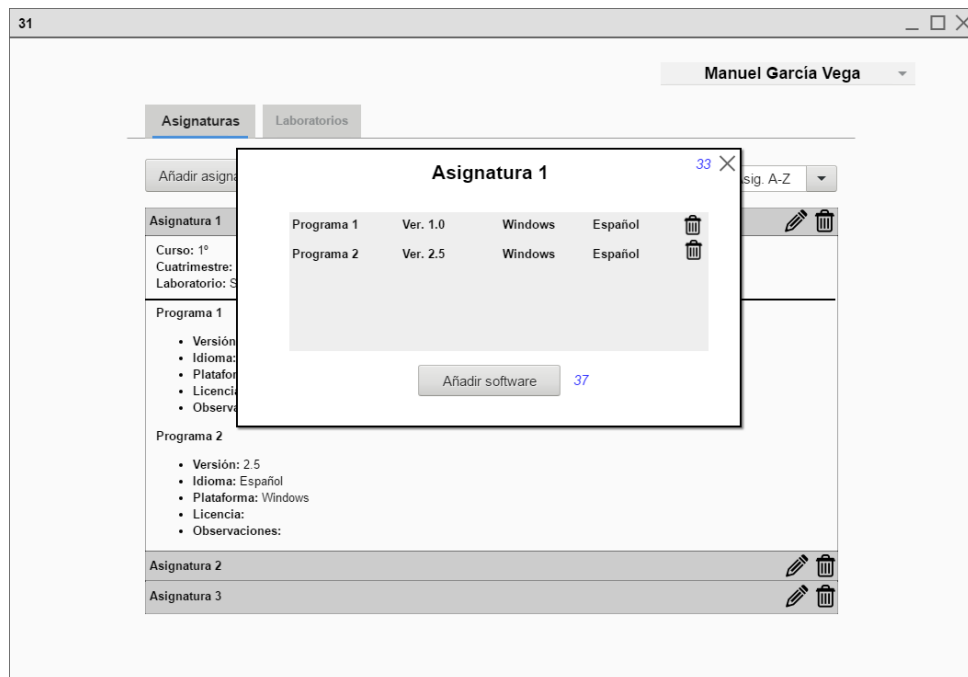
Página 28 - Prototipo en papel 3



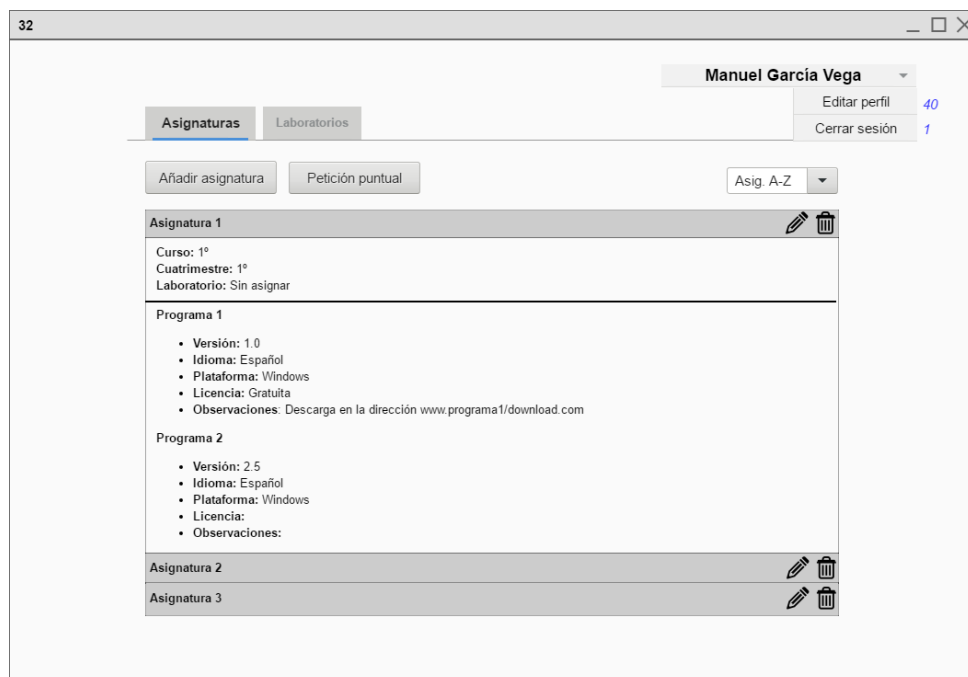
Página 29 - Prototipo en papel 3



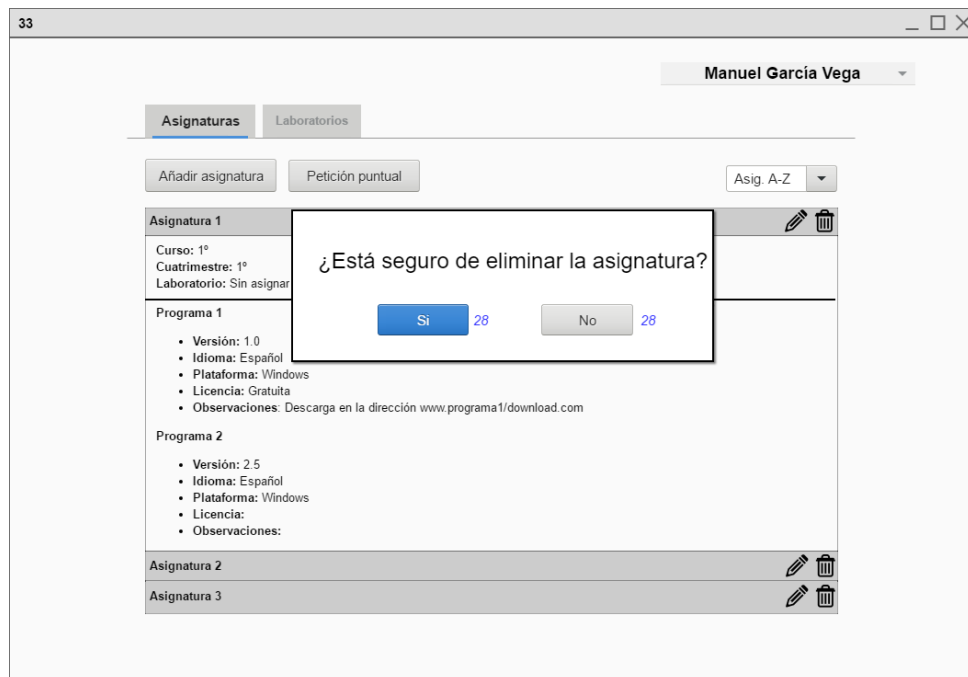
Página 30 - Prototipo en papel 3



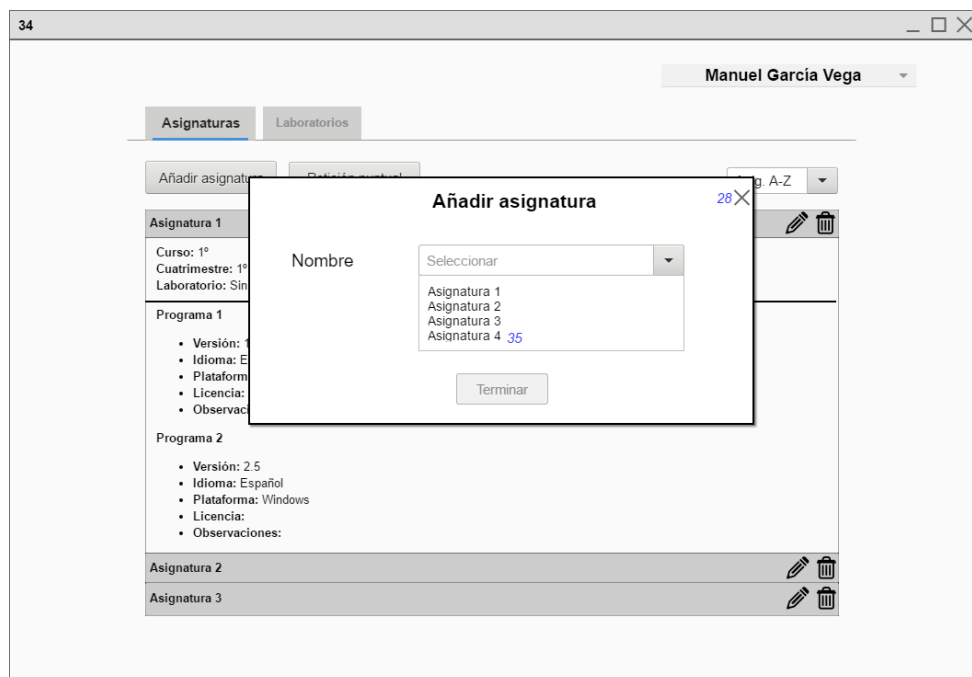
Página 31 - Prototipo en papel 3



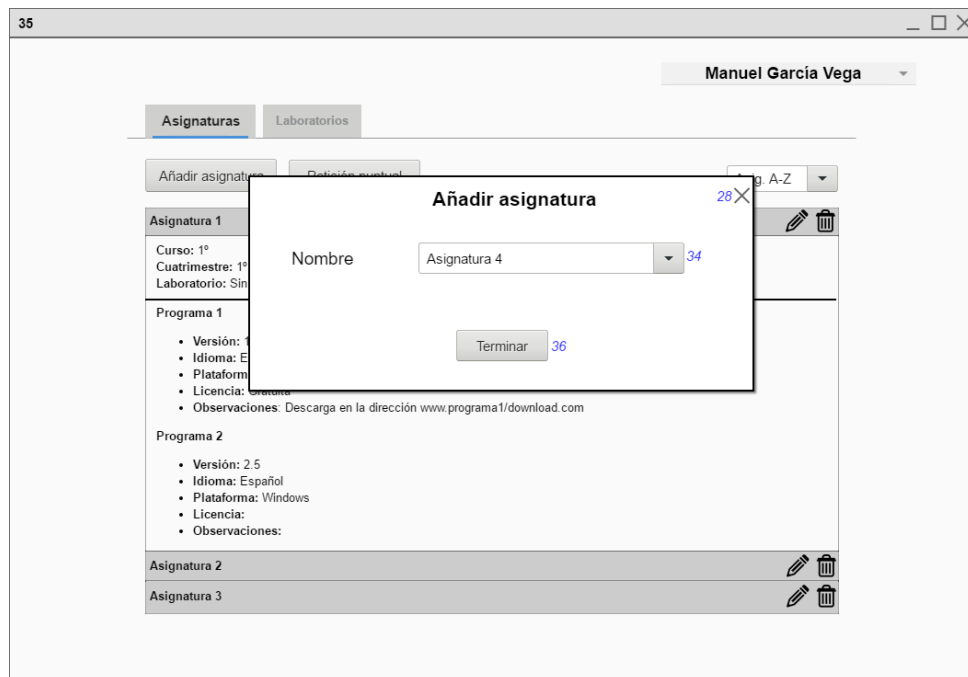
Página 32 - Prototipo en papel 3



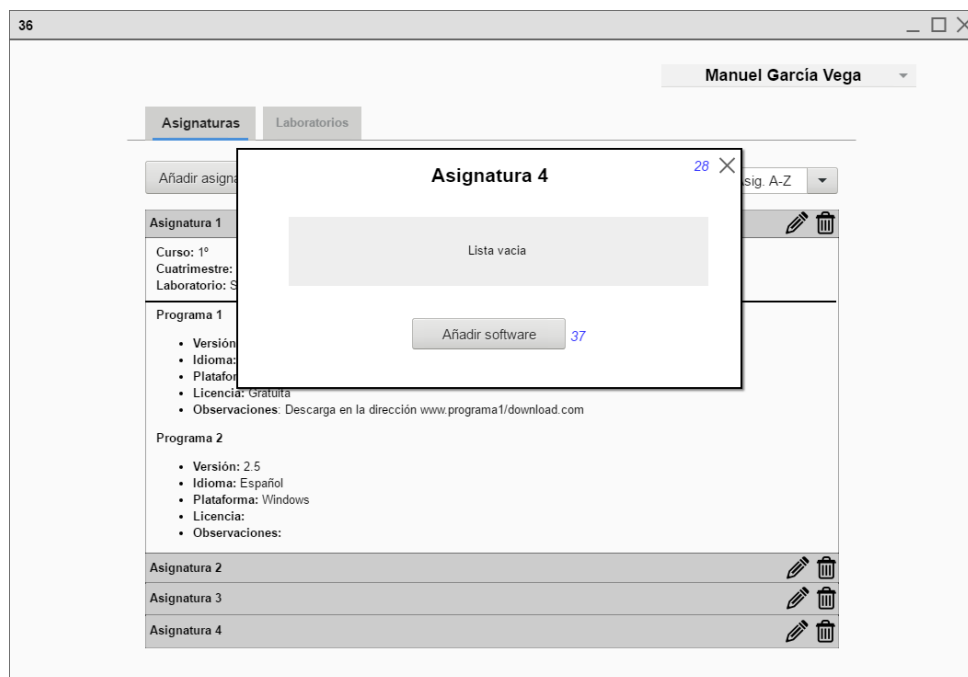
Página 33 - Prototipo en papel 3



Página 34 - Prototipo en papel 3



Página 35 - Prototipo en papel 3



Página 36 - Prototipo en papel 3

37

Manuel García Vega

Asignaturas

Añadir asignatura

Asignatura 1

Curso: 1º
Cuatrimestre: 1º
Laboratorio: Sin as

Programa 1

- Versión: 1.0
- Idioma: Esp
- Plataforma:
- Licencia: Gr
- Observacion

Programa 2

- Versión: 2.5
- Idioma: Esp
- Plataforma:
- Licencia:
- Observacion

Asignatura 2

Asignatura 3

Asignatura 4

Añadir software 28 X

Nombre 37.1

Versión

Plataforma

Idioma

Observaciones

Añadir 38

Asig. A-Z

Asig. A-Z

Asig. A-Z

Asig. A-Z

Página 37 - Prototipo en papel 3

37.1

Manuel García Vega

Asignaturas

Añadir asignatura

Asignatura 1

Curso: 1º
Cuatrimestre: 1º
Laboratorio: Sin as

Programa 1

- Versión: 1.0
- Idioma: Esp
- Plataforma:
- Licencia: Gr
- Observacion

Programa 2

- Versión: 2.5
- Idioma: Esp
- Plataforma:
- Licencia:
- Observacion

Asignatura 2

Asignatura 3

Asignatura 4

Añadir software 28 X

Nombre 37.3

Versión 37.2

Plataforma

Idioma

Idioma

Observaciones

Añadir 38

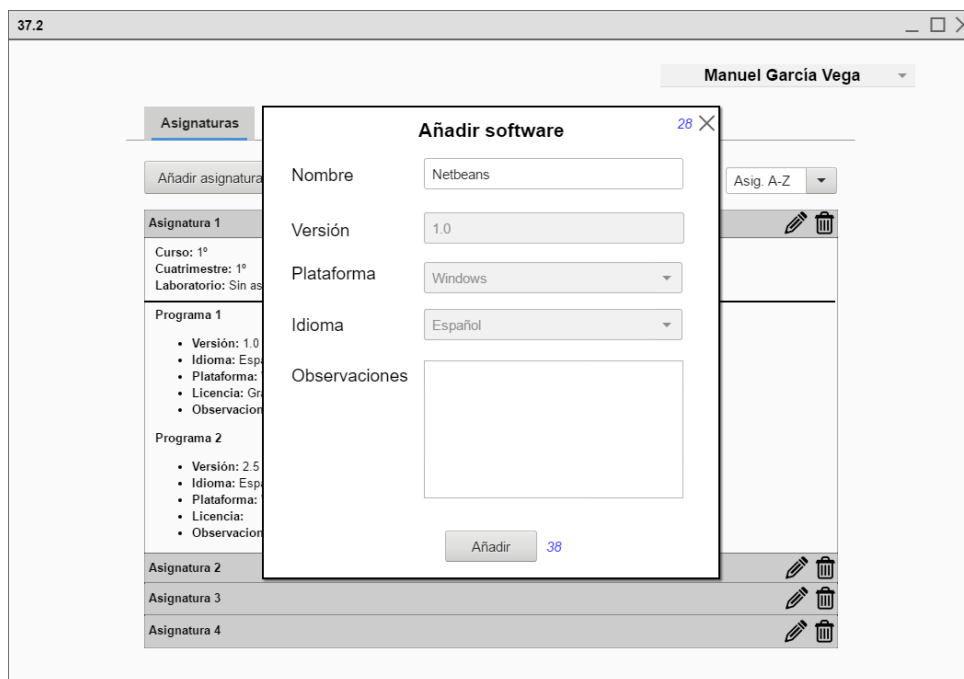
Asig. A-Z

Asig. A-Z

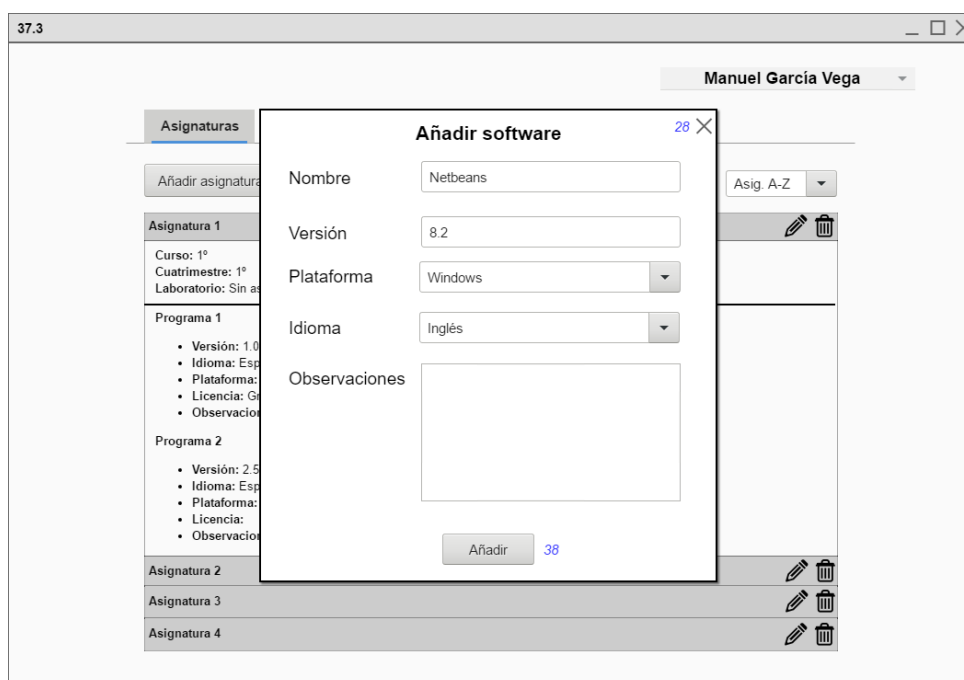
Asig. A-Z

Asig. A-Z

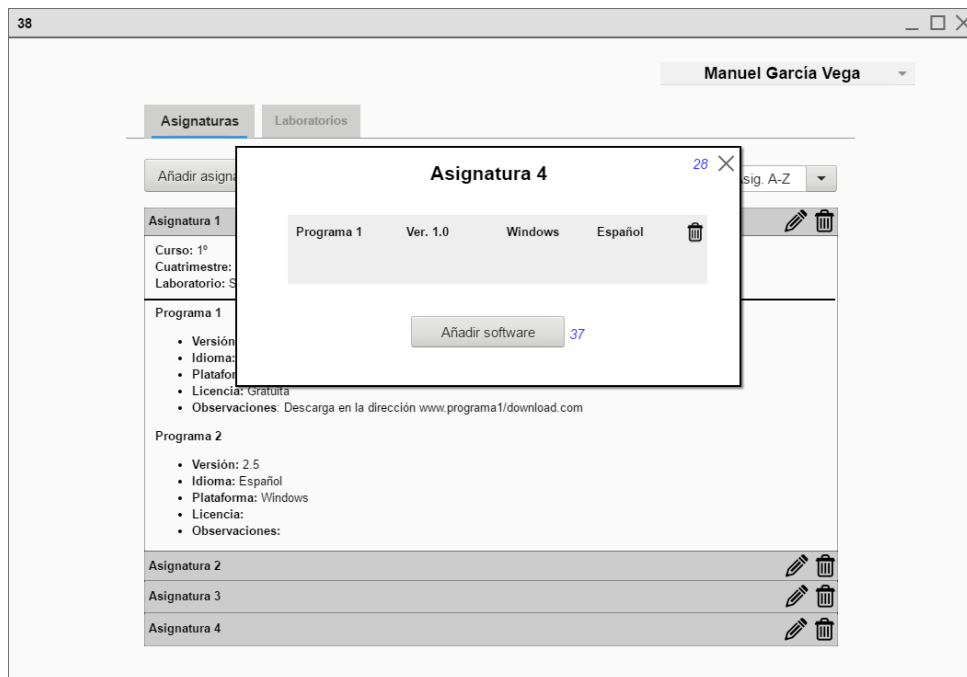
Página 37.1 - Prototipo en papel 3



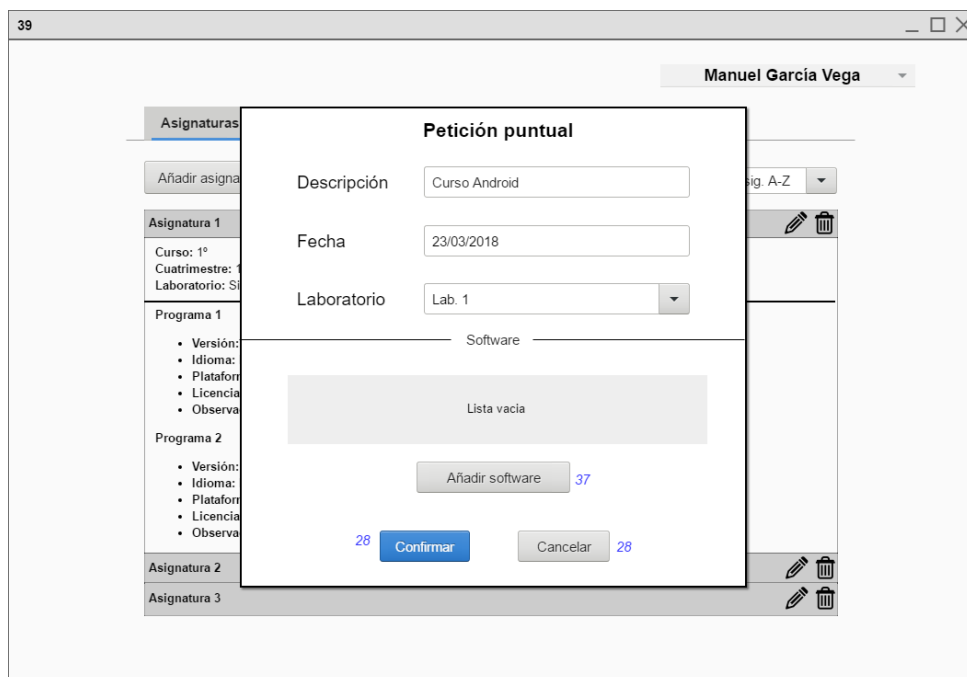
Página 37.2 - Prototipo en papel 3



Página 37.3 - Prototipo en papel 3



Página 38 - Prototipo en papel 3



Página 39 - Prototipo en papel 3

40

Manuel García Vega

Asignaturas Laboratorios

Añadir asignatura

Asignatura 1

Curso: 1º
Cuatrimestre: 1º
Laboratorio: Sin

Programa 1

- Versión: 1
- Idioma: Es
- Plataforma
- Licencia: G
- Observación

Programa 2

- Versión: 2
- Idioma: Es
- Plataforma
- Licencia: G
- Observación

Asignatura 2

Asignatura 3

Editar perfil

Nombre: Manuel

Apellidos: García Vega

Correo electrónico: mgarcia@ujaen.es

Cambiar contraseña

Nueva contraseña

Repetir contraseña

Guardar

Página 40 - Prototipo en papel 3

41

Manuel García Vega

28 41

Asignaturas Laboratorios

Laboratorio 1 (Dep. 172)

Nº de puestos: 20
Observaciones:
[Lista Programas \(2\)](#) 42

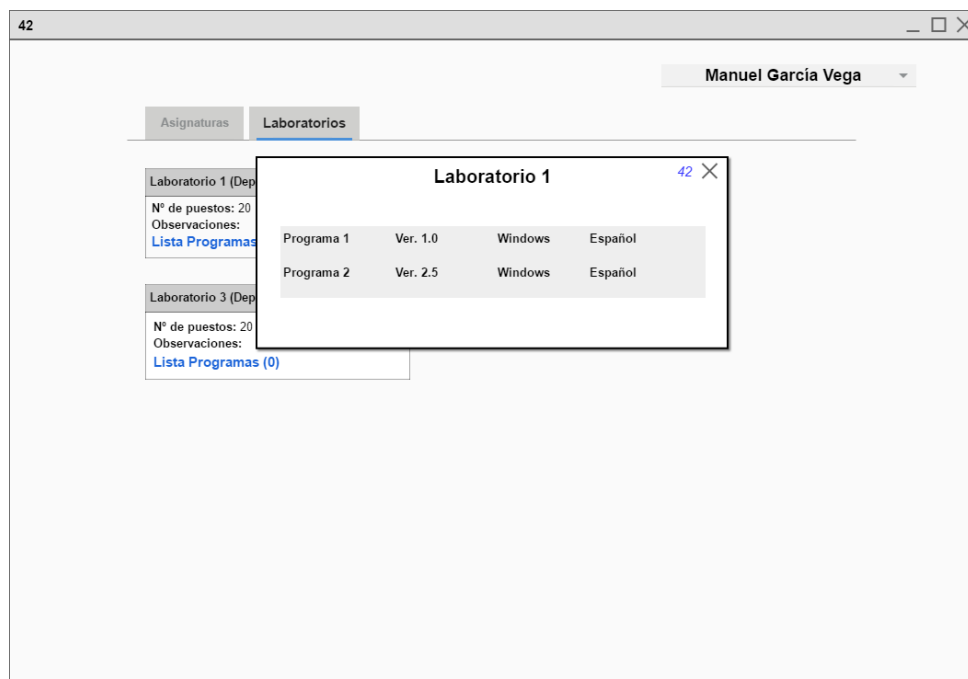
Laboratorio 2 (Dep. 173)

Nº de puestos: 20
Observaciones:
[Lista Programas \(0\)](#)

Laboratorio 3 (Dep. 174)

Nº de puestos: 20
Observaciones:
[Lista Programas \(0\)](#)

Página 41 - Prototipo en papel 3



Página 42 - Prototipo en papel 3

2.5.5. Recorridos cognitivos

La evaluación del prototipo en papel 3 se ha realizado siguiendo la técnica denominada **recorrido cognitivo**.

“...El recorrido cognitivo es un método de inspección de la usabilidad que se centra en evaluar en un diseño su facilidad de aprendizaje, básicamente por exploración y está motivado por la observación que muchos usuarios prefieren aprender software por exploración [1].”

Para la realización de los diferentes recorridos cognitivos se van a seguir una serie de pasos:

1. Seleccionamos una tarea.
2. Ejecución de la tarea por los usuarios.
3. Detectar el éxito o fracaso en el proceso de resolver la tarea.

Como tenemos dos tipos de usuarios que van a usar nuestro sistema las tareas a realizar en los recorridos cognitivos están especificadas para el tipo de usuario al que van dirigidas, si profesor o técnico. Hay que tener en cuenta que estas tareas se realizan en base a un prototipo en papel por lo que no se realizan modificaciones ni acciones reales además, todas las tareas comienzan en la primera página del prototipo.

Tarea 1: Añadir una asignatura a su lista de asignaturas (Profesor)

El usuario comienza iniciando sesión en la web. Tras esto encuentra fácilmente el botón “Añadir asignatura” y procede a “pulsarlo”. A continuación, selecciona la asignatura que desea añadir y finaliza la tarea.

Esta tarea no ha sido difícil de ejecutar por el usuario y se puede destacar la facilidad que ha encontrado el usuario en realizar dicha tarea. Alguna consideración sería modificar el estilo del botón “Añadir asignatura” para que destaque más en la página.

Tarea 2: Añadir software a una asignatura (Profesor)

Tras iniciar sesión el usuario identifica una lista de software en una asignatura, pero no ve a simple vista una interacción para añadir software a una asignatura por lo que se dirige a “pulsar” el icono que identifica como editar. En la siguiente pantalla el usuario encuentra fácilmente el botón “Añadir software” y procede a “pulsarlo”. A continuación, aparece un formulario y comienza a rellenarlo. Reconoce que el campo de nombre le permite

buscar software disponible, aunque al seleccionar el software encuentra un problema al querer modificar la plataforma y el idioma para tener disponible el software en distintas plataformas y en distintos idiomas. Por último, el usuario intenta añadir un software que no aparece en la lista y comprueba que ahora sí están disponibles los campos de plataforma e idioma, pero con la limitación de que sólo puede seleccionar una plataforma y un idioma y finaliza la tarea.

Tras este recorrido cognitivo podemos comprobar que el formulario para añadir software a una asignatura no cumple las expectativas del usuario y debe de ser modificado para admitir varias plataformas y varios idiomas. También hemos comprobado que la metáfora utilizada para representar la edición es adecuada.

Tarea 3: Eliminar software de una asignatura (Profesor)

El usuario comienza iniciando sesión. Para realizar esta tarea el usuario se dirige directamente al icono identificado como editar. En la lista de software reconoce el icono de una papelera y procede a eliminar el software de la asignatura.

En esta sencilla tarea hemos podido comprobar que el usuario ya conoce más el sistema y se ha dirigido directamente al icono de editar. El usuario también ha identificado claramente el icono de eliminar por lo que la metáfora usada para esta operación es correcta.

Tarea 4: Eliminar una asignatura (Profesor)

El usuario inicia sesión y se dirige directamente al icono de eliminar de una de las asignaturas y confirma la eliminación.

Esta tarea no ha sido nada difícil de ejecutar por el usuario.

Tarea 5: Creación de una petición puntual (Profesor)

El usuario inicia sesión e identifica el botón “Petición puntual” rápidamente y se decide a “pulsarlo”. En la siguiente pantalla comienza a rellenar el formulario y finaliza la tarea.

Para esta tarea el usuario no ha mostrado ninguna dificultad, aunque vuelve a hacer hincapié en la posibilidad de marcar distintas plataformas e idiomas al añadir software a la “Petición puntual”.

Tarea 6: Editar perfil (Profesor)

El usuario inicia sesión e identifica en la parte superior derecha de la vista un nombre y decide “pulsarlo”. Se despliega un menú y el usuario encuentra la opción “Editar perfil” sin ningún problema. En la siguiente vista comprueba que puede modificar sus datos (Nombre, Apellidos y teléfono).

El usuario tampoco ha mostrado ninguna dificultad al realizar esta tarea.

Tarea 7: Consultar las tareas pendientes y finalizar una tarea (Técnico)

El usuario inicia sesión y tras examinar la vista confirma encontrarse ante las tareas pendientes. Examina la parte derecha de la vista dónde se encuentran los diferentes filtros para una mejor búsqueda en las tareas. Se encuentra conforme en cuanto al número y al tipo de filtros utilizados, aunque echa en falta un filtro para el curso y continúa buscando un método para finalizar la tarea. El usuario encuentra fácilmente el botón “Finalizar” y se dispone a “pulsarlo”. A continuación, el usuario se encuentra ante una vista que no entiende en un principio, comprende que es para notificar de alguna forma al solicitante de la tarea y tras explicarle que se notificará mediante un correo al solicitante y que puede añadir un mensaje se muestra satisfecho y finaliza la tarea.

En este caso el usuario se encuentra satisfecho con la interfaz, aunque para mejorar la visualización del software para instalar sería mejor colocar los datos en forma de tabla. Se debe incluir un nuevo filtro para el curso y, en la vista para notificar al solicitante, se podría modificar con un mensaje de información que explique lo que sucederá al notificar al solicitante.

Tarea 8: Añadir un nuevo laboratorio, modificarlo y eliminarlo (Técnico)

El usuario comienza nuevamente iniciando sesión y detecta en la parte superior de la vista las diferentes opciones que dispone. Encuentra entre ellas la pestaña laboratorios y tras “pulsar” sobre ella se dirige a una nueva vista. El usuario encuentra fácilmente el botón “Añadir laboratorio” y tras pulsarlo completa el formulario y termina de añadir el nuevo laboratorio. El usuario pasa a editar el laboratorio pulsando en el icono editar y accede a un formulario donde puede editar los datos del laboratorio. También encuentra dónde eliminar el laboratorio, aunque no le gusta el diseño, prefiere un botón normal para eliminar. Por lo tanto, el usuario ya conoce como añadir, modificar y eliminar un laboratorio.

El usuario ha realizado bien la tarea, sin dificultades, aunque se debe modificar el aspecto del botón “Eliminar” de un laboratorio.

Tarea 9: Añadir una nueva asignatura, modificarla y eliminarla (Técnico)

El usuario inicia sesión y se dirige directamente a seleccionar la pestaña “Asignaturas”. Ya en la siguiente vista identifica el botón “Añadir asignatura” y lo “pulsa”. Se encuentra con un formulario que contiene todos los elementos que necesita saber para una asignatura, aunque hace hincapié en que el elemento “Laboratorio” no debe de ser obligatorio rellenarlo al crear la asignatura. A continuación, el usuario vuelve a la vista anterior y se dispone a editar una asignatura, aunque como es similar al caso anterior del laboratorio nos encontramos con el mismo problema en el botón “Eliminar”.

En este caso igual que en el anterior el usuario ha realizado la tarea sin dificultades, le gusta el diseño en tabla para mostrar las asignaturas y sus datos, aunque se debe modificar el botón “Eliminar”.

Tarea 10: Asignar laboratorios para las asignaturas (Técnico)

Tras iniciar sesión y dirigirse a la vista de las asignaturas el usuario “pulsa” el botón “Asignar laboratorios”. Se encuentra ahora ante una nueva vista dónde identifica perfectamente a la izquierda una lista de asignaturas existentes y a la derecha un desplegable para seleccionar un laboratorio de los existentes. Continúa “seleccionando” un laboratorio y comprueba que ahora a la derecha también aparecen las asignaturas que ya tienen ese laboratorio seleccionado y si se seleccionan asignaturas de la izquierda uno de los iconos de las flechas se “activa”. Si “pulsa” dicha flecha activada las asignaturas seleccionadas pasan a la lista del laboratorio. En este punto, el usuario comprende el funcionamiento y comprende que si selecciona asignaturas de la lista del laboratorio la otra flecha se “activará” y dichas asignaturas no estarán asignadas al laboratorio y pasarán a la lista de la izquierda.

Hemos comprobado que esta funcionalidad está bien diseñada ya que intenta ser lo más intuitiva posible, aunque se han detectado problemas en el orden en que los elementos aparecen, es decir, para hacer la tarea en orden es mejor que la parte del laboratorio se encuentre a la izquierda y la lista de asignaturas a la derecha. Ya que al recorrer la vista de izquierda a derecha encontramos antes el laboratorio ya que es el primer paso que debemos de dar es seleccionar un laboratorio.

Tarea 11: Añadir nuevo software y modificarlo (Técnico)

Tras iniciar sesión el usuario se dirige hacia la pestaña “Software”. Una vez allí identifica el botón “Añadir software” y lo “pulsa”. Se encuentra con un formulario que contiene todos los elementos necesarios para describir el software, aunque se deben poder elegir diferentes plataformas y diferentes idiomas para que sea más rápido de introducir nuevo software. Después el usuario vuelve a la vista del software y se dispone a modificarlo. El usuario se dirige a “pulsar” el icono de editar y se encuentra un formulario similar al de añadir software que también habría que modificarlo para varias plataformas e idiomas.

La tarea se ha realizado rápidamente y sólo habría que modificar los formularios para que permitan seleccionar varias plataformas e idiomas.

Tarea 12: Añadir un usuario (Técnico)

El usuario inicia sesión y se dirige a la pestaña “Usuarios”, una vez allí localiza el botón “Añadir usuario”. En la siguiente vista el usuario comprende el funcionamiento para añadir un usuario ya que aparece un mensaje informando de qué va a ocurrir, aunque falta poder seleccionar el tipo de usuario que queremos añadir.

La tarea se ha realizado rápidamente y sólo faltaría añadir un método para seleccionar el tipo de usuario que queremos añadir.

Tarea 13: Editar perfil del usuario (Técnico)

Tras iniciar sesión el usuario se dirige hacia la parte superior derecha donde encuentra un nombre y decide “pulсарlo”. Después se despliega un menú y elige la opción “Editar perfil”. En la siguiente vista comprueba que puede modificar los datos (Nombre, Apellidos y teléfono).

El usuario no ha mostrado ninguna dificultad al realizar la tarea.

2.6. Requisitos del sistema finales

Una vez realizados los diferentes prototipos en papel y tras haber realizado un análisis de los mismos podemos especificar mucho mejor los requisitos de nuestro sistema.

Definimos requisito como “una condición o capacidad exigida por el usuario para solucionar un problema o alcanzar un objetivo”.

Según la clasificación realizada por K. Pohl [4] podemos clasificar los requisitos en dos tipos:

Requisitos funcionales: en ellos se especifica la funcionalidad que el sistema debe proporcionar a los usuarios. Los requisitos funcionales de nuestro sistema son:

1. Identificación de usuario: el usuario debe poder identificarse en la plataforma, introduciendo su correo electrónico y contraseña.
2. Tipos de usuarios: existen dos tipos de usuarios Administrador y Profesor.
3. Añadir un nuevo usuario: los usuarios de tipo Administrador pueden añadir nuevos usuarios facilitando el correo electrónico. Al añadir un usuario se notificará a este con un correo electrónico con los datos necesarios para iniciar sesión.
4. Modificar los datos de un usuario: cada usuario debe ser capaz de modificar sus datos, pero no los de otros usuarios.
5. Visualizar los datos de un usuario: los usuarios de tipo Administrador deben de tener acceso a una lista con los datos de todos los usuarios del sistema.
6. Añadir nuevo software: los usuarios deben poder crear nuevo software aportando los datos necesarios (nombre, versión, plataformas e idiomas).
7. Modificar software: sólo los usuarios de tipo Administrador deben poder modificar los datos de un software concreto.
8. Añadir asignaturas al sistema: sólo los usuarios de tipo Administrador deben poder añadir asignaturas al sistema aportando los datos necesarios (nombre, curso, cuatrimestre).
9. Modificar datos de las asignaturas: sólo los usuarios de tipo Administrador deben ser capaces de modificar los datos de las asignaturas.

10. Eliminar asignaturas del sistema: sólo los usuarios de tipo Administrador deben poder eliminar asignaturas del sistema.
11. Visualizar los datos de las asignaturas: los usuarios de tipo Administrador deben poder acceder a una lista con los datos de todas las asignaturas del sistema.
12. Añadir un laboratorio al sistema: sólo los usuarios de tipo Administrador deben añadir laboratorios al sistema aportando los datos necesarios (nombre, departamento y número de puestos).
13. Modificar los datos de un laboratorio: sólo los usuarios de tipo Administrador deben modificar los datos de los laboratorios.
14. Eliminar un laboratorio del sistema: sólo los usuarios de tipo Administrador deben eliminar un laboratorio del sistema.
15. Consultar el software instalado en un laboratorio concreto: sólo los usuarios de tipo Administrador pueden consultar una lista con el software instalado en un laboratorio.
16. Consultar las tareas pendientes y finalizadas: sólo los usuarios de tipo Administrador pueden consultar una lista de tareas pendientes o finalizadas.
17. Finalizar tareas pendientes: sólo los usuarios de tipo Administrador pueden finalizar una tarea pendiente. La finalización de dicha tarea conlleva a la instalación del software mostrado en la tarea en el laboratorio correcto. Además, al finalizar la tarea se notificará al usuario solicitante que el software ha sido instalado mediante un correo electrónico.
18. Añadir una asignatura a la lista de un usuario de tipo Profesor: sólo los usuarios de dicho tipo pueden añadir asignaturas disponibles en el sistema a su lista de asignaturas.
19. Añadir software a una asignatura: sólo los usuarios de tipo Profesor pueden añadir software (existente o no en el sistema) a las asignaturas disponibles. Además, el usuario dispondrá de un campo para aportar más información si lo desea.
20. Creación de Peticiones puntuales: sólo los usuarios de tipo Profesor pueden crear peticiones puntuales aportando los datos necesarios (descripción, laboratorios y fecha). Dichas peticiones sólo serán visibles por el usuario que las ha creado.
21. Añadir software a una petición puntual: sólo los usuarios de tipo Profesor que han creado dicha petición pueden añadir software a la petición. Además, el usuario dispondrá de un campo para aportar más información si lo desea.

22. Creación de tareas pendientes: la creación de tareas pendientes está disponible para el usuario tipo Profesor cuando tiene una asignatura en su lista de asignaturas y dicha asignatura tiene asignado un software a instalar o bien, ha creado una petición puntual y le ha asignado software. Además, sólo se creará dicha tarea si el usuario ha completado los datos de su perfil.
23. Cerrar la sesión: los usuarios deben poder cerrar la sesión y salir del sistema.

Requisitos de calidad: requisitos no funcionales relacionados con la calidad del sistema en desarrollo.

1. Usabilidad: La aplicación deberá de tener una gran facilidad de uso. El grado de esfuerzo del usuario para interpretar las entradas o salidas del sistema deberá de ser muy bajo.
2. Disponibilidad. El sistema deberá de tener un alto porcentaje de tiempo durante el cual estará plenamente operativo y disponible para usarlo.
3. Confiabilidad: La probabilidad de que el sistema funcione sin fallos debe ser alta.

2.7. Propuesta de solución

Una vez definidos los requisitos que nuestro sistema debe incorporar, se deben diseñar los mecanismos que se utilizarán para llevarlo a cabo. En este caso, se ha decidido aplicar un patrón básico de la arquitectura software, el “Modelo-Vista-Controlador”. Dicho patrón nos permite mantener separadas la parte de interfaz de usuario, y por otra, la lógica de negocio, es decir, la forma en la que los datos son tratados. Este patrón de diseño no garantiza una aplicación robusta independientemente de las tecnologías que se usen.

Dentro de las diferentes opciones para trabajar con MVC se va a recurrir al *framework* de desarrollo “CodeIgniter” [5], en concreto la versión 3.1.8. Se ha decidido usar dicho *framework* por disponer de un código ordenado, facilidad de instalación y porque ya se ha trabajado antes con él.

Como dicho *framework* no nos permite crear un servidor por sí solo para implementar nuestro sistema también utilizaremos “WampServer” [6]. Esta herramienta de desarrollo nos permite crear aplicaciones con Apache, PHP y MySQL. En concreto usaremos la versión 3.1.3.

En el apartado de la interfaz de usuario se ha optado por utilizar los *frameworks* libres “Jquery” [7], “Bootstrap 3”[8] y “FontAwesome”[9]. Los dos primeros *frameworks* nos facilitarán el trabajo puesto que de manera gratuita nos proporcionan distintos elementos para utilizar en nuestras interfaces, además de seguir unos estilos más agradables a la vista. Y el último nos permite usar iconos de forma muy sencilla.

2.8. Modelo de dominio

El *modelo de dominio* [10] es un gráfico conceptual en el que se representan todas las entidades relacionadas en un contexto específico, en el caso de este documento, todos los temas relacionados con la representación de la solución.

En el modelo de dominio se representan los aspectos claves del dominio del problema, así como las relaciones entre las entidades en él representadas.

Para construir el modelo de dominio podemos seguir los siguientes pasos:

1. Identificar las clases conceptuales del sistema.
2. Representar estas clases como clases en un Diagrama de Clases UML.
3. Agregar las relaciones entre estas clases, y los atributos que dan forma a las clases.

A continuación, se detallarán los distintos elementos que componen el modelo de dominio:

- **Objetos:** han de tener una identidad única.
- **Clases:** representan grupos de objetos que comparten las mismas propiedades, comportamientos y relaciones.
- **Asociaciones:** representa la conexión existente entre objetos de un Modelo de Dominio, normalmente de forma física o conceptual.
- **Atributos:** identifican información que pertenece a un objeto.

En la siguiente figura se muestra el diagrama asociado al modelo de dominio del sistema.

Este diagrama se ha realizado a partir del punto de vista del análisis por lo que algunas relaciones podrán ser modificadas o algunos elementos podrán ser eliminados o añadidos.

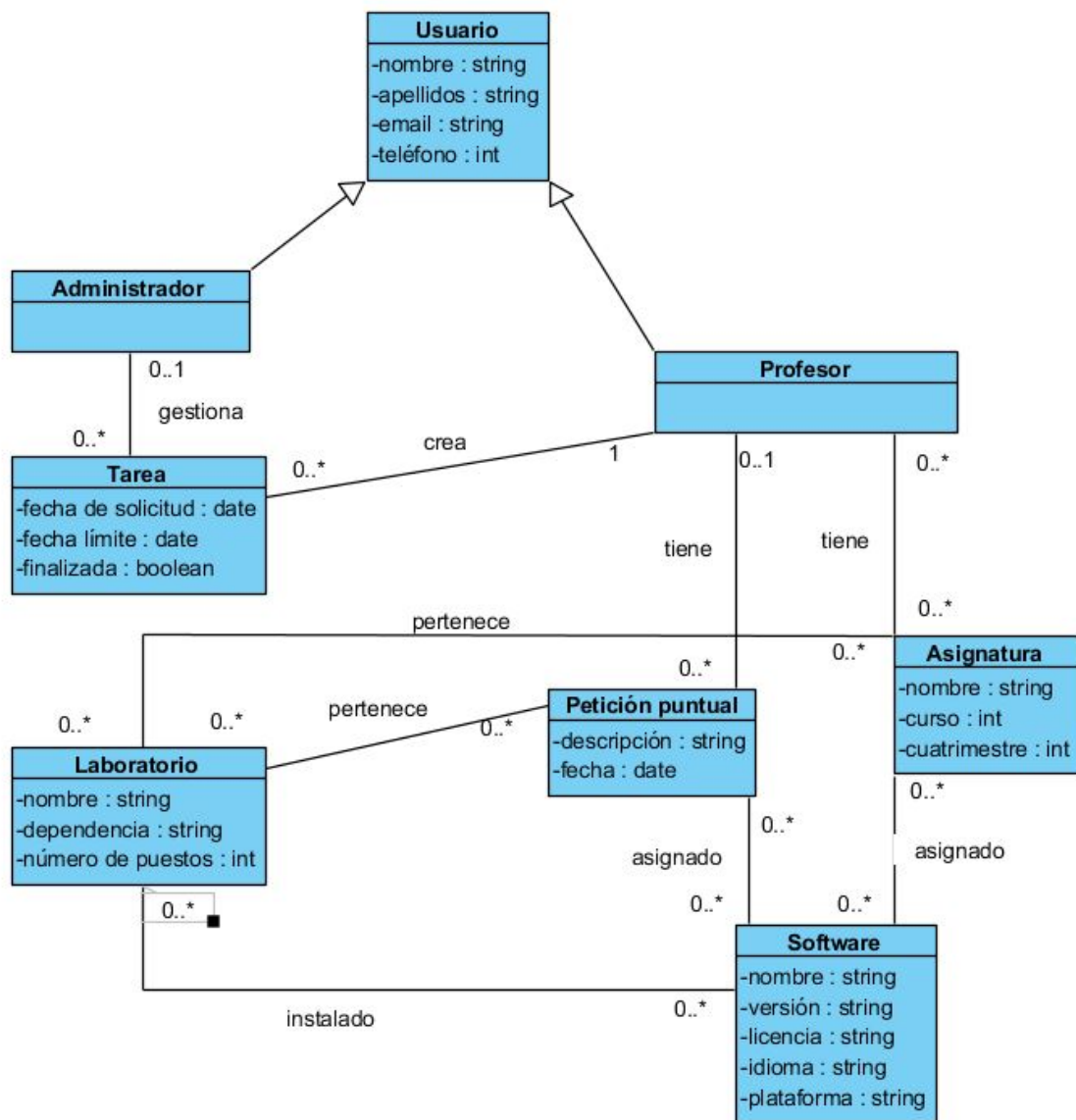


Figura 2.3: Diagrama del modelo de dominio

Capítulo 3

Diseño

Una vez finalizada la fase de análisis de nuestro proyecto pasamos a la fase de diseño. Gracias a la metodología de prototipado ya se ha realizado un diseño de la interfaz de usuario por lo que en este apartado nos centraremos en el diseño del modelo. Comenzaremos con diferentes diagramas para mostrar la estructura de la información.

Hay que destacar que este proceso no se realiza directamente, sino que es un proceso iterativo, que se va ajustando a medida que avanza el proyecto, aunque en esta memoria se desarrolla de forma secuencial.

3.1. Diagrama Entidad-Relación

Definimos el Modelo Entidad-Relación como una herramienta de representación gráfica que incorpora información relacionada con los datos de un sistema, y la relación existente entre estos. Dicha herramienta es necesaria para esbozar el diseño de una base de datos. Partiremos del modelo de dominio anteriormente realizado para obtener la información que debe existir en nuestro sistema, y basándonos en la persistencia de datos la solución final obtenida será una base de datos relacional.

El modelo Entidad-Relación se basa en una serie de objetos básicos, **entidades** y de las **relaciones** entre estos objetos.

- **Entidad:** es una cosa u objeto del mundo real que se distingue de otra según sus características (Figura 3.1).
- **Atributos:** son las “características” de una entidad, sirven para identificarla o describirla.

- **Relaciones:** es una asociación entre varias entidades. Cada relación tiene una “cardinalidad”, que indica el número de entidades con las que otra entidad se puede asociar. Existen tres tipos de relaciones posibles siendo A y B dos entidades relacionadas:
 - Uno a uno: para cada ocurrencia de A, le corresponde una ocurrencia de B (Figura 3.2 a)).
 - Uno a muchos: para cada ocurrencia de A, le pueden corresponder muchas de B, pero para cada ocurrencia de B sólo le corresponde una de A (Figura 3.2 b)).
 - Muchos a uno: para cada ocurrencia de A, le corresponden muchas de B, y viceversa (Figura 3.2 c)).

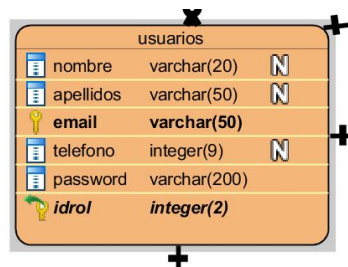


Figura 3.1: Entidad

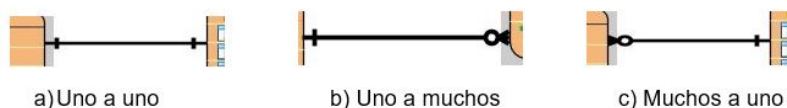


Figura 3.2: Relación con cardinalidad uno a uno

Otro concepto dentro del modelo Entidad-Relación es el de “Claves de una entidad:”

- **Clave primaria:** conjunto de uno o más atributos de una entidad que permite identificar de manera única a cada ocurrencia de dicha entidad (Figura 3.3 a)).
- **Clave foránea:** conjunto de uno o más atributos de una entidad que son clave primaria de otra entidad con la que esta está relacionada (Figura 3.3 b)).

En la figura 3.4 se muestra el diagrama Entidad-Relación perteneciente a la base de datos de nuestro proyecto.



Figura 3.3: Clave primaria

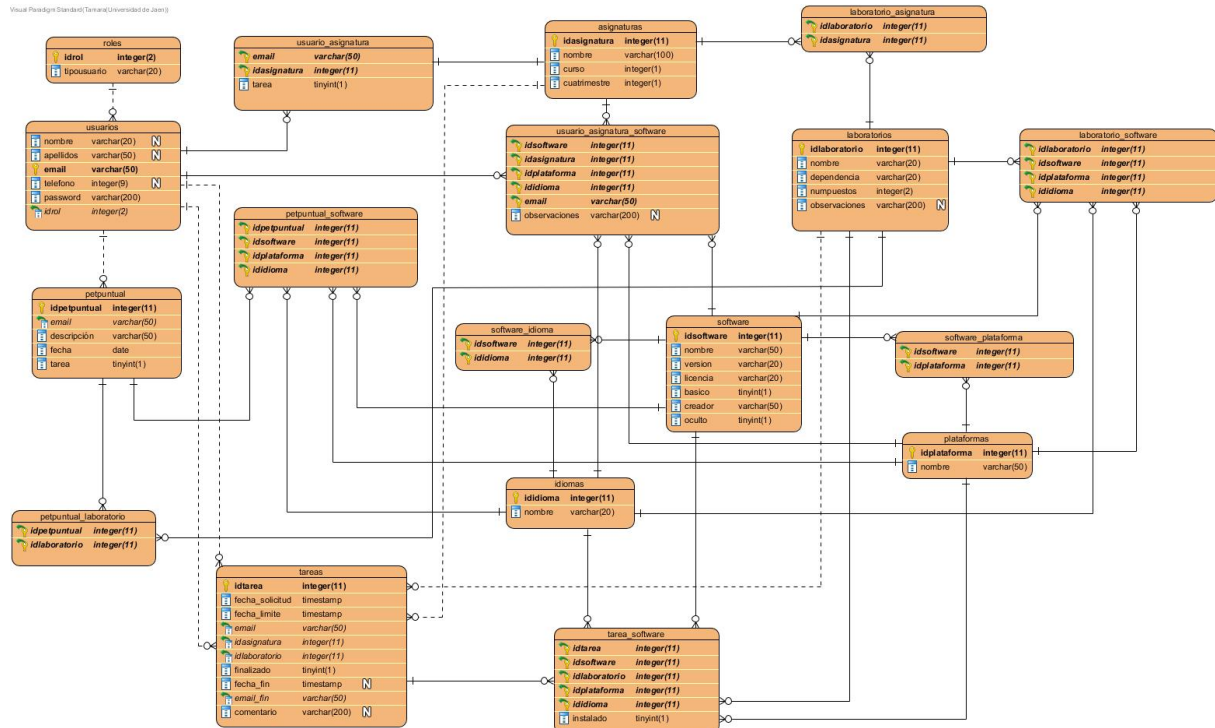


Figura 3.4: Diagrama Entidad-Relación

3.1.1. Definición de las tablas

En este apartado se mostrarán las diferentes tablas a crear en la base de datos, detallando el tipo de dato que almacenarán. El número de tablas totales de nuestro sistema es de **9 tablas que representan las entidades** y de **9 tablas que representan las diferentes relaciones entre las entidades**, en este caso sólo se detallarán las tablas que representan a las entidades.

Campo	Tipo de dato	Clave	Descripción
NOMBRE	Cadena de caracteres	-	Nombre del usuario
APELLIDOS	Cadena de caracteres	-	Apellidos del usuario
TELEFONO	Entero	-	Teléfono del usuario
EMAIL	Cadena de caracteres	Primaria	Correo electrónico del usuario
PASSWORD	Cadena de caracteres	-	Contraseña del usuario
IDROL	Entero	Foránea	Tipo de usuario

Tabla 3.1: Tabla USUARIOS

Campo	Tipo de dato	Clave	Descripción
IDROL	Entero	Primaria	Identificador para el tipo de usuario
TIPOUSUARIO	Cadena de caracteres	-	Nombre dado al tipo de usuario

Tabla 3.2: Tabla ROLES

Campo	Tipo de dato	Clave	Descripción
IDPETPUNTUAL	Entero	Primaria	Identificador para cada petición puntual
DESCRIPCIÓN	Cadena de caracteres	-	Descripción para la petición puntual
EMAIL	Cadena de caracteres	Foránea	Correo electrónico del usuario tipo "Profesor" que ha realizado la petición puntual
FECHA	Fecha	-	Fecha límite para instalar el software de la petición puntual
TAREA	Booleano	-	Marca si existe una tarea asociada a dicha petición puntual

Tabla 3.3: Tabla PETPUNTUAL

Campo	Tipo de dato	Clave	Descripción
IDASIGNATURA	Entero	Primaria	Identificador de la asignatura
NOMBRE	Cadena de caracteres	-	Nombre de la asignatura
CURSO	Entero	-	Curso al que pertenece la asignatura
CUATRIMESTRE	Entero	-	Cuatrimestre al que pertenece la asignatura

Tabla 3.4: Tabla ASIGNATURAS

Campo	Tipo de dato	Clave	Descripción
IDLABORATORIO	Entero	Primaria	Identificador del laboratorio
NOMBRE	Cadena de caracteres	-	Nombre del laboratorio
DEPENDENCIA	Cadena de caracteres	-	Dependencia del laboratorio
NUMPUESTOS	Entero	-	Número de puestos disponibles en el laboratorio
OBSERVACIONES	Cadena de caracteres	-	Campo disponible para aportar más información sobre el laboratorio

Tabla 3.5: Tabla LABORATORIOS

Campo	Tipo de dato	Clave	Descripción
IDSOFTWARE	Entero	Primaria	Identificador del software
NOMBRE	Cadena de caracteres	-	Nombre del software
VERSIÓN	Cadena de caracteres	-	Versión del software
LICENCIA	Cadena de caracteres	-	Tipo de licencia (Gratuita o de pago)
BASICO	Booleano	-	Identifica si el software es básico (si debe de instalarse en todos los laboratorios)
CREADOR	Cadena de caracteres	-	Correo electrónico perteneciente al usuario que insertó el software en la base de datos
OCULTO	Booleano	-	Marca si el software debe de estar oculto o no

Tabla 3.6: Tabla SOFTWARE

Campo	Tipo de dato	Clave	Descripción
IDPLATAFORMA	Entero	Primaria	Identificador de la plataforma
NOMBRE	Cadena de caracteres	-	Nombre de la plataforma

Tabla 3.7: Tabla PLATAFORMAS

Campo	Tipo de dato	Clave	Descripción
IDIDIOMA	Entero	Primaria	Identificador del idioma
NOMBRE	Cadena de caracteres	-	Nombre del idioma

Tabla 3.8: Tabla IDIOMAS

Campo	Tipo de dato	Clave	Descripción
IDTAREA	Entero	Primaria	Identificador de la tarea
FECHA_SOLICITUD	Fecha	-	Fecha en la que se creó la tarea
FECHA_LIMITE	Fecha	-	Fecha límite para completar la tarea
EMAIL	Cadena de caracteres	Foránea	Correo electrónico del usuario tipo profesor que creó la tarea
IDASIGNATURA	Entero	Foránea	Identificador de la asignatura asociada a la tarea
IDLABORATORIO	Entero	Foránea	Identificador del laboratorio asociado a la tarea
FINALIZADO	Booleano	-	Marca si la tarea ha sido finalizada
FECHA_FIN	Fecha	-	Fecha en la que la tarea ha sido finalizada
EMAIL_FIN	Cadena de caracteres	Foránea	Correo electrónico del usuario tipo "Técnico" que ha finalizado la tarea
COMENTARIO	Cadena de caracteres	-	Texto que el usuario tipo "Técnico" puede añadir al finalizar la tarea

Tabla 3.9: Tabla TAREAS

3.2. Diagrama de clases

Un diagrama de clases es una herramienta para comprender cómo se estructura y organiza el código. En esta estructura aparecen las clases creadas y las relaciones entre ellas, además de los atributos y métodos necesarios para realizar la comunicación.

Nuestro proyecto se basa en el patrón arquitectónico Modelo-Vista-Controlador, se ha escogido este patrón por su modularidad ya que podemos identificar tres componentes:

- **Modelo:** clases que contienen los datos, los mecanismos que nos indican cómo trabajar con dichos datos, acceder a la información y actualizarla. También nos permitirá acceder a la base de datos, realizar consultas, etc.
- **Vista:** representan la interfaz de usuario, contienen todo el código HTML, CSS, etc. de la aplicación.
- **Controlador:** comunica las vistas con los modelos, envía los datos que obtiene de

la vista al modelo, y viceversa.

Esta división nos permite separar la interfaz de las interacciones con la base de datos por lo que nos da una mayor rapidez a la hora de programar. En la figura 3.5 se muestra un esquema que representa la comunicación entre los diferentes elementos del patrón y cómo se comunican entre sí y con los demás elementos.

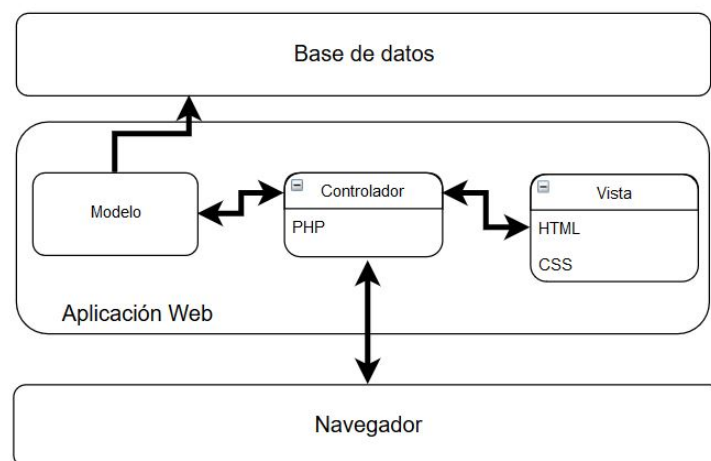


Figura 3.5: Patrón Modelo-Vista-Controlador

Una vez explicado en qué consiste el patrón M-V-C se mostrará cómo está organizado en nuestro proyecto. En la figura 3.6 se pueden distinguir los tres tipos de objetos pertenecientes al patrón organizados en las carpetas: modelos, vistas y controladores. Cada carpeta contiene las diferentes clases que han sido creadas y con flechas se ha representado la conexión entre las distintas clases. Es de destacar que en la carpeta vistas no aparecen clases, sino dos carpetas más, “admin” y “users”, esto es porque en la carpeta vistas no se almacenan clases si no los distintos archivos PHP que muestran la interfaz del usuario.

Se puede apreciar que tanto los controladores como los modelos están agrupados en tres grupos:

- “USERS”: contienen las funciones relacionadas con las operaciones realizadas por los usuarios de tipo “Profesor”.
- “ADMIN”: contiene las funciones relacionadas con las operaciones realizadas por los usuarios de tipo “Administrador”.

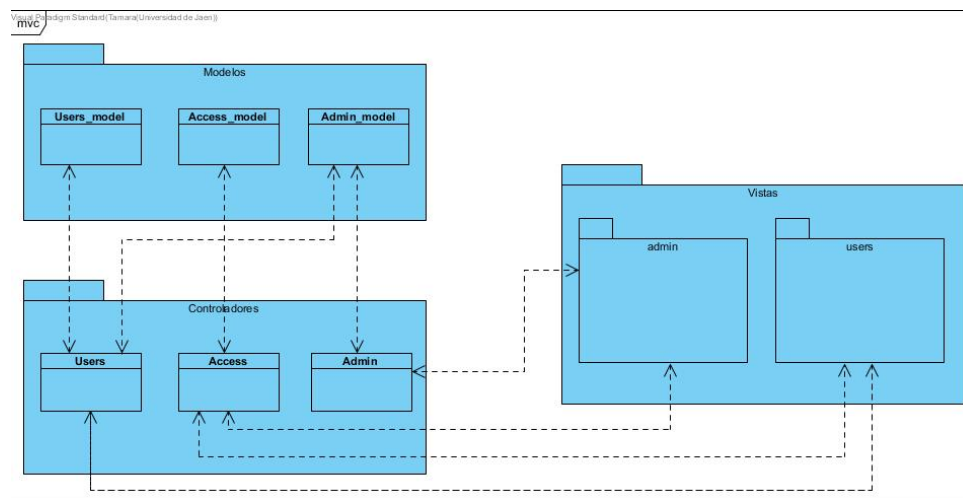


Figura 3.6: Esquema del proyecto con patrón M-V-C

- “ACCESS”: contiene las funciones que permiten autenticar a un usuario e iniciar sesión en el sistema.

Se ha decidido esta organización ya que da mucha más facilidad en el proceso de implementación.

3.3. Modelo de Casos de Uso

“...Un diagrama de caso de uso modela la funcionalidad del sistema según lo perciben los usuarios externos, llamados actores. Un caso de uso es una unidad coherente de funcionalidad, expresada como transición entre los actores y el sistema [11].”

En la figura 3.7 podemos ver un esquema sencillo donde se representan los diferentes componentes de un diagrama de caso de uso.

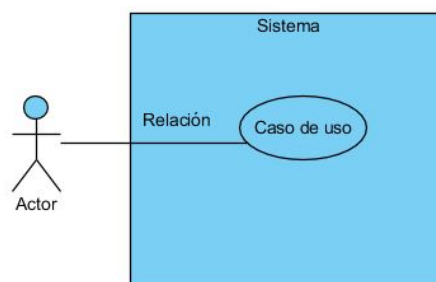


Figura 3.7: Esquema diagrama de caso de uso

Actor

“Un actor es una idealización de una persona externa, un proceso, o de una cosa que interactúa con un sistema, subsistema o clase. Caracteriza las interacciones que los usuarios exteriores pueden tener con el sistema.[...] Un actor puede ser un humano, otro sistema informático, o un cierto proceso ejecutable [11].”

En nuestro sistema podemos encontrar los siguientes actores:

- **Profesor:** es el actor que hace referencia a los usuarios que cumplen el rol “Profesor”.
- **Administrador:** actor que hace referencia a los usuarios que cumplen el rol “Administrador”.

Relaciones La relación existente entre un actor y un caso de uso puede ser de cuatro tipos:

- **Asociación:** comunicación entre un actor y un caso de uso en el que participa.
- **Extensión:** incluye un comportamiento adicional en el caso de uso.
- **Generalización de casos de uso:** relación entre un caso de uso general y uno más específico.
- **Inclusión:** incluye comportamiento adicional, que describe la inserción.

En la figura 3.8 aparece la representación en el diagrama de las diferentes relaciones explicadas anteriormente.

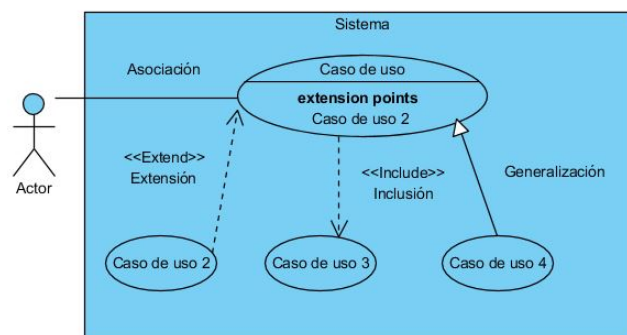


Figura 3.8: Tipos de relaciones de un diagrama de caso de uso

3.3.1. Diagramas de caso de uso

Una vez descritos los diferentes elementos comenzaremos con un diagrama de caso de uso general (Figura 3.9), donde aparecen los diferentes casos de uso pertenecientes a las principales funcionalidades que pueden realizar los actores.

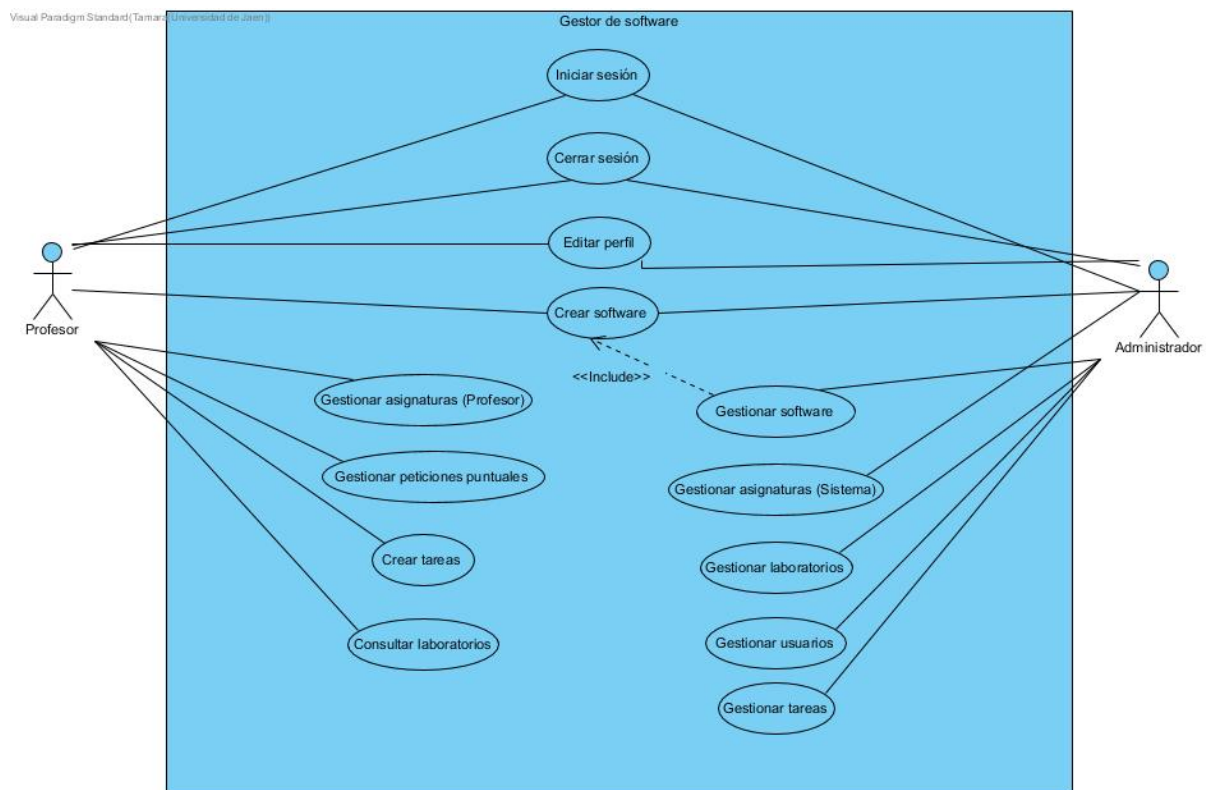
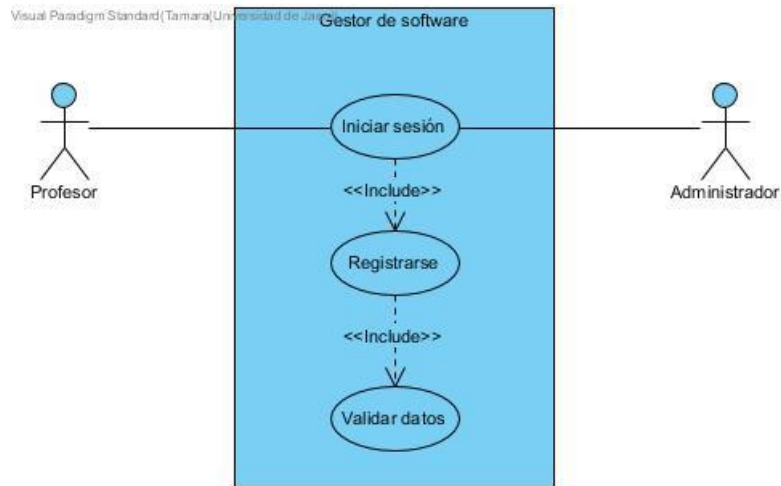
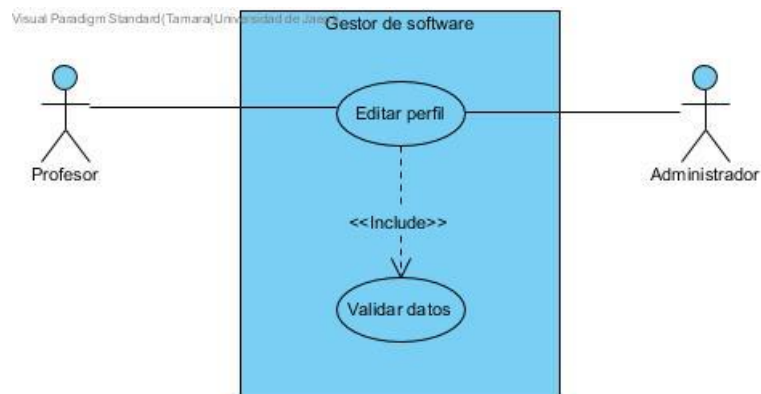
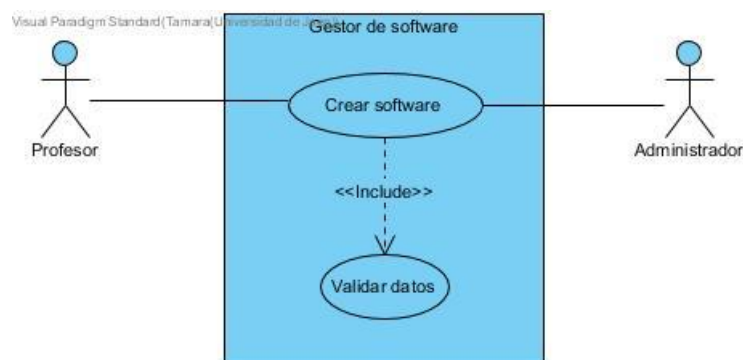


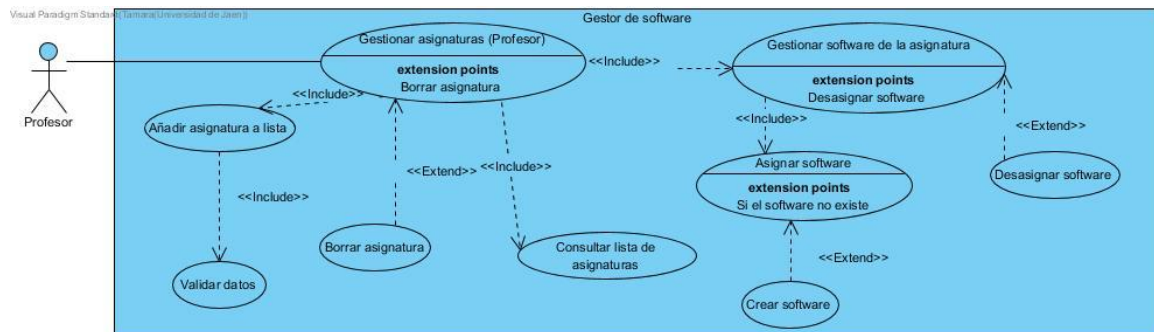
Figura 3.9: Diagrama de caso de uso general

En dicho diagrama aparecen los dos actores mencionados anteriormente y los casos de uso con los que estos están relacionados. Podemos destacar que hay cuatro casos de uso con los que los dos actores mantienen una relación: “Iniciar sesión”, “Cerrar sesión”, “Editar perfil”, y “Crear software”.

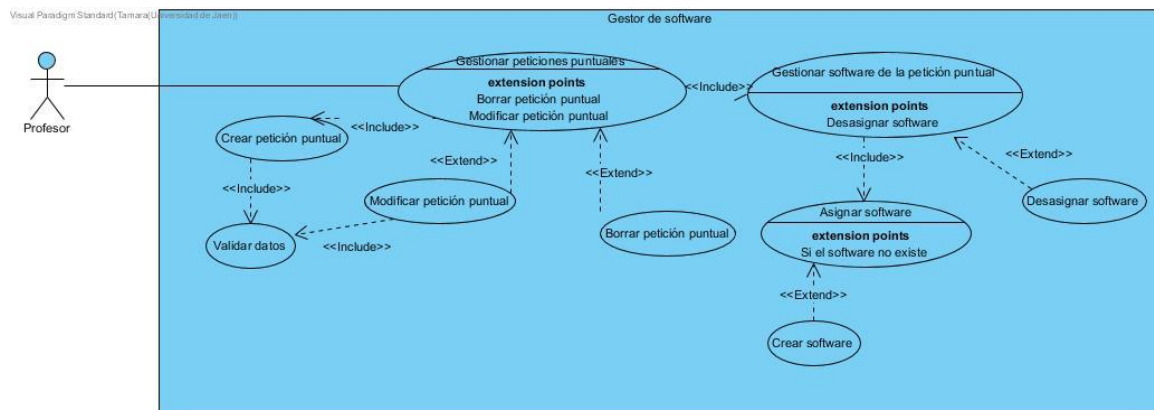
A continuación, se mostrarán los diagramas de los diferentes casos de uso que aparecen el diagrama general para mostrarlos con más detalle. No todos los casos de uso tendrán asociado otro diagrama por simpleza, es decir, no tienen involucrados otros casos de uso.

Caso de uso 1 - Iniciar sesión**Caso de uso 2 - Editar perfil****Caso de uso 3 - Crear software**

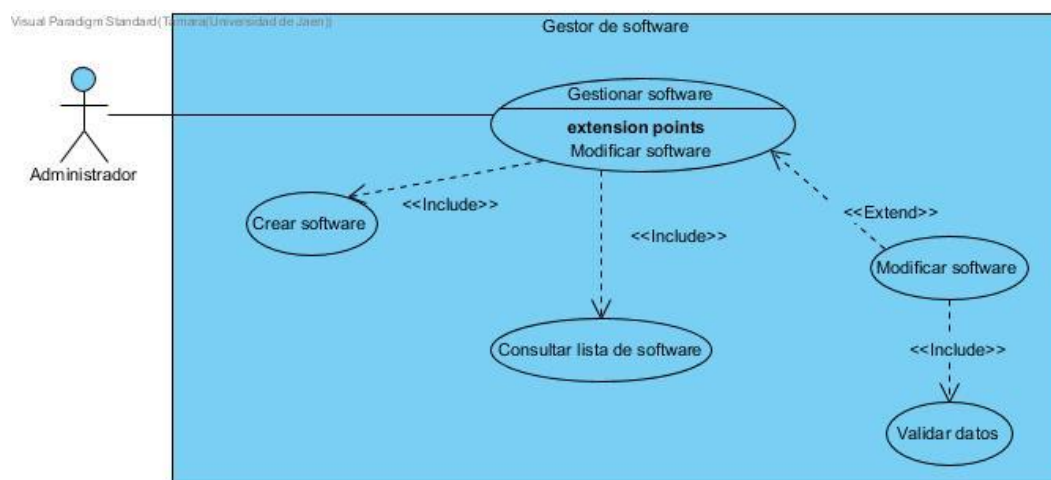
Caso de uso 4 - Gestionar asignaturas (Profesor)



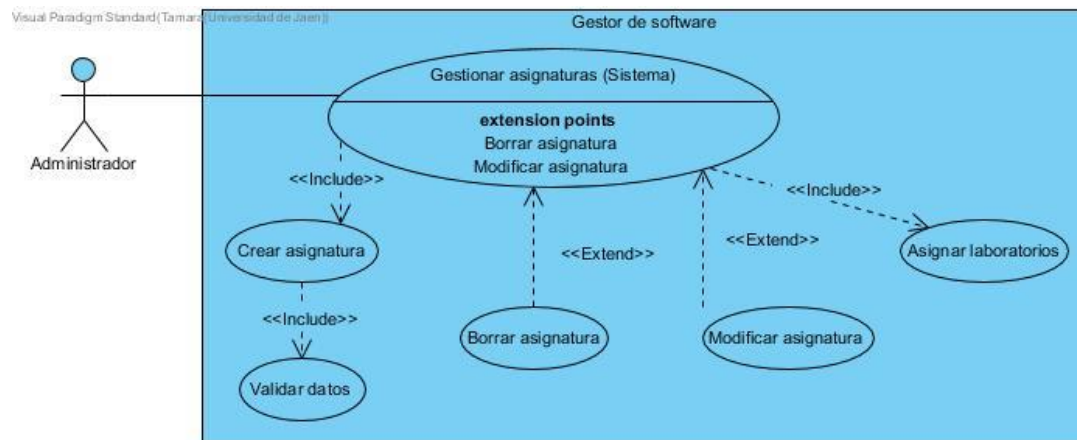
Caso de uso 5 - Gestionar peticiones puntuales



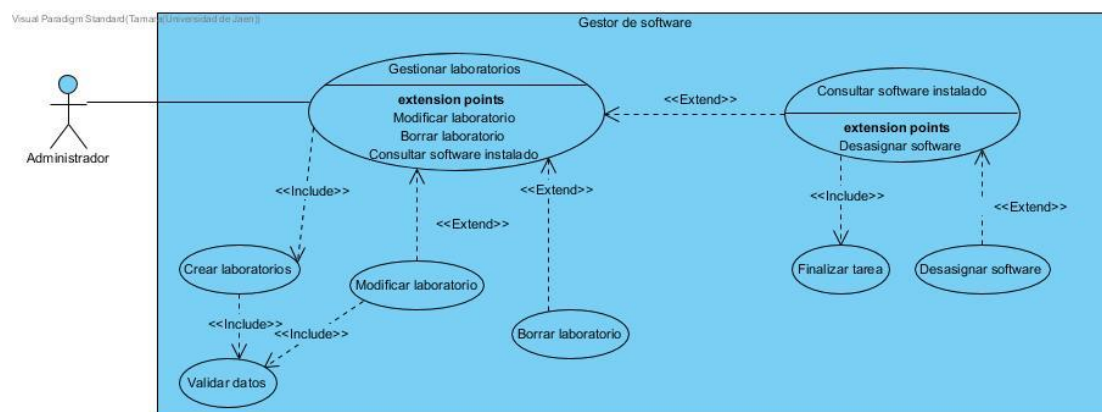
Caso de uso 6 - Gestionar software



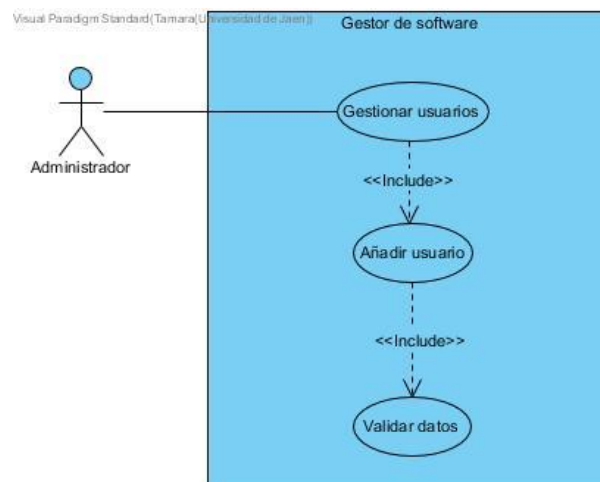
Caso de uso 7 - Gestionar asignaturas (Sistema)

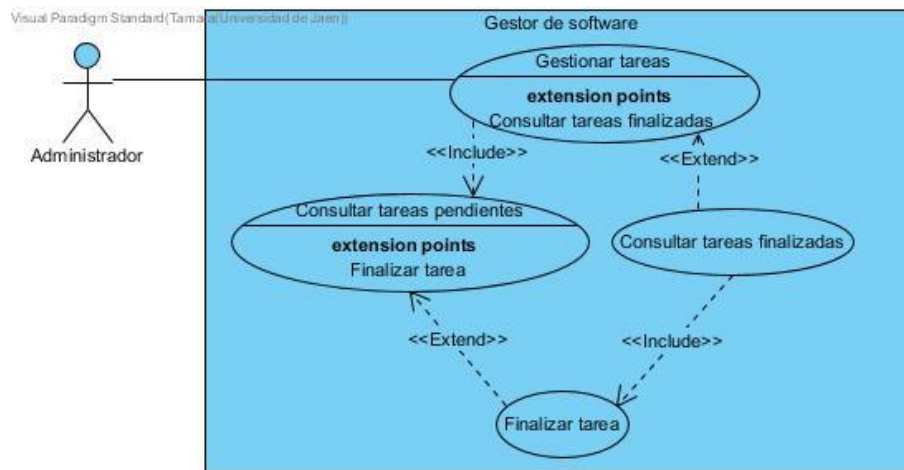


Caso de uso 8 - Gestionar laboratorios



Caso de uso 9 - Gestionar usuarios



Caso de uso 10 - Gestionar tareas

3.3.2. Especificaciones de los casos de uso

“La especificación de los casos de uso se refiere a la descripción de cada una de las partes definidas para lograr su descripción completa [12].”

La especificación se realiza siguiendo el formato representado en la tabla 3.10.

Caso de Uso	Nombre del caso de uso
Actor primario	Actor o actores relacionados
Sistema	Sistema en el que tiene lugar el caso de uso
Participantes	Listado de los actores que intervienen en el caso de uso, así como del sistema en el que tiene lugar
Precondiciones	Condiciones que debe cumplir el sistema antes de iniciar el caso de uso
Descripción	Resumen del objetivo del caso de uso
Operaciones básicas	Pasos a seguir en el proceso de ejecución
Alternativas	Pasos alternativos a seguir

Tabla 3.10: Tabla modelo para la especificación de un caso de uso

Especificación del Caso de Uso: Iniciar sesión	
Caso de Uso	Iniciar sesión
Actor primario	Profesor Administrador
Sistema	Gestor de software
Participantes	Profesor - Administrador - Sistema
Precondiciones	El usuario (Profesor o Administrador) debe estar registrado en el sistema
Descripción	Un usuario accede a la página principal de la plataforma para identificarse y poder acceder al contenido de la misma
Operaciones básicas	
1	El usuario accede a la página web de la plataforma
2	Introduce sus datos de usuario: correo electrónico y contraseña
3	¿Los datos son correctos?
4	El sistema comprueba el rol del usuario y lo redirecciona a la página adecuada
5	El usuario accede al contenido de la plataforma
Alternativas	
3.1	Si los datos son incorrectos el sistema envía un mensaje de error y redirecciona al paso 2
3.2	Si los datos son correctos, ir a paso 4
3.3	Si el usuario no está registrado el sistema manda un mensaje de error y redirecciona al paso 2

Tabla 3.11: Especificación de caso de uso: Iniciar sesión

Especificación del Caso de Uso: Crear software

Caso de Uso	Crear software
Actor primario	Profesor Administrador
Sistema	Gestor de software
Participantes	Profesor - Administrador - Sistema
Precondiciones	El usuario debe de estar registrado en la plataforma El usuario 'Profesor' debe tener una asignatura en su lista de asignaturas
Descripción	Los usuarios pueden añadir un nuevo software a la base de datos del sistema
Operaciones básicas	
1	El usuario accede a la página web de la plataforma
2	Introduce sus datos de usuario: correo electrónico y contraseña
3	¿Los datos son correctos?
4	El sistema comprueba el rol del usuario y lo redirecciona a la página adecuada
5	El usuario accede al contenido de la plataforma
6	El usuario añade un nuevo software
7	El sistema añade el software a la base de datos
Alternativas	
3.1	Si los datos son incorrectos el sistema envía un mensaje de error y redirecciona al paso 2
3.2	Si los datos son correctos, ir a paso 4
3.3	Si el usuario no está registrado el sistema manda un mensaje de error y redirecciona al paso 2
5.1	Si el usuario es Administrador accede a "Software", después ir a paso 6
5.2	Si el usuario es Profesor accede a "Añadir software" y después a "No existe el software", pasar a paso 6
6.1	Si los datos son correctos, ir a paso 7
6.2	Si los datos no son correctos, el sistema se mantiene en la misma página

Tabla 3.12: Especificación de caso de uso: Crear software

Especificación del Caso de Uso: Consultar lista de Asignaturas	
Caso de Uso	Consultar lista de asignaturas
Actor primario	Profesor
Sistema	Gestor de software
Participantes	Profesor - Sistema
Precondiciones	El usuario debe estar registrado en el sistema El usuario debe ser rol tipo “Profesor” El usuario debe tener asignada al menos una asignatura
Descripción	El usuario consulta su lista de asignaturas
Operaciones básicas	
1	El usuario accede a la página web de la plataforma
2	Introduce sus datos de usuario: correo electrónico y contraseña
3	¿Los datos son correctos?
4	El sistema comprueba el rol del usuario y lo redirecciona a la página adecuada
5	El usuario accede al contenido de la plataforma
6	El usuario consulta la lista de asignaturas asignadas
Alternativas	
3.1	Si los datos son incorrectos el sistema envía un mensaje de error y redirecciona al paso 2
3.2	Si los datos son correctos, ir a paso 4
3.3	Si el usuario no está registrado el sistema manda un mensaje de error y redirecciona al paso 2

Tabla 3.13: Especificación de caso de uso: Consultar lista de asignaturas

Especificación del Caso de Uso: Asignar asignaturas a lista	
Caso de Uso	Asignar asignaturas a lista
Actor primario	Profesor
Sistema	Gestor de software
Participantes	Profesor - Sistema
Precondiciones	El usuario debe estar registrado en el sistema El usuario debe ser rol tipo "Profesor"
Descripción	El usuario asigna una nueva asignatura a su lista de asignaturas
Operaciones básicas	
1	El usuario accede a la página web de la plataforma
2	Introduce sus datos de usuario: correo electrónico y contraseña
3	¿Los datos son correctos?
4	El sistema comprueba el rol del usuario y lo redirecciona a la página adecuada
5	El usuario accede al contenido de la plataforma
6	El usuario acceda a "Añadir asignatura"
7	El usuario escoge la asignatura deseada
8	El sistema realiza las operaciones necesarias y redirige al usuario a su lista de asignaturas
Alternativas	
3.1	Si los datos son incorrectos el sistema envía un mensaje de error y redirecciona al paso 2
3.2	Si los datos son correctos, ir a paso 4
3.3	Si el usuario no está registrado el sistema manda un mensaje de error y redirecciona al paso 2
7.1	Si los datos son correctos el sistema manda un mensaje informativo, ir a paso 8
7.2	Si los datos no son correctos el sistema se mantiene en la misma página
7.3	Si ocurre un error en el proceso el sistema manda un mensaje de error y vuelve a la lista de asignaturas del usuario

Tabla 3.14: Especificación de caso de uso: Asignar asignaturas a lista

Especificación del Caso de Uso: Borrar asignatura de la lista	
Caso de Uso	Borrar asignatura de la lista
Actor primario	Profesor
Sistema	Gestor de software
Participantes	Profesor - Sistema
Precondiciones	El usuario debe estar registrado en el sistema El usuario debe ser rol tipo "Profesor" El usuario debe tener asignada al menos una asignatura
Descripción	El usuario elimina una asignatura de su lista
Operaciones básicas	
1	El usuario accede a la página web de la plataforma
2	Introduce sus datos de usuario: correo electrónico y contraseña
3	¿Los datos son correctos?
4	El sistema comprueba el rol del usuario y lo redirecciona a la página adecuada
5	El usuario accede al contenido de la plataforma
6	El usuario accede al menú de la asignatura
7	El usuario escoge "Borrar"
8	El usuario confirma la eliminación
9	El sistema realiza las operaciones necesarias y redirige a la lista de asignaturas del usuario
Alternativas	
3.1	Si los datos son incorrectos el sistema envía un mensaje de error y redirecciona al paso 2
3.2	Si los datos son correctos, ir a paso 4
3.3	Si el usuario no está registrado el sistema manda un mensaje de error y redirecciona al paso 2
9.1	Si todo es correcto, el sistema manda un mensaje de todo correcto
9.2	Si ocurre algún problema, el sistema manda un mensaje de error

Tabla 3.15: Especificación de caso de uso: Borrar asignatura de la lista

Especificación del Caso de Uso: Asignar software	
Caso de Uso	Asignar software
Actor primario	Profesor
Sistema	Gestor de software
Participantes	Profesor - Sistema
Precondiciones	El usuario debe estar registrado en el sistema El usuario debe ser rol tipo “Profesor” El usuario debe tener asignada al menos una asignatura
Descripción	El usuario desea asignar un software a una de las asignaturas de su lista de asignaturas
Operaciones básicas	
1	El usuario accede a la página web de la plataforma
2	Introduce sus datos de usuario: correo electrónico y contraseña
3	¿Los datos son correctos?
4	El sistema comprueba el rol del usuario y lo redirecciona a la página adecuada
5	El usuario accede al menú de la asignatura
6	El usuario escoge “Añadir software”
7	El usuario introduce los datos
8	El sistema realiza las operaciones necesarias y redirige a la lista de asignaturas
Alternativas	
3.1	Si los datos son incorrectos el sistema envía un mensaje de error y redirecciona al paso 2
3.2	Si los datos son correctos, ir a paso 4
3.3	Si el usuario no está registrado el sistema manda un mensaje de error y redirecciona al paso 2
5.1	Si el software que desea se encuentra disponible, ir a paso 6
5.2	Si el software no se encuentra, el sistema redirige a “Nuevo software” e ir a paso 6
7.1	Si los datos son correctos, ir a paso 7
7.2	Si los datos no son correctos, el sistema mantiene al usuario en la misma página
8.1	Si todo es correcto, el sistema manda un mensaje de todo correcto
8.2	Si se produce un error en el proceso, el sistema manda un mensaje de error

Tabla 3.16: Especificación de caso de uso: Asignar software

Especificación del Caso de Uso: Crear petición puntual

Caso de Uso	Crear petición puntual
Actor primario	Profesor
Sistema	Gestor de software
Participantes	Profesor - Sistema
Precondiciones	El usuario debe estar registrado en el sistema El usuario debe ser rol tipo "Profesor"
Descripción	El usuario crea una nueva petición puntual
Operaciones básicas	
1	El usuario accede a la página web de la plataforma
2	Introduce sus datos de usuario: correo electrónico y contraseña
3	¿Los datos son correctos?
4	El sistema comprueba el rol del usuario y lo redirecciona a la página adecuada
5	El usuario accede al apartado "Petición puntual"
6	El sistema redirige a la lista de peticiones puntuales
7	El usuario accede a "Añadir petición puntual"
8	El usuario rellena los datos
9	El sistema realiza las operaciones necesarias y redirige a la lista de asignaturas
Alternativas	
3.1	Si los datos son incorrectos el sistema envía un mensaje de error y redirecciona al paso 2
3.2	Si los datos son correctos, ir a paso 4
3.3	Si el usuario no está registrado el sistema manda un mensaje de error y redirecciona al paso 2
8.1	Si los datos son correctos, ir a paso 9
8.2	Si los datos no son correctos, el sistema mantiene al usuario en la misma página
9.1	Si todo es correcto, el sistema manda un mensaje de todo correcto
9.2	Si se produce un error en el proceso, el sistema manda un mensaje de error

Tabla 3.17: Especificación de caso de uso: Crear petición puntual

Especificación del Caso de Uso: Modificar petición puntual	
Caso de Uso	Modificar petición puntual
Actor primario	Profesor
Sistema	Gestor de software
Participantes	Profesor - Sistema
Precondiciones	El usuario debe estar registrado en el sistema El usuario debe ser rol tipo “Profesor” El usuario debe tener una “Petición puntual” creada
Descripción	El usuario modifica una petición puntual
Operaciones básicas	
1	El usuario accede a la página web de la plataforma
2	Introduce sus datos de usuario: correo electrónico y contraseña
3	¿Los datos son correctos?
4	El sistema comprueba el rol del usuario y lo redirecciona a la página adecuada
5	El usuario accede al apartado “Peticiones puntuales”
6	El sistema redirige a la lista de peticiones puntuales
7	El usuario accede al menú de la petición puntual
8	El usuario escoge a “Editar”
9	El usuario modifica los datos
10	El sistema realiza las operaciones necesarias y redirige a la lista de asignaturas
Alternativas	
3.1	Si los datos son incorrectos el sistema envía un mensaje de error y redirecciona al paso 2
3.2	Si los datos son correctos, ir a paso 4
3.3	Si el usuario no está registrado el sistema manda un mensaje de error y redirecciona al paso 2
9.1	Si los datos son correctos, ir a paso 10
9.2	Si los datos no son correctos, el sistema mantiene al usuario en la misma página
10.1	Si todo es correcto, el sistema manda un mensaje de todo correcto
10.2	Si se produce un error en el proceso, el sistema manda un mensaje de error

Tabla 3.18: Especificación de caso de uso: Modificar petición puntual

Especificación del Caso de Uso: Borrar petición puntual

Caso de Uso	Borrar petición puntual
Actor primario	Profesor
Sistema	Gestor de software
Participantes	Profesor - Sistema
Precondiciones	El usuario debe estar registrado en el sistema El usuario debe ser rol tipo "Profesor" El usuario debe tener una "Petición puntual" creada
Descripción	El usuario elimina una petición puntual
Operaciones básicas	
1	El usuario accede a la página web de la plataforma
2	Introduce sus datos de usuario: correo electrónico y contraseña
3	¿Los datos son correctos?
4	El sistema comprueba el rol del usuario y lo redirige a la página adecuada
5	El usuario accede al apartado "Peticiones puntuales"
6	El sistema redirige a la lista de peticiones puntuales
7	El usuario accede al menú de la petición puntual
8	El usuario accede a "Borrar"
9	El usuario confirma la eliminación
10	El sistema realiza las operaciones necesarias y redirige a la lista de asignaturas
Alternativas	
3.1	Si los datos son incorrectos el sistema envía un mensaje de error y redirige al paso 2
3.2	Si los datos son correctos, ir a paso 4
3.3	Si el usuario no está registrado el sistema manda un mensaje de error y redirige al paso 2
10.1	Si todo es correcto, el sistema manda un mensaje de todo correcto
10.2	Si se produce un error en el proceso, el sistema manda un mensaje de error

Tabla 3.19: Especificación de caso de uso: Borrar petición puntual

Especificación del Caso de Uso: Consultar lista de software	
Caso de Uso	Consultar lista de software
Actor primario	Administrador
Sistema	Gestor de software
Participantes	Administrador - Sistema
Precondiciones	El usuario debe estar registrado en el sistema El usuario debe ser rol tipo “Administrador” Debe existir al menos un “software” en el sistema
Descripción	El usuario consulta la lista de software disponible
Operaciones básicas	
1	El usuario accede a la página web de la plataforma
2	Introduce sus datos de usuario: correo electrónico y contraseña
3	¿Los datos son correctos?
4	El sistema comprueba el rol del usuario y lo redirecciona a la página adecuada
5	El usuario accede al apartado “Software”
6	El sistema redirige a la lista de software
Alternativas	
3.1	Si los datos son incorrectos el sistema envía un mensaje de error y redirecciona al paso 2
3.2	Si los datos son correctos, ir a paso 4
3.3	Si el usuario no está registrado el sistema manda un mensaje de error y redirecciona al paso 2

Tabla 3.20: Especificación de caso de uso: Consultar lista de software

Especificación del Caso de Uso: Modificar software	
Caso de Uso	Modificar software
Actor primario	Administrador
Sistema	Gestor de software
Participantes	Administrador - Sistema
Precondiciones	El usuario debe estar registrado en el sistema El usuario debe ser rol tipo "Administrador" Debe existir al menos un "software" en el sistema
Descripción	El usuario modifica un "software" de la lista de software disponible
Operaciones básicas	
1	El usuario accede a la página web de la plataforma
2	Introduce sus datos de usuario: correo electrónico y contraseña
3	¿Los datos son correctos?
4	El sistema comprueba el rol del usuario y lo redirecciona a la página adecuada
5	El usuario accede al apartado "Software"
6	El sistema redirige a la lista de software
7	El usuario accede al menú de edición
8	El usuario modifica los datos
9	El sistema realiza las operaciones necesarias y redirige a la pantalla inicial
Alternativas	
3.1	Si los datos son incorrectos el sistema envía un mensaje de error y redirecciona al paso 2
3.2	Si los datos son correctos, ir a paso 4
3.3	Si el usuario no está registrado el sistema manda un mensaje de error y redirecciona al paso 2
8.1	Si los datos son correctos, ir a paso 9
8.2	Si los datos no son correctos, el sistema se mantiene en la misma página
9.1	Si todo finaliza correctamente, el sistema manda un mensaje de todo correcto
9.2	Si se produce un error en el proceso, el sistema manda un mensaje de error

Tabla 3.21: Especificación de caso de uso: Modificar software

Especificación del Caso de Uso: Crear asignatura	
Caso de Uso	Crear asignatura
Actor primario	Administrador
Sistema	Gestor de software
Participantes	Administrador - Sistema
Precondiciones	El usuario debe estar registrado en el sistema El usuario debe ser rol tipo "Administrador"
Descripción	El usuario añade una nueva asignatura al sistema
Operaciones básicas	
1	El usuario accede a la página web de la plataforma
2	Introduce sus datos de usuario: correo electrónico y contraseña
3	¿Los datos son correctos?
4	El sistema comprueba el rol del usuario y lo redirecciona a la página adecuada
5	El usuario accede al apartado "Asignaturas"
6	El sistema redirige a la lista de asignaturas
7	El usuario accede a "Añadir asignatura"
8	El usuario completa los datos
9	El sistema realiza las operaciones necesarias y redirige a la página principal
Alternativas	
3.1	Si los datos son incorrectos el sistema envía un mensaje de error y redirecciona al paso 2
3.2	Si los datos son correctos, ir a paso 4
3.3	Si el usuario no está registrado el sistema manda un mensaje de error y redirecciona al paso 2
8.1	Si los datos son correctos, ir a paso 9
8.2	Si los datos no son correctos, el sistema se mantiene en la misma página
9.1	Si todo sale correcto, el sistema manda un mensaje de todo correcto
9.2	Si se produce un error, el sistema manda un mensaje de error

Tabla 3.22: Especificación de caso de uso: Crear asignatura

Especificación del Caso de Uso: Modificar asignatura	
Caso de Uso	Modificar asignatura
Actor primario	Administrador
Sistema	Gestor de software
Participantes	Administrador - Sistema
Precondiciones	El usuario debe estar registrado en el sistema El usuario debe ser rol tipo "Administrador" Debe existir al menos una asignatura en el sistema
Descripción	El usuario modifica una asignatura
Operaciones básicas	
1	El usuario accede a la página web de la plataforma
2	Introduce sus datos de usuario: correo electrónico y contraseña
3	¿Los datos son correctos?
4	El sistema comprueba el rol del usuario y lo redirecciona a la página adecuada
5	El usuario accede al apartado "Asignaturas"
6	El sistema redirige a la lista de asignaturas
7	El usuario accede al icono de edición
8	El usuario modifica los datos
9	El sistema realiza las operaciones necesarias y redirige a la pantalla inicial
Alternativas	
3.1	Si los datos son incorrectos el sistema envía un mensaje de error y redirecciona al paso 2
3.2	Si los datos son correctos, ir a paso 4
3.3	Si el usuario no está registrado el sistema manda un mensaje de error y redirecciona al paso 2
8.1	Si los datos son correctos, ir a paso 9
8.2	Si los datos no son correctos, el sistema se mantiene en la misma página
9.1	Si todo finaliza correctamente, el sistema manda un mensaje de todo correcto
9.2	Si se produce un error en el proceso, el sistema manda un mensaje de error

Tabla 3.23: Especificación de caso de uso: Modificar asignatura

Especificación del Caso de Uso: Borrar asignatura	
Caso de Uso	Borrar asignatura
Actor primario	Administrador
Sistema	Gestor de software
Participantes	Administrador - Sistema
Precondiciones	El usuario debe estar registrado en el sistema El usuario debe ser rol tipo "Administrador" Debe existir al menos una asignatura en el sistema
Descripción	El usuario elimina una asignatura del sistema
Operaciones básicas	
1	El usuario accede a la página web de la plataforma
2	Introduce sus datos de usuario: correo electrónico y contraseña
3	¿Los datos son correctos?
4	El sistema comprueba el rol del usuario y lo redirecciona a la página adecuada
5	El usuario accede al apartado "Asignaturas"
6	El sistema redirige a la lista de asignaturas
7	El usuario accede al icono de editar
8	El usuario selecciona "Eliminar"
9	¿confirma la eliminación?
10	El sistema realiza las operaciones necesarias y redirecciona a la página principal
Alternativas	
3.1	Si los datos son incorrectos el sistema envía un mensaje de error y redirecciona al paso 2
3.2	Si los datos son correctos, ir a paso 4
3.3	Si el usuario no está registrado el sistema manda un mensaje de error y redirecciona al paso 2
9.1	El usuario confirma la eliminación, ir a paso 9
9.2	El usuario no confirma la eliminación, ir a paso 6
10.1	Si todo finaliza correctamente, el sistema manda un mensaje de todo correcto
10.2	Si ocurre algún error en el proceso, el sistema manda un mensaje de error

Tabla 3.24: Especificación de caso de uso: Borrar asignatura

Especificación del Caso de Uso: Asignar laboratorios

Caso de Uso	Asignar laboratorios
Actor primario	Administrador
Sistema	Gestor de software
Participantes	Administrador - Sistema
Precondiciones	El usuario debe estar registrado en el sistema El usuario debe ser rol tipo "Administrador" Debe existir al menos una asignatura en el sistema Debe existir al menos un laboratorio en el sistema
Descripción	El usuario desea asignar un laboratorio a una asignatura
Operaciones básicas	
1	El usuario accede a la página web de la plataforma
2	Introduce sus datos de usuario: correo electrónico y contraseña
3	¿Los datos son correctos?
4	El sistema comprueba el rol del usuario y lo redirecciona a la página adecuada
5	El usuario accede al apartado "Asignaturas"
6	El sistema redirige a la lista de asignaturas
7	El usuario accede a "Asignar laboratorios"
8	El sistema realiza las operaciones necesarias y recarga los elementos
Alternativas	
3.1	Si los datos son incorrectos el sistema envía un mensaje de error y redirecciona al paso 2
3.2	Si los datos son correctos, ir a paso 4
3.3	Si el usuario no está registrado el sistema manda un mensaje de error y redirecciona al paso 2
7.1	El usuario asigna un laboratorio a una asignatura
7.2	El usuario desasigna un laboratorio de una asignatura

Tabla 3.25: Especificación de caso de uso: Asignar laboratorios

Especificación del Caso de Uso: Crear laboratorio	
Caso de Uso	Crear laboratorio
Actor primario	Administrador
Sistema	Gestor de software
Participantes	Administrador - Sistema
Precondiciones	El usuario debe estar registrado en el sistema El usuario debe ser rol tipo "Administrador"
Descripción	El usuario añade un nuevo laboratorio al sistema
Operaciones básicas	
1	El usuario accede a la página web de la plataforma
2	Introduce sus datos de usuario: correo electrónico y contraseña
3	¿Los datos son correctos?
4	El sistema comprueba el rol del usuario y lo redirecciona a la página adecuada
5	El usuario accede al apartado "Laboratorios"
6	El sistema redirige a la lista de laboratorio
7	El usuario accede a "Añadir laboratorio"
8	El usuario completa los datos
9	El sistema realiza las operaciones necesarias y redirige a la página principal
Alternativas	
3.1	Si los datos son incorrectos el sistema envía un mensaje de error y redirecciona al paso 2
3.2	Si los datos son correctos, ir a paso 4
3.3	Si el usuario no está registrado el sistema manda un mensaje de error y redirecciona al paso 2
8.1	Si los datos son correctos, ir a paso 9
8.2	Si los datos no son correctos, el sistema se mantiene en la misma página
9.1	Si todo sale correcto, el sistema manda un mensaje de todo correcto
9.2	Si se produce un error, el sistema manda un mensaje de error

Tabla 3.26: Especificación de caso de uso: Crear laboratorio

Especificación del Caso de Uso: Modificar laboratorio	
Caso de Uso	Modificar laboratorio
Actor primario	Administrador
Sistema	Gestor de software
Participantes	Administrador - Sistema
Precondiciones	El usuario debe estar registrado en el sistema El usuario debe ser rol tipo "Administrador" Debe existir al menos una asignatura en el sistema
Descripción	El usuario modifica un laboratorio
Operaciones básicas	
1	El usuario accede a la página web de la plataforma
2	Introduce sus datos de usuario: correo electrónico y contraseña
3	¿Los datos son correctos?
4	El sistema comprueba el rol del usuario y lo redirecciona a la página adecuada
5	El usuario accede al apartado "Laboratorios"
6	El sistema redirige a la lista de laboratorios
7	El usuario accede al menú del laboratorio
8	El usuario selecciona "Editar"
9	El usuario modifica los datos
10	El sistema realiza las operaciones necesarias y redirige a la pantalla inicial
Alternativas	
3.1	Si los datos son incorrectos el sistema envía un mensaje de error y redirecciona al paso 2
3.2	Si los datos son correctos, ir a paso 4
3.3	Si el usuario no está registrado el sistema manda un mensaje de error y redirecciona al paso 2
9.1	Si los datos son correctos, ir a paso 10
9.2	Si los datos no son correctos, el sistema se mantiene en la misma página
10.1	Si todo finaliza correctamente, el sistema manda un mensaje de todo correcto
10.2	Si se produce un error en el proceso, el sistema manda un mensaje de error

Tabla 3.27: Especificación de caso de uso: Modificar laboratorio

Especificación del Caso de Uso: Borrar laboratorio	
Caso de Uso	Borrar laboratorio
Actor primario	Administrador
Sistema	Gestor de software
Participantes	Administrador - Sistema
Precondiciones	El usuario debe estar registrado en el sistema El usuario debe ser rol tipo "Administrador" Debe existir al menos un laboratorio en el sistema
Descripción	El usuario elimina un laboratorio del sistema
Operaciones básicas	
1	El usuario accede a la página web de la plataforma
2	Introduce sus datos de usuario: correo electrónico y contraseña
3	¿Los datos son correctos?
4	El sistema comprueba el rol del usuario y lo redirecciona a la página adecuada
5	El usuario accede al apartado "Laboratorios"
6	El sistema redirige a la lista de laboratorios
7	El usuario accede al menú del laboratorio
8	El usuario selecciona "Editar"
9	El usuario selecciona "Eliminar"
10	¿confirma la eliminación?
11	El sistema realiza las operaciones necesarias y redirecciona a la página principal
Alternativas	
3.1	Si los datos son incorrectos el sistema envía un mensaje de error y redirecciona al paso 2
3.2	Si los datos son correctos, ir a paso 4
3.3	Si el usuario no está registrado el sistema manda un mensaje de error y redirecciona al paso 2
10.1	El usuario confirma la eliminación, ir a paso 10
10.2	El usuario no confirma la eliminación, ir a paso 6
11.1	Si todo finaliza correctamente, el sistema manda un mensaje de todo correcto
11.2	Si ocurre algún error en el proceso, el sistema manda un mensaje de error

Tabla 3.28: Especificación de caso de uso: Borrar laboratorio

Especificación del Caso de Uso: Consultar software instalado	
Caso de Uso	Consultar software instalado
Actor primario	Administrador
Sistema	Gestor de software
Participantes	Administrador - Sistema
Precondiciones	El usuario debe estar registrado en el sistema El usuario debe ser rol tipo “Administrador” Debe existir al menos un laboratorio en el sistema Debe existir al menos un “software” en el sistema
Descripción	El usuario consulta el software instalado en un laboratorio
Operaciones básicas	
1	El usuario accede a la página web de la plataforma
2	Introduce sus datos de usuario: correo electrónico y contraseña
3	¿Los datos son correctos?
4	El sistema comprueba el rol del usuario y lo redirecciona a la página adecuada
5	El usuario accede al apartado “Laboratorios”
6	El sistema redirige a la lista de laboratorios
7	El usuario accede al menú del laboratorio
8	El usuario selecciona “Software instalado”
9	El sistema carga la lista de software instalado
Alternativas	
3.1	Si los datos son incorrectos el sistema envía un mensaje de error y redirecciona al paso 2
3.2	Si los datos son correctos, ir a paso 4
3.3	Si el usuario no está registrado el sistema manda un mensaje de error y redirecciona al paso 2

Tabla 3.29: Especificación de caso de uso: Consultar software instalado

Especificación del Caso de Uso: Desasignar software	
Caso de Uso	Desasignar software
Actor primario	Administrador
Sistema	Gestor de software
Participantes	Administrador - Sistema
Precondiciones	El usuario debe estar registrado en el sistema El usuario debe ser rol tipo “Administrador” Debe existir al menos un laboratorio en el sistema Debe existir al menos un “software” en el sistema El laboratorio debe tener asignado al menos un “software”
Descripción	El usuario quiere desasignar un “software” de un laboratorio para que no aparezca como instalado
Operaciones básicas	
1	El usuario accede a la página web de la plataforma
2	Introduce sus datos de usuario: correo electrónico y contraseña
3	¿Los datos son correctos?
4	El sistema comprueba el rol del usuario y lo redirecciona a la página adecuada
5	El usuario accede al apartado “Laboratorios”
6	El sistema redirige a la lista de laboratorios
7	El usuario accede al menú del laboratorio
8	El usuario selecciona “Software instalado”
9	El sistema carga la lista de software instalado
10	El usuario selecciona el icono de eliminar del software
11	El sistema carga asincrónicamente los datos
Alternativas	
3.1	Si los datos son incorrectos el sistema envía un mensaje de error y redirecciona al paso 2
3.2	Si los datos son correctos, ir a paso 4
3.3	Si el usuario no está registrado el sistema manda un mensaje de error y redirecciona al paso 2

Tabla 3.30: Especificación de caso de uso: Desasignar software

Especificación del Caso de Uso: Consultar tareas pendientes	
Caso de Uso	Consultar tareas pendientes
Actor primario	Administrador
Sistema	Gestor de software
Participantes	Administrador - Sistema
Precondiciones	El usuario debe estar registrado en el sistema El usuario debe ser rol tipo "Administrador" Debe existir al menos una tarea pendiente en el sistema
Descripción	El usuario desea consultar las tareas pendientes
Operaciones básicas	
1	El usuario accede a la página web de la plataforma
2	Introduce sus datos de usuario: correo electrónico y contraseña
3	¿Los datos son correctos?
4	El sistema comprueba el rol del usuario y lo redirecciona a la lista de tareas pendientes
Alternativas	
3.1	Si los datos son incorrectos el sistema envía un mensaje de error y redirecciona al paso 2
3.2	Si los datos son correctos, ir a paso 4
3.3	Si el usuario no está registrado el sistema manda un mensaje de error y redirecciona al paso 2

Tabla 3.31: Especificación de caso de uso: Consultar tareas pendientes

Especificación del Caso de Uso: Consultar tareas finalizadas	
Caso de Uso	Consultar tareas finalizadas
Actor primario	Administrador
Sistema	Gestor de software
Participantes	Administrador - Sistema
Precondiciones	El usuario debe estar registrado en el sistema El usuario debe ser rol tipo "Administrador" Debe existir al menos una tarea finalizada en el sistema
Descripción	El usuario desea consultar las tareas finalizadas
Operaciones básicas	
1	El usuario accede a la página web de la plataforma
2	Introduce sus datos de usuario: correo electrónico y contraseña
3	¿Los datos son correctos?
4	El sistema comprueba el rol del usuario y lo redirecciona a la página principal adecuada
5	El usuario accede a "Tareas finalizadas"
Alternativas	
3.1	Si los datos son incorrectos el sistema envía un mensaje de error y redirecciona al paso 2
3.2	Si los datos son correctos, ir a paso 4
3.3	Si el usuario no está registrado el sistema manda un mensaje de error y redirecciona al paso 2

Tabla 3.32: Especificación de caso de uso: Consultar tareas finalizadas

Especificación del Caso de Uso: Finalizar tarea	
Caso de Uso	Finalizar tarea
Actor primario	Administrador
Sistema	Gestor de software
Participantes	Administrador - Sistema
Precondiciones	El usuario debe estar registrado en el sistema El usuario debe ser rol tipo “Administrador” Debe existir al menos una tarea pendiente en el sistema
Descripción	El usuario desea finalizar una tarea pendiente
Operaciones básicas	
1	El usuario accede a la página web de la plataforma
2	Introduce sus datos de usuario: correo electrónico y contraseña
3	¿Los datos son correctos?
4	El sistema comprueba el rol del usuario y lo redirecciona a la lista de tareas pendientes
5	El usuario accede a una de las tareas
6	El usuario accede a “Finalizar”
7	El usuario confirma la finalización de la tarea
8	El sistema realiza las operaciones necesarias y redirige a la página principal
Alternativas	
3.1	Si los datos son incorrectos el sistema envía un mensaje de error y redirecciona al paso 2
3.2	Si los datos son correctos, ir a paso 4
3.3	Si el usuario no está registrado el sistema manda un mensaje de error y redirecciona al paso 2
6.1	El usuario no agrega ningún comentario, ir a paso 7
6.2	El usuario agrega un comentario, ir a paso 7
8.1	Si todo finaliza correctamente, el sistema manda un mensaje de todo correcto
8.2	Si ocurre algún error en el proceso, el sistema manda un mensaje de error

Tabla 3.33: Especificación de caso de uso: Finalizar tarea

3.4. Diagramas de secuencia

“Un diagrama de secuencia muestra un conjunto de mensajes, dispuestos en una secuencia temporal. Cada rol en la secuencia se muestra como una línea de vida, es decir, una línea vertical que representa el rol durante cierto plazo de tiempo, con la interacción completa. Los mensajes se muestran como flechas entre las líneas de vida. [11].”

Los diagramas de secuencia representan las principales funcionalidades de nuestro sistema y nos muestra con mayor claridad la interacción entre los diferentes componentes. En la figura 3.10 podemos ver los distintos componentes que forman parte del diagrama:

- Actor, igual que en el diagrama de caso de uso es una idealización de una persona externa, interactúa con los diferentes roles.
- Rol, en este caso representa los diferentes controladores, vistas y modelos de nuestro proyecto.
- Mensaje/Respuesta, interacción entre el actor y los roles o entre roles.
- Fragmento alt, contiene una secuencia de mensajes alternativos. En cada caso sólo se ejecutará una de estas secuencias a la vez, cuando se cumpla la condición.

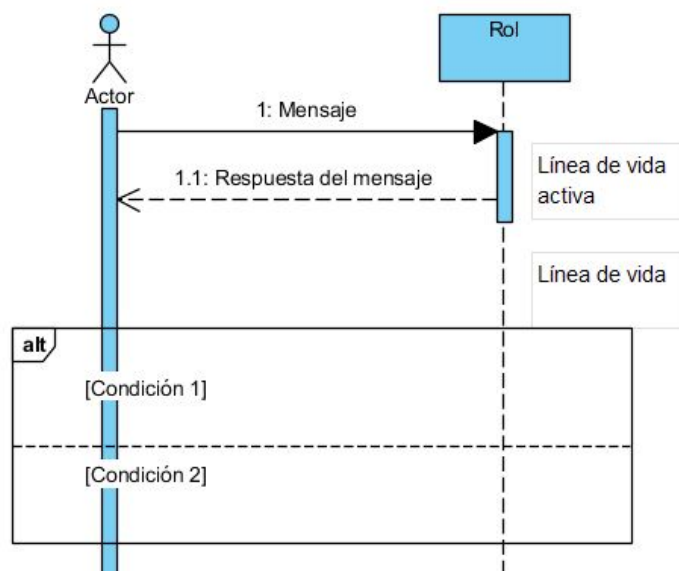


Figura 3.10: Esquema diagrama de secuencia

Diagrama de secuencia 1 - Iniciar Sesión

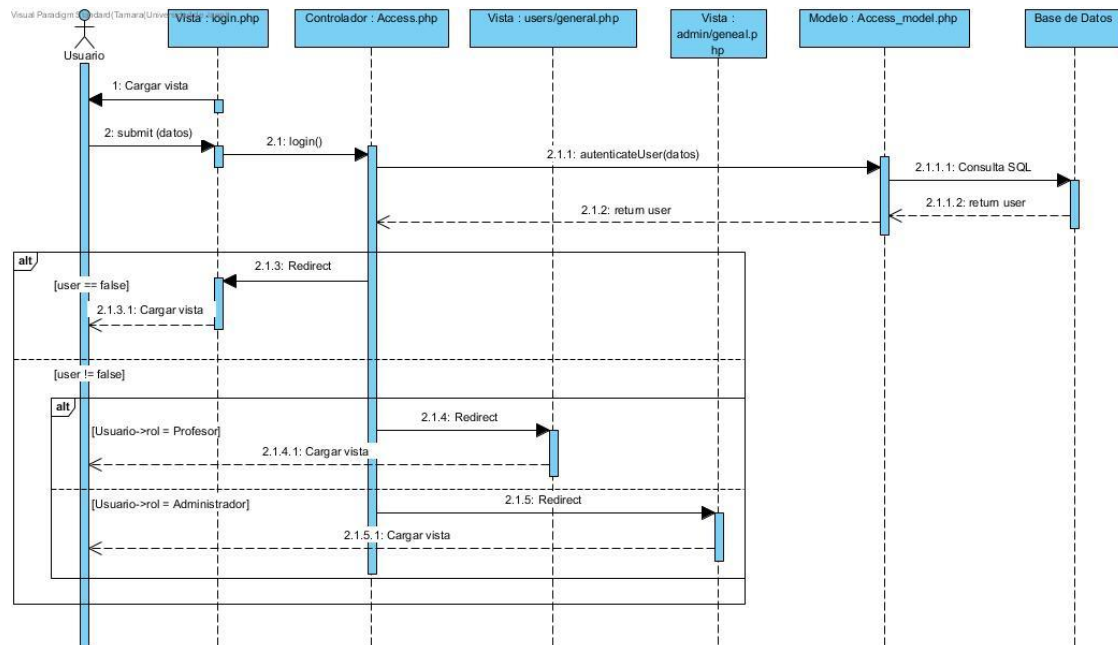


Diagrama de secuencia 2 - Editar perfil

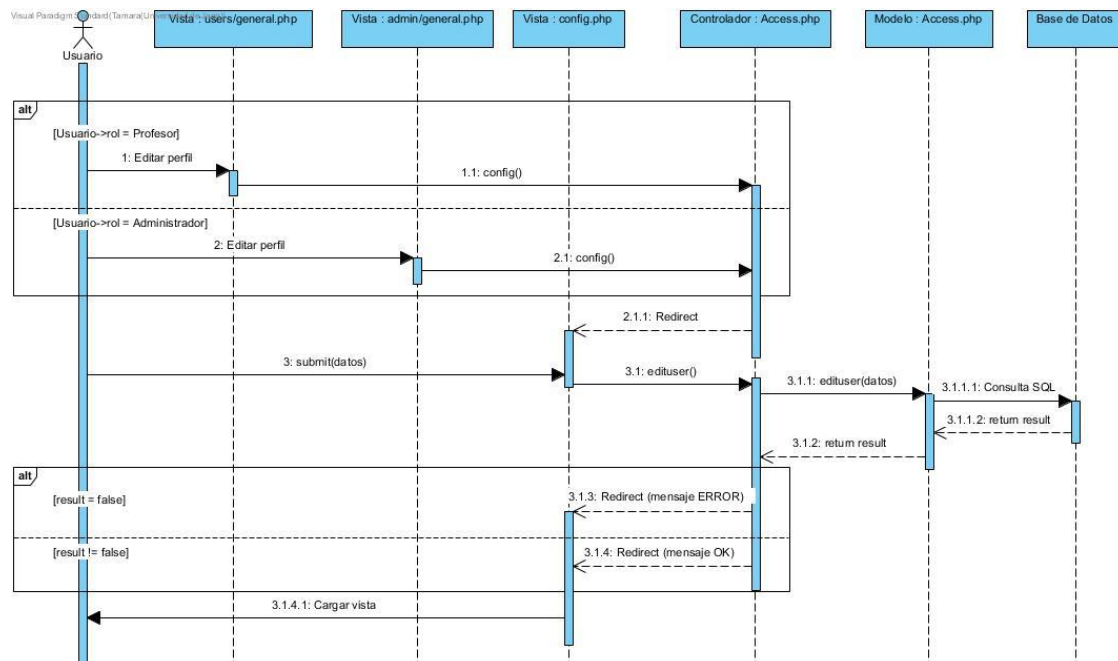


Diagrama de secuencia 3 - Crear software (Actor Profesor)

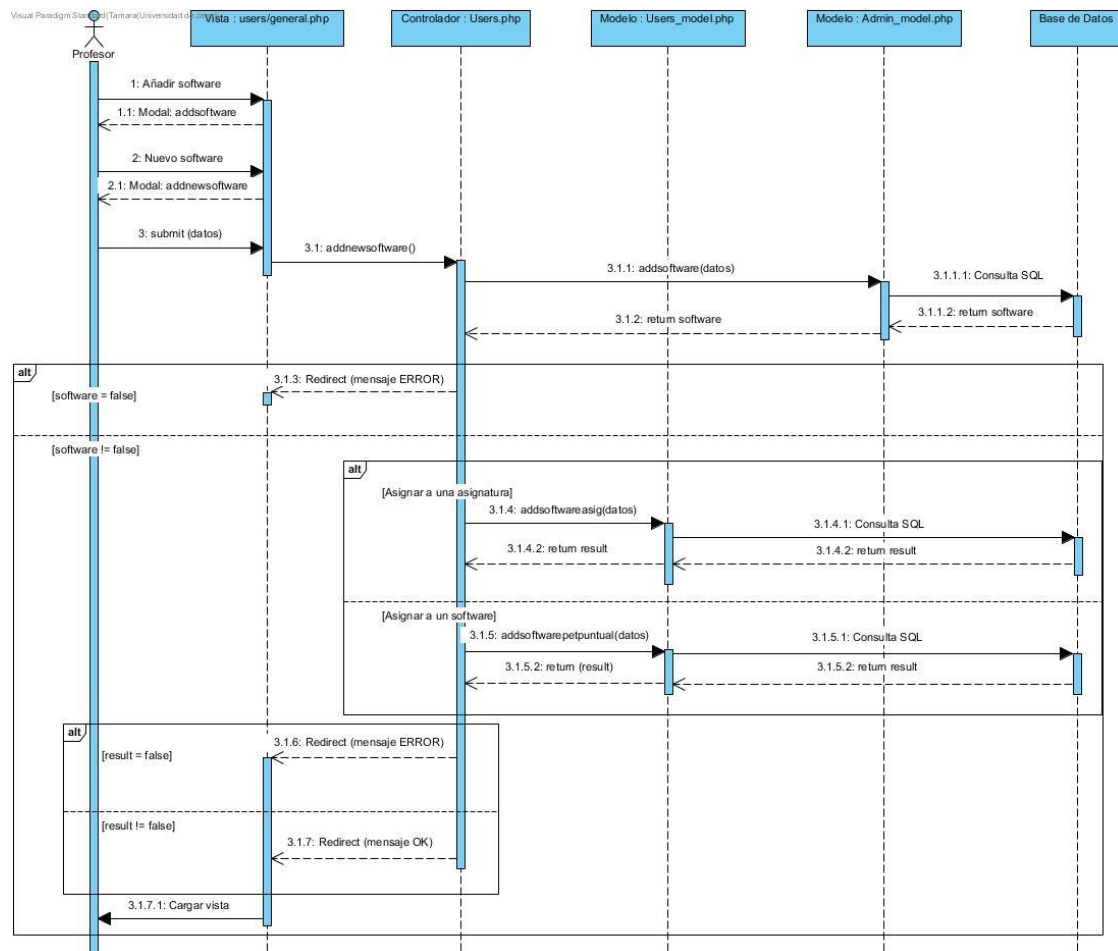


Diagrama de secuencia 4 - Crear software (Actor Administrador)

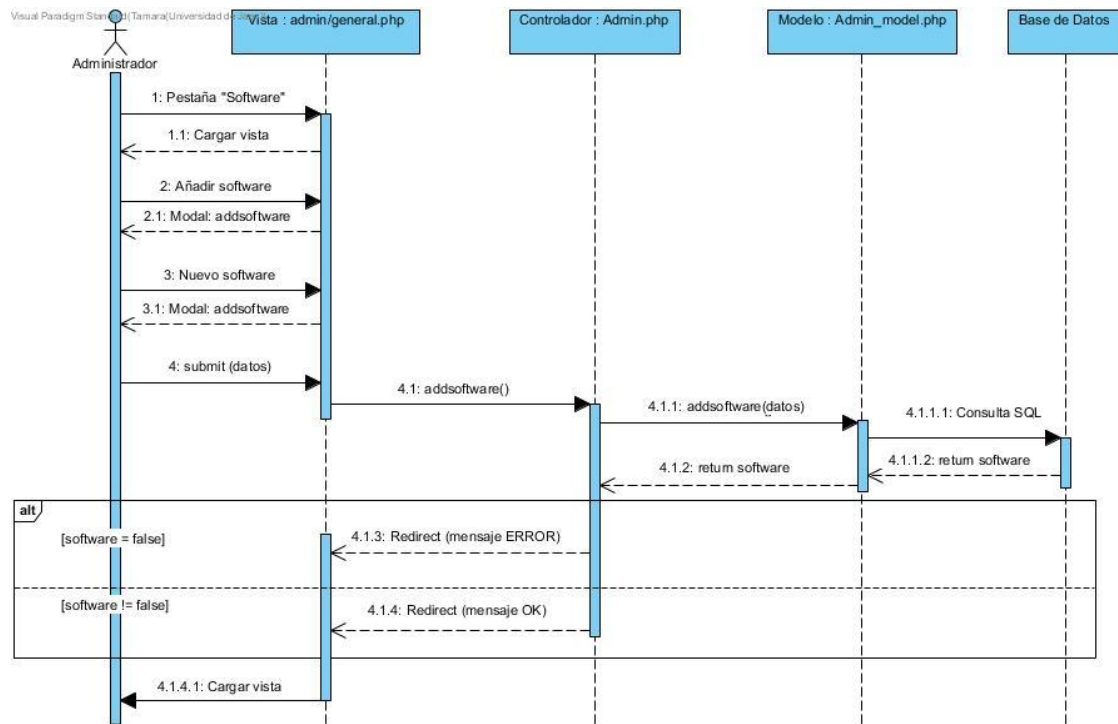


Diagrama de secuencia 5 - Añadir asignatura a lista

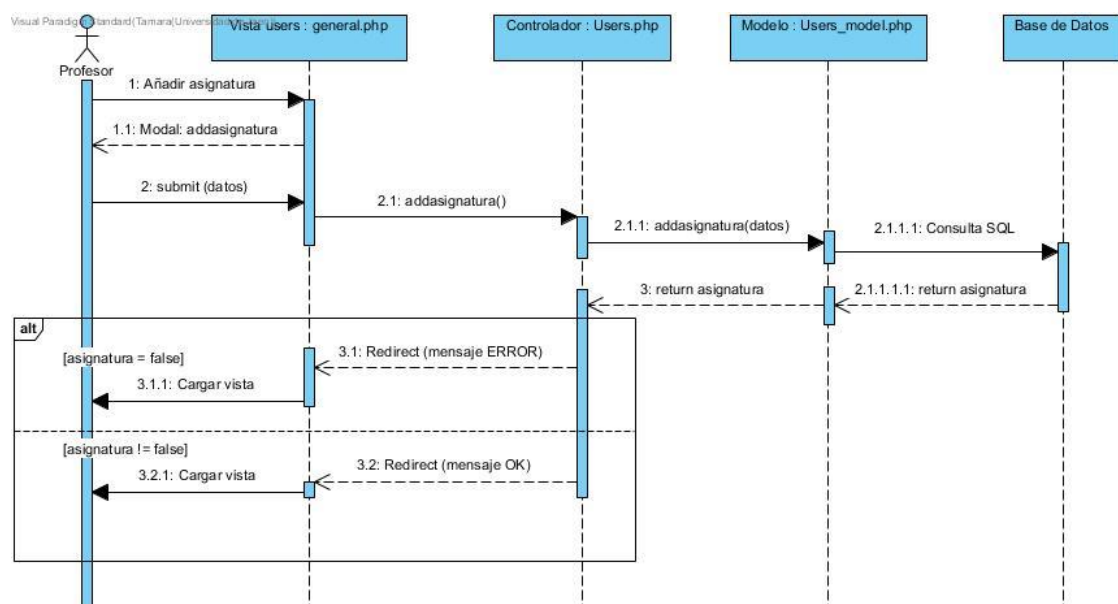


Diagrama de secuencia 6 - Borrar asignatura

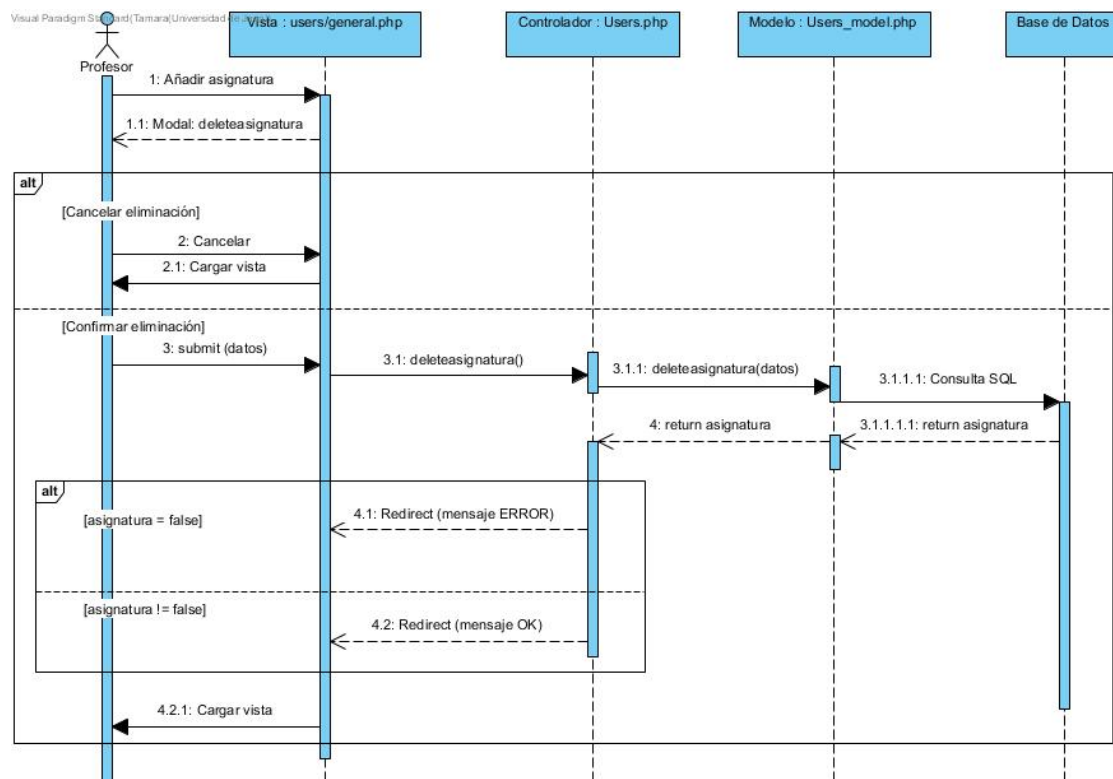


Diagrama de secuencia 7 - Crear petición puntual

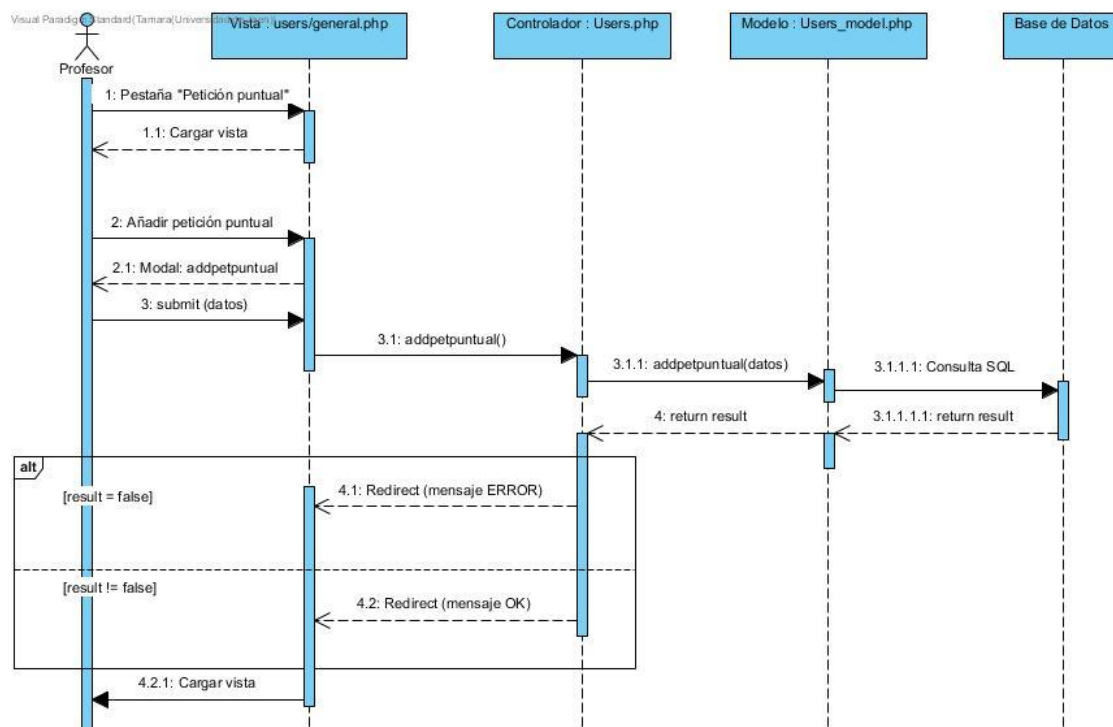


Diagrama de secuencia 8 - Modificar petición puntual

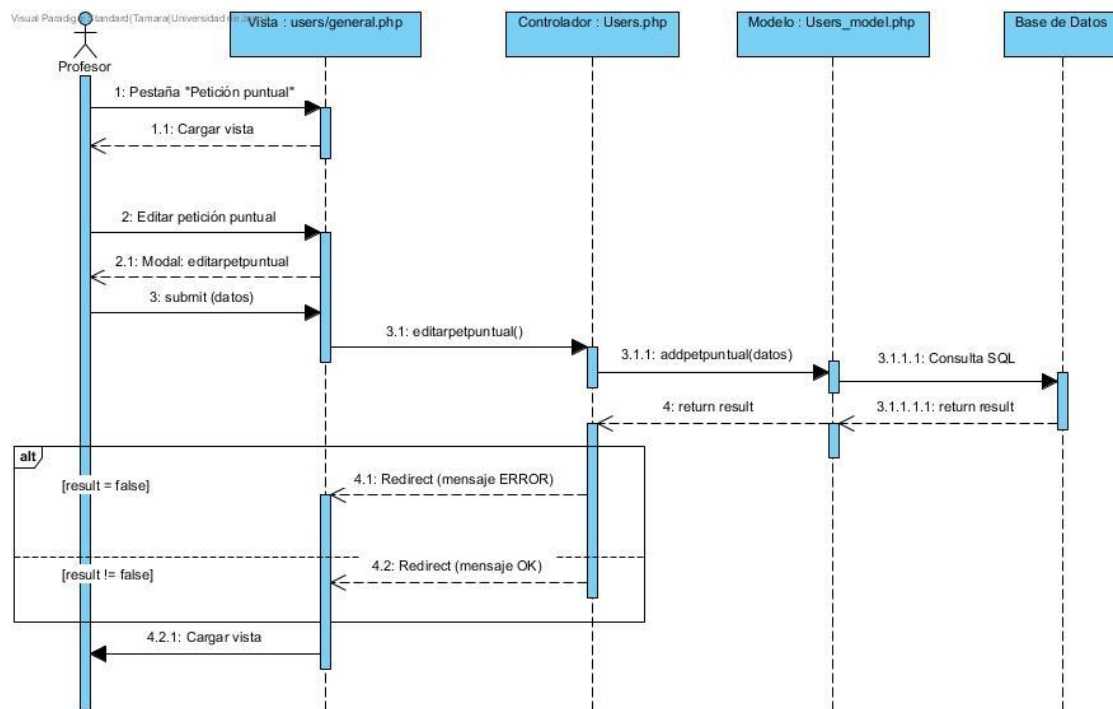


Diagrama de secuencia 9 - Borrar petición puntual

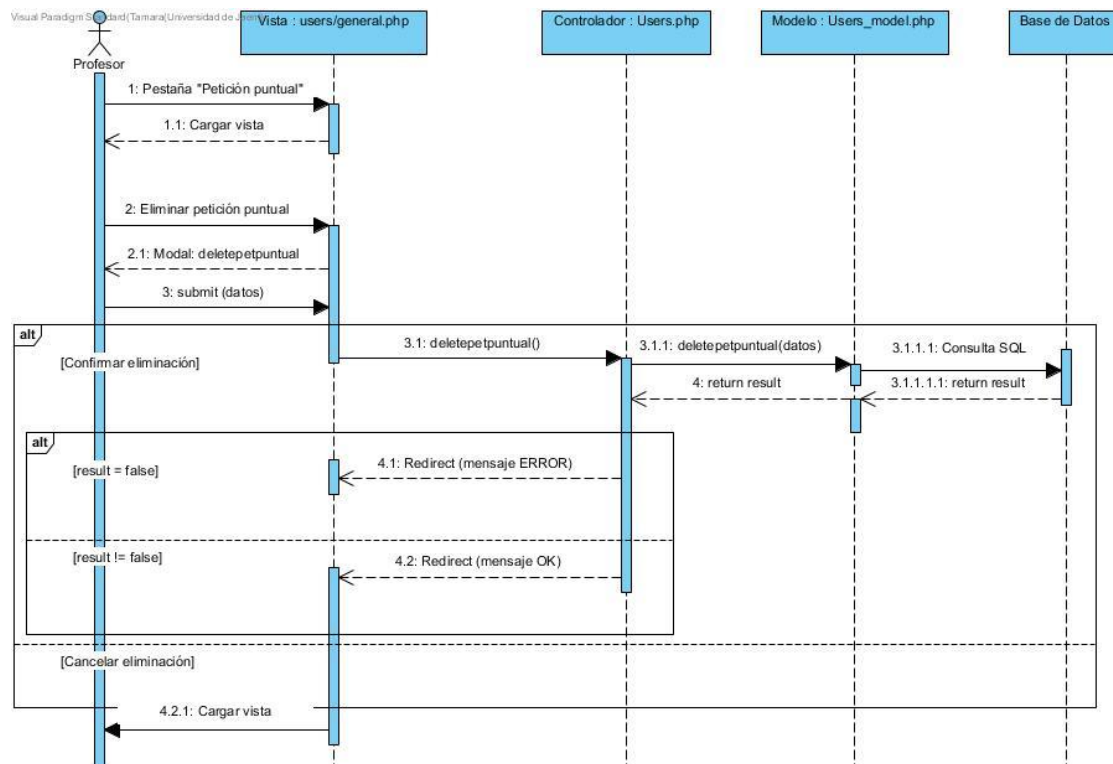


Diagrama de secuencia 10 - Modificar software

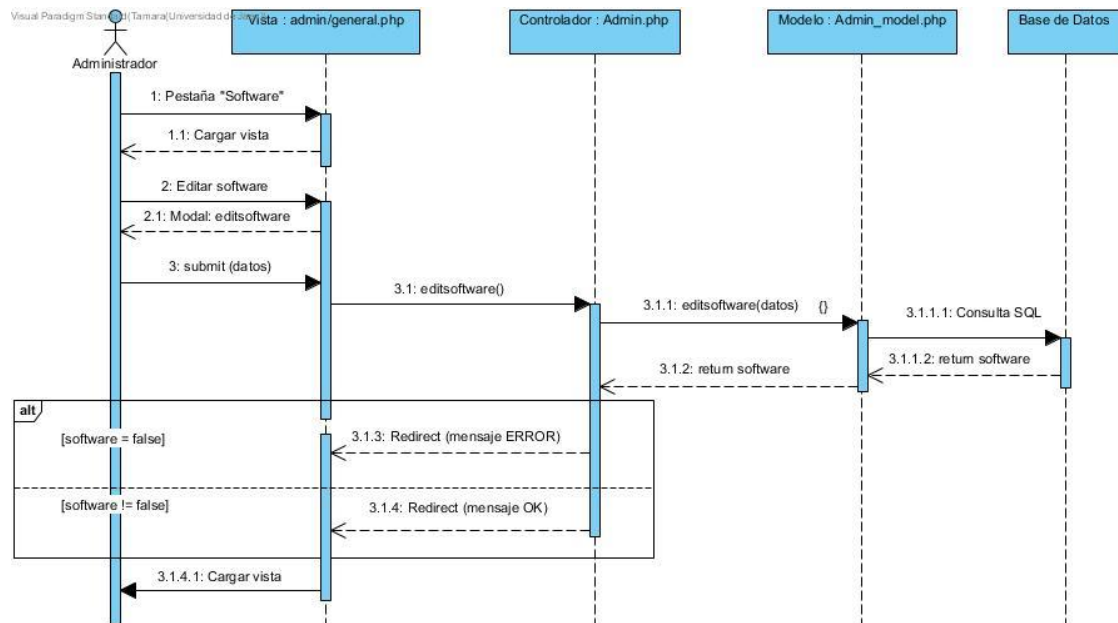


Diagrama de secuencia 11 - Añadir laboratorio

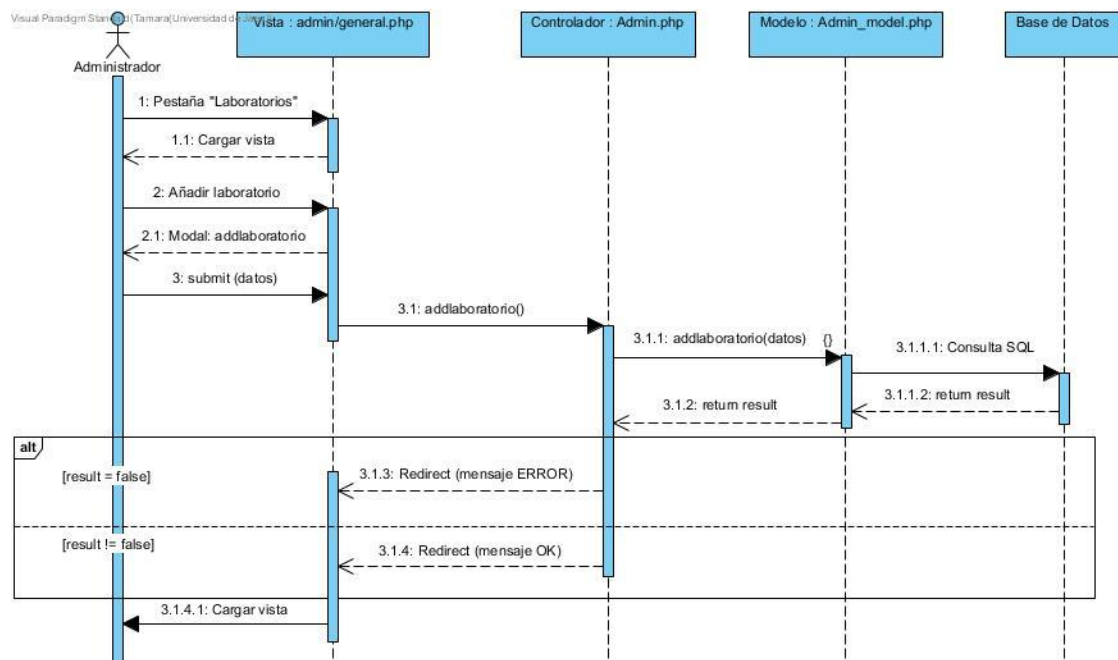


Diagrama de secuencia 12 - Modificar laboratorio

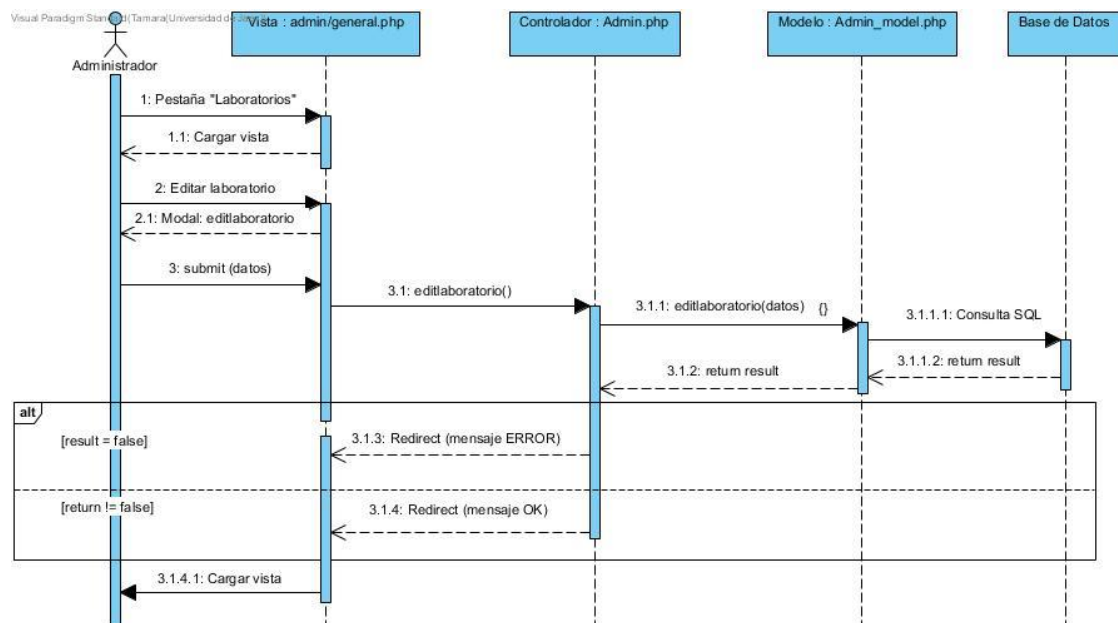


Diagrama de secuencia 13 - Borrar laboratorio

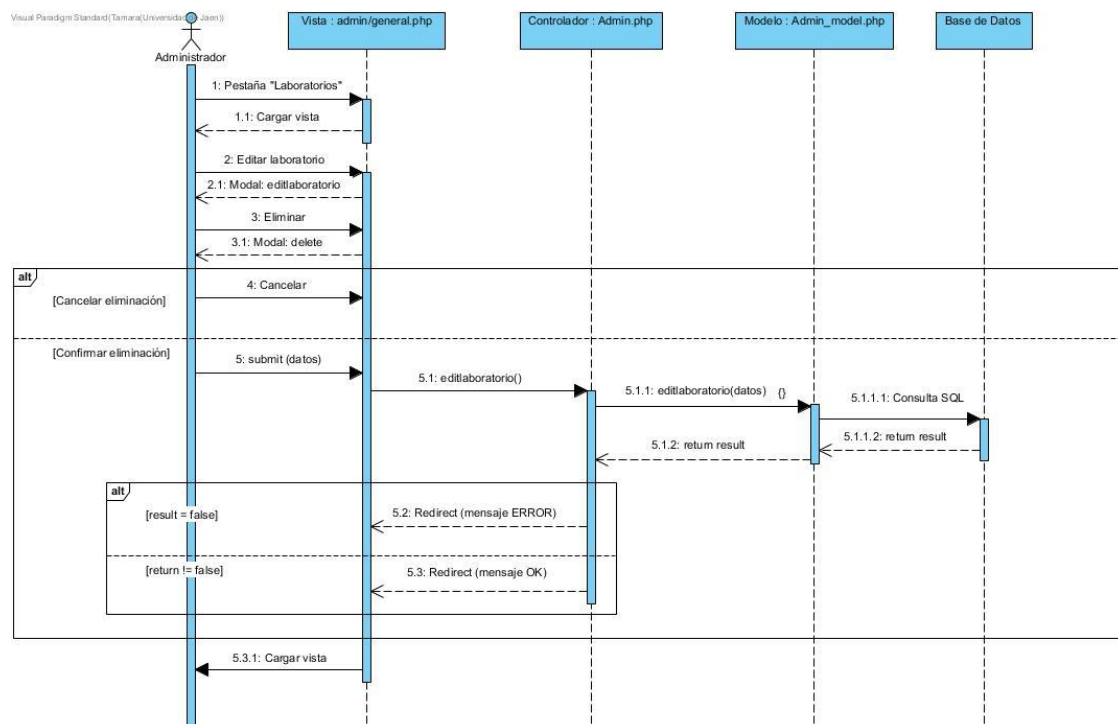


Diagrama de secuencia 14 - Crear asignatura

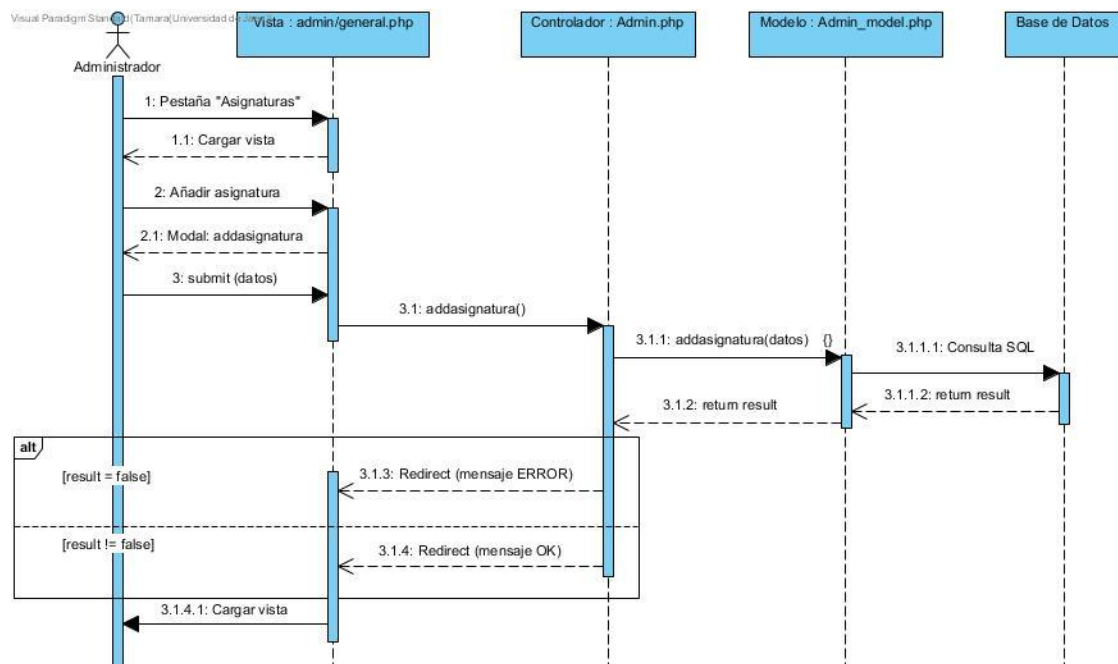


Diagrama de secuencia 15 - Borrar asignatura

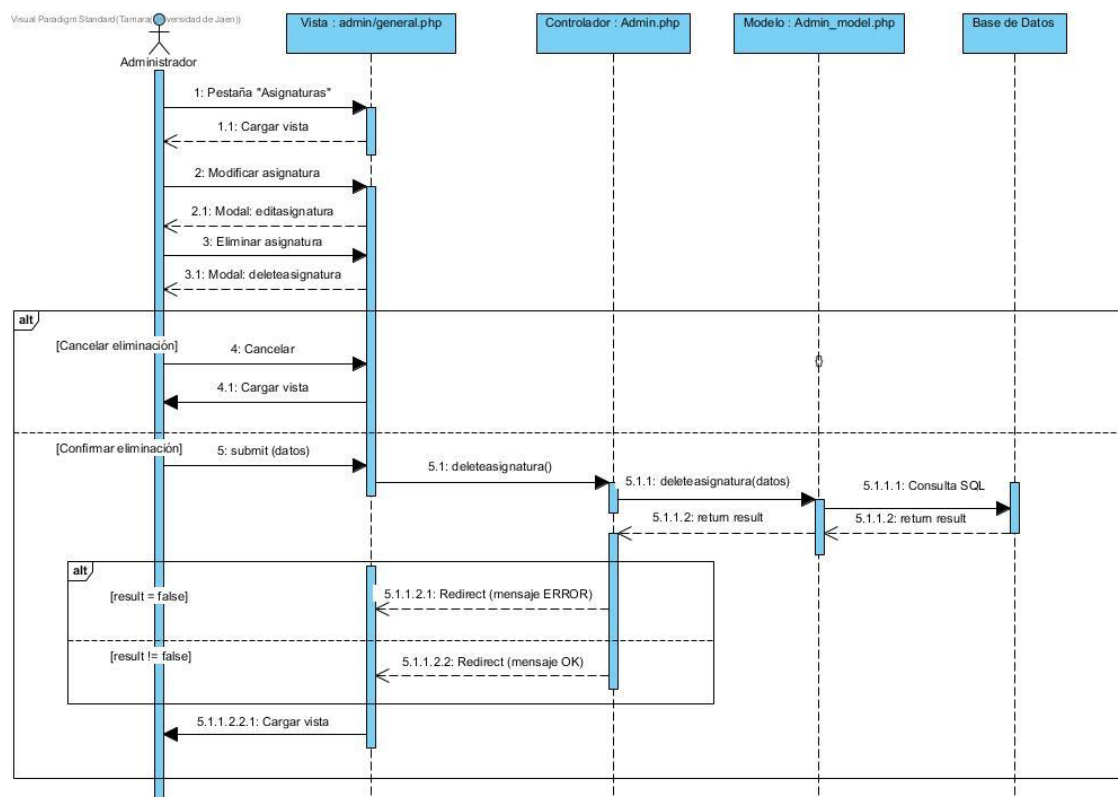


Diagrama de secuencia 16 - Asignar laboratorios

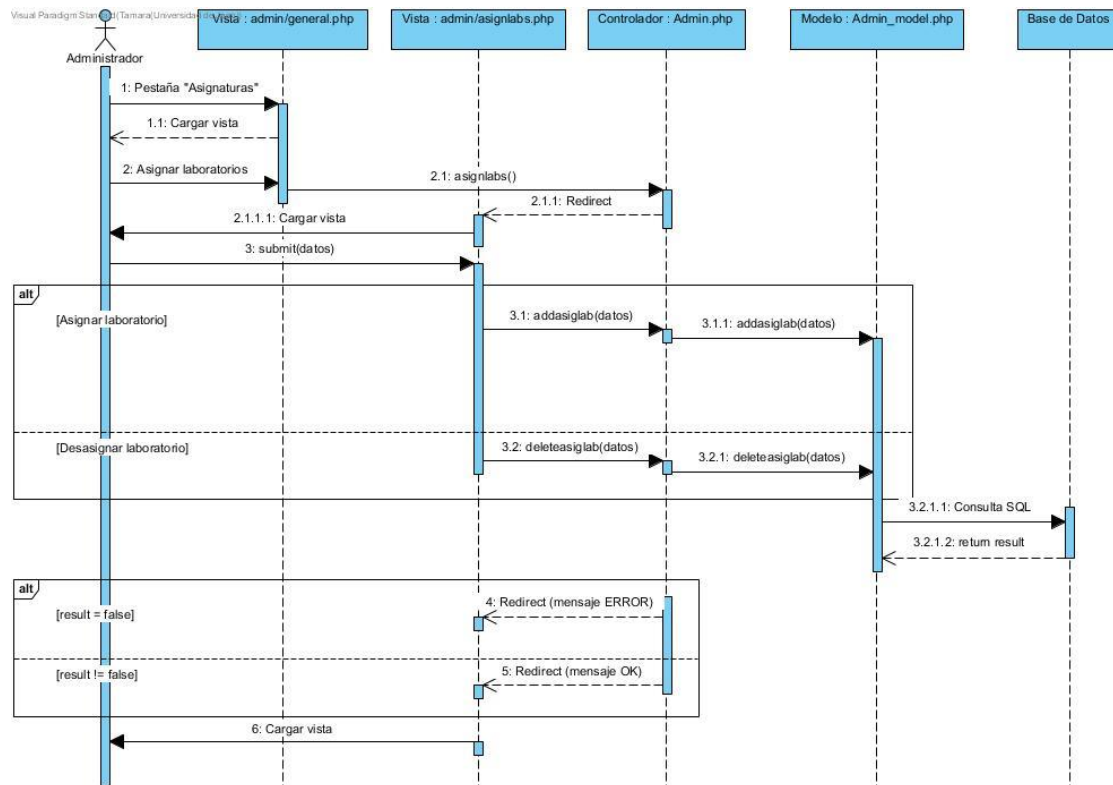


Diagrama de secuencia 17 - Añadir usuario

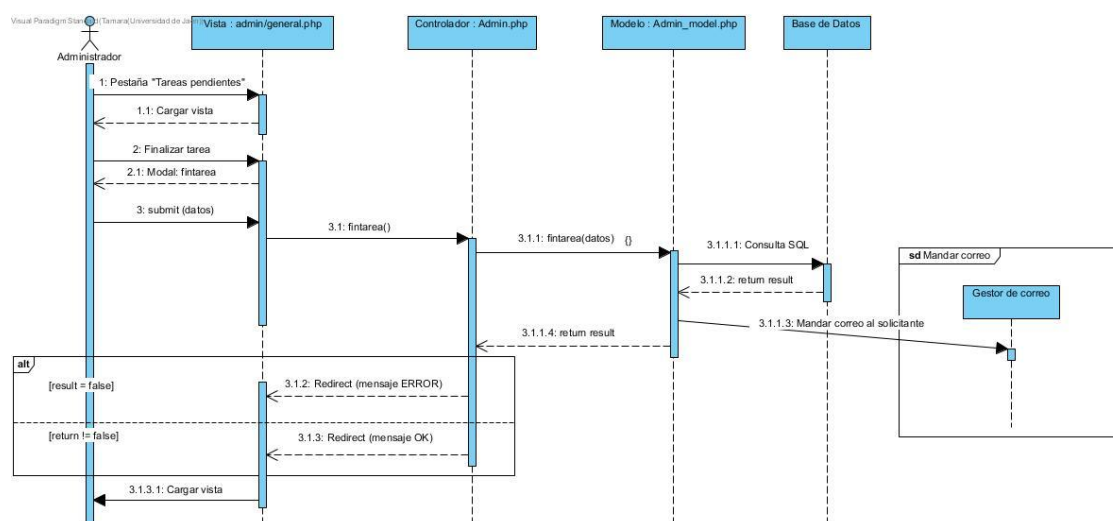
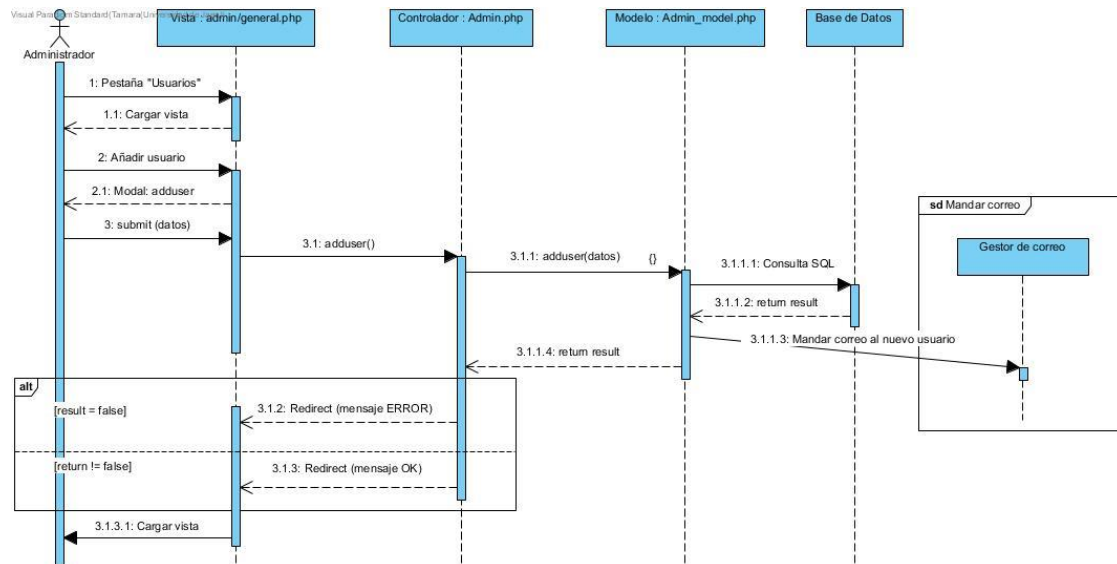


Diagrama de secuencia 18 - Finalizar tarea



Capítulo 4

Implementación

En este apartado hablaremos sobre todo lo relacionado con la fase de implementación del prototipo de la aplicación web. Esta implementación seguirá el diseño de la interfaz obtenida de los prototipos en papel (Capítulo 1, Análisis), aunque esta interfaz no será exactamente la misma del prototipo ya que se debe adecuar a los diferentes elementos disponibles para páginas web.

Comenzaremos viendo las diferentes tecnologías utilizadas, los requerimientos software y hardware que debe disponer el sistema para el desarrollo de una aplicación de este tipo. Posteriormente se detallarán diferentes fragmentos del código fuente utilizado durante el proceso, y por último, se describirán los diferentes casos de prueba para el sistema.

4.1. Arquitectura

Nuestro proyecto incluye dos arquitecturas:

- **Cliente-servidor:** arquitectura tradicional de una aplicación web.
- **Modelo-Vista-Controlador:** patrón aplicado en la arquitectura del servidor.

4.1.1. Arquitectura Cliente-Servidor

Esta arquitectura se compone por dos elementos, el cliente y el servidor. La parte del cliente hace referencia a la interfaz de usuario y es la encargada de realizar peticiones al servidor. Y la parte del servidor hace referencia a la lógica de negocio, se encarga de conectar con la base de datos y se mantiene a la espera de recibir peticiones y realizarlas.

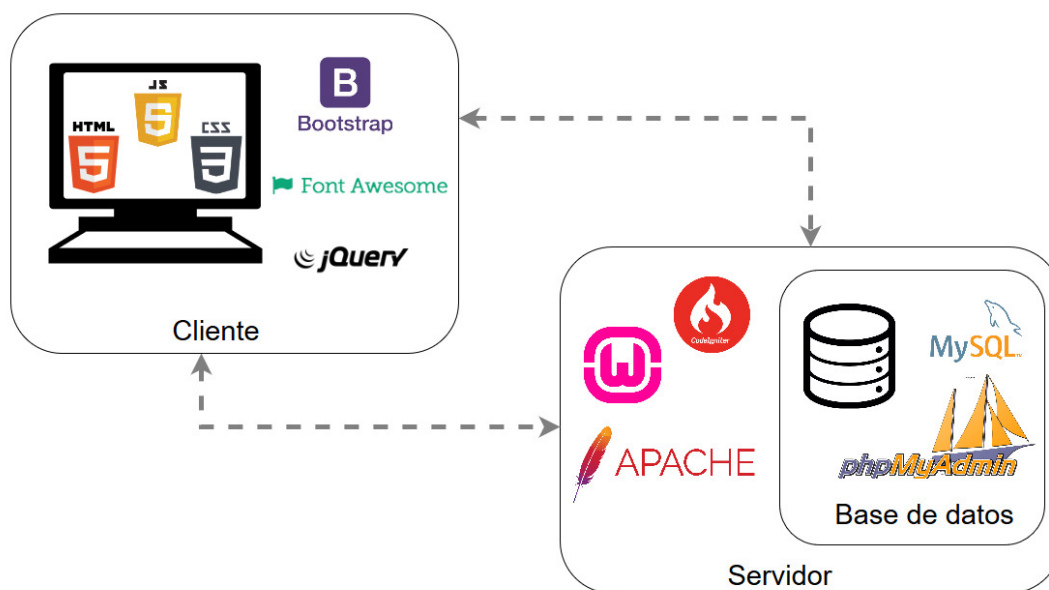


Figura 4.1: Arquitectura cliente-servidor

En la imagen 4.1 aparecen representadas las diferentes tecnologías utilizadas tanto en el cliente como en el servidor.

- Cliente: *HTML5, JavaScript, CSS, Bootstrap 3, jQuery y FontAwesome.*
- Servidor: *WAMP, CodeIgniter, Apache y para la base de datos phpMyAdmin con el motor MySQL.*

4.1.2. Arquitectura Modelo-Vista-Controlador

Este patrón arquitectónico ya ha sido explicado anteriormente en el capítulo 3, “Diseño”, apartado “Diagrama de clases”.

El uso del *framework CodeIgniter* nos permite implementar MVC fácilmente ya que está basado en dicho patrón. Como lenguaje de programación *CodeIgniter* utiliza PHP, tanto para los controladores como los modelos. Para las vistas se utiliza HTML y PHP. En el siguiente apartado se detallará mejor el uso de esta tecnología.

4.2. Detalles sobre la implementación

En este apartado se explicará el funcionamiento de las tecnologías utilizadas y se aportarán ejemplos de uso extraídos del código fuente de la aplicación.

4.2.1. WAMPServer

WAMPServer [6] es un desarrollo de entornos web, es decir, nos aporta diferentes funcionalidades para desarrollar una aplicación web. Algunas de estas funcionalidades son: Apache2, PHP, phpMyAdmin, etc.

Una vez instalado WAMPServer tendremos acceso a un servidor local dónde podemos empezar a implementar nuestra aplicación web.

Creación de la base de datos

Para el manejo de la base de datos usaremos una de las herramientas que nos aporta WAMPServer, phpMyAdmin [13]. Esta herramienta nos permite la gestión de una base de datos, creación de tablas, columnas, relaciones, usuarios, permisos, etc.

Para acceder a esta herramienta, una vez instalado e iniciado WAMPServer, nos dirigiremos a nuestro navegador y accederemos a “http://localhost/phpmyadmin”. Tras esto nos dirigirá a la pantalla que aparece en la figura 4.1, donde iniciaremos sesión con el usuario “root”.



Figura 4.2: Pantalla principal de phpMyAdmin

Para usar phpMyAdmin no necesitamos conocimientos avanzados de SQL ya que todo se puede realizar mediante interfaz, tanto crear tablas, añadir filas, etc.

4.2.2. CodeIgniter

Como ya se ha mencionado anteriormente vamos a utilizar el *framework* CodeIgniter [5]. El desarrollo con este *framework* es mucho más rápido una vez que se conoce, aunque hay que destacar que la curva de aprendizaje es empinada, por lo que los conocimientos son obtenidos rápidamente. Otra de las ventajas de CodeIgniter es que está basado en MVC, esta modularidad también nos dará más rapidez a la hora de implementar los diferentes componentes.

La instalación de CodeIgniter se realiza tras la de WAMPServer, ya que CodeIgniter nos provee de un conjunto de archivos que debemos de añadir al directorio “C:\wamp64\www” de WAMPServer. En estos archivos podemos encontrar diferentes carpetas “controllers”, “models”, y “views”, dónde colocaremos respectivamente nuestros controladores modelos y vistas.

Creación de un Controlador

Para explicar la creación de un controlador nos ayudaremos de un fragmento del código fuente de nuestra aplicación. En la figura 4.3 podemos ver cómo se define una clase de tipo controlador, dentro de esta clase podemos añadir las funciones correspondientes a dicho controlador.

```
1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  class Admin extends CI_Controller {
5
```

Figura 4.3: Creación del controlador Admin

Creación de un Modelo

La creación de un modelo es similar a la de un controlador, en este caso también nos ayudaremos de un fragmento del código para mostrarlo. En la figura 4.4 podemos ver como hemos definido el modelo “Admin_model”.

```
1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  class Admin_model extends CI_Model {
5
```

Figura 4.4: Creación del modelo Admin_model

Conexión con la base de datos

Ya hemos mencionado antes cómo crear nuestra base de datos gracias a phpMyAdmin. Ahora necesitamos conectar la base de datos con CodeIgniter para que nuestros modelos tengan acceso a esta. Para ello nos dirigiremos al siguiente directorio "C:\wamp64\www\application\config". En esta carpeta debemos dirigirnos al archivo "database.php", dicho archivo contiene la configuración de la base de datos para CodeIgniter, nombre de la base de datos, usuario, contraseña, etc. Debemos modificar dichos datos para que coincidan con la base de datos creada y nuestros modelos tendrán acceso a la base de datos.

Ahora bien, para que un modelo acceda a la base de datos podemos seguir varios métodos:

- Consulta SQL: dentro de una función de nuestro modelo podemos realizar una consulta SQL, en la figura 4.5 realizamos una consulta para obtener toda la información almacenada en la tabla "software".

```
$this->db->query("SELECT * FROM software");
```

Figura 4.5: Consulta SQL a base de datos

- Active Record [14]: se tratan de diferentes clases disponibles que nos permiten consultar, insertar, borrar o modificar la base de datos. La figura 4.6 representa la inserción de una nueva fila en la tabla "software_idioma" mediante active record.

```
$data = array(
    'idsoftware' => $result->idsoftware,
    'ididioma' => $ididioma
);

if($this->db->insert('software_idioma',$data)){
    return true;
}
```

Figura 4.6: Consulta SQL a base de datos

4.3. Interfaz de usuario

Como ya se ha mencionado antes la implementación de la interfaz de usuario se realiza mediante vistas. Estas vistas se realizan en lenguaje HTML, aunque también utilizaremos PHP y JavaScript.

Ya que los elementos puros de HTML como los botones, los campos para insertar texto, etc. tienen un estilo predeterminado que no sigue los estándares actuales haremos uso de Bootstrap y jQuery. Además, como en nuestro proyecto hemos incluido iconos hacemos uso de Font Awesome.

Bootstrap

Bootstrap [8] es una herramienta de desarrollo que nos permite usar diferentes componentes como botones, menús, campos de texto, etc. con un estilo más moderno.

Para añadir Bootstrap a nuestro proyecto primero debemos descargarnos los archivos necesarios en la página web de Bootstrap [8]. Una vez descargado debemos incluir los archivos en el directorio de nuestro proyecto, en este caso se han incluido en “C:\wamp64\www\third_party”. Una vez hecho esto, debemos incluir, en cada una de nuestras vistas que vayan a usar Bootstrap, dentro de la etiqueta “<head>” el acceso a estos archivos. En la figura 4.7 podemos ver cómo añadir dichos archivos.

```
<link rel="stylesheet" type="text/css" href="/thirdparty/css/bootstrap.min.css"/>
<script src="/thirdparty/js/bootstrap.min.js"></script>
```

Figura 4.7: Inclusión de Bootstrap en el proyecto

jQuery

jQuery [7] igual que Bootstrap nos permite utilizar diferentes componentes HTML e incluso añadir nuevos, con un estilo diferente a Bootstrap.

El proceso para añadir jQuery a nuestro proyecto es igual al de Bootstrap, debemos descargarlo en la página web de jQuery [7]. Posteriormente incluimos los archivos en el directorio “C:\wamp64\www\third_party”. Y por último, incluir el acceso a estos archivos. En la figura 4.8 podemos observar cómo hemos añadido estos archivos en nuestro proyecto.

```
<link rel="stylesheet" type="text/css" href="/thirdparty/css/jquery-ui.min.css"/>
<link rel="stylesheet" type="text/css" href="/thirdparty/css/jquery-ui.structure.min.css"/>
<link rel="stylesheet" type="text/css" href="/thirdparty/css/jquery-ui.theme.min.css"/>
<script src="/thirdparty/js/jquery-3.3.1.min.js"></script>
<script src="/thirdparty/js/jquery-ui.min.js"></script>
```

Figura 4.8: Inclusión de jQuery en el proyecto

Font Awesome

Font Awesome [9] es una plataforma open source que aporta un gran catálogo de iconos vectoriales y estilos css. Es compatible con la mayoría de dispositivos y navegadores ya que no requiere otras librerías para su uso y, al tratarse de iconos vectoriales, nos permite adaptarlos sin problemas de resolución.

Para añadir Font Awesome a nuestro proyecto podemos hacerlo directamente con un proceso similar a los anteriores. En cada vista HTML, en la etiqueta “<head>” debemos añadir el código que aparece en la página web “Getting Started on the Web” de Font Awesome [15].

El problema de este método es que si nuestra aplicación no tiene acceso a Internet no podremos ver los iconos, ya que nuestro proyecto no los incluye en sus directorios. Si eso es un problema también podemos seguir el método utilizado en Bootstrap y jQuery, descargando los archivos e incluyéndolos en nuestro proyecto.

Interfaz Responsive

Ya que hoy en día accedemos a páginas web desde una multitud de dispositivos (ordenadores, móviles, tabletas, etc.) necesitamos que nuestra interfaz sea “responsive”, es decir, que se adapte al dispositivo que estamos utilizando en ese momento.

En nuestro caso hemos usado el sistema de rejilla (*Grid System*) de Bootstrap [16]. Este sistema utiliza una serie de contenedores, filas y columnas para alinear el contenido a la pantalla.

En primer lugar debemos definir una fila **<div class=“row”>**, de esta forma, dentro de esta etiqueta el tamaño disponible se dividirá en doce columnas para ocupar todo el ancho. Para definir el ancho que queremos que ocupe nuestro elemento dentro de la fila debemos incluir la etiqueta:

- **col-xs-xx**, cuando la anchura del container es mayor a 576px.

- **col-md-xx**, cuando la anchura del container es mayor a 768px.
- **col-lg-xx**, cuando la anchura del container es mayor a 992px.
- **col-xl-xx**, cuando la anchura del container es mayor a 1200px.

Dónde “xx” es el tamaño que le queremos asignar, teniendo en cuenta que sumando todos los elementos dentro de una fila deben de sumar doce, que es el máximo de columnas en una fila. Podemos especificar una o varias etiquetas, dependiendo de los elementos o los dispositivos compatibles, ya que para cada tamaño de pantalla se modificará.

La figura 4.9 corresponde a un fragmento del código fuente de nuestro proyecto en el que hacemos uso del *Grid System*, en este caso, tenemos dos contenedores, el de tareas pendientes y el de filtros, podemos observar que la suma de las mismas etiquetas de los diferentes contenedores da doce.

```
<!-- Pestaña tareas pendientes -->
<div id="tab-pendientes" class="tab-pane fade in active">
  <div class="row">
    <!--Tareas pendientes-->
    <div class="col-sm-9" id="tareas"> ...
  </div>
  <!--Filtros-->
  <div class="col-sm-3"> ...
  </div>
</div>
```

Figura 4.9: Ejemplo de uso *Grid System*

Capítulo 5

Conclusiones

Para finalizar la memoria de este proyecto realizaremos una breve reflexión sobre el alcance de los objetivos inicialmente propuestos.

La aplicación se ha desarrollado empleando las diferentes metodologías y tecnologías que se han estudiado a lo largo de la carrera. Estas metodologías y tecnologías aplicadas han servido para realizar una aplicación de mayor calidad.

Se ha realizado un análisis de las necesidades de los usuarios mediante el uso del prototipado. Esta técnica nos permitió conseguir una aproximación más detallada de los requisitos funcionales para desarrollar el proyecto. El uso de esta técnica también nos ha ayudado a diseñar una interfaz de usuario adecuada y acorde a las exigencias de dichos usuarios.

El uso de los *frameworks* CodeIgniter y WAMP nos ha aportado grandes ventajas para la realización de este proyecto. No se ha necesitado consultar mucha bibliografía ya que estos *frameworks* ya han sido vistos en la asignatura de Tecnologías Basadas en Web, perteneciente al grado de Ingeniería Informática.

Por último, el desarrollo de este proyecto ha sido un gran reto, ya que nunca me había enfrentado al desarrollo de una aplicación a este nivel. Aunque también ha sido una gran motivación para adquirir mayores conocimientos en el ámbito del desarrollo web.

5.1. Mejoras y trabajos futuros

Aunque se han alcanzado los objetivos principales con la realización del proyecto sí es verdad que se podría dar un paso más. Ya en la Universidad de Jaén existe un método para realizar una gestión del software de los laboratorios en el Dpto. de Informática mediante un sistema de “tickets”. Este proyecto podría avanzar aún más uniéndose con este método o incluso reemplazarlo si se implementaran diferentes funcionalidades más específicas.

Otro aspecto que se podría mejorar aún más es la gestión del software que almacenamos en la base de datos ya que, en un principio se quería hacer un control más exhaustivo para evitar introducir datos ya existentes en el sistema o datos erróneos. Pero finalmente no se ha llegado a este punto.

Bibliografía

- [1] J.J. Cañas M. Gea A.N. Gil J. Lorés A.B. Martínez M. Ortega P. Valero M.Velez. I. Abascal, I. Aedo. *La interacción persona-ordenador*.
- [2] Tablas salariales, boe. https://www.boe.es/diario_boe/txt.php?id=BOE-A-2017-4475. Accedido 26-08-2018.
- [3] Web de la herramienta pencil project. <https://pencil.evolus.vn/>. Accedido 26-08-2018.
- [4] K. Pohl. *Requirements engineering: Fundamentals, principles, and techniques*.
- [5] Web de codeigniter. <https://www.codeigniter.com/>. Accedido 26-08-2018.
- [6] Web de wampserver. <http://www.wampserver.com/en/>. Accedido 26-08-2018.
- [7] Web de jquery. <https://jquery.com/>. Accedido 26-08-2018.
- [8] Web de bootstrap. <https://getbootstrap.com/>. Accedido 26-08-2018.
- [9] Web de font awesome. <https://fontawesome.com/>. Accedido 26-08-2018.
- [10] Sommerville. *Software engineering (9th edition)*. Addison-Wesley Educational Publishers.
- [11] G. Booch y J. Rumbaugh. *El lenguaje unificado de modelado*.
- [12] A. Pérez. *Estructuración y Especificación de Casos de Uso*.
- [13] Web de phpmyadmin. <https://www.phpmyadmin.net/>. Accedido 26-08-2018.
- [14] Documentación active record (codeigniter). https://www.codeigniter.com/userguide2/database/active_record.html. Accedido 26-08-2018.

- [15] Getting started on the web, font awesome. <https://fontawesome.com/how-to-use/on-the-web/setup/getting-started?using=web-fonts-with-css>. Accedido 26-08-2018.
- [16] Grid system, bootstrap. <https://getbootstrap.com/docs/4.0/layout/grid/>. Accedido 26-08-2018.
- [17] Manual de referencia mysql. <https://dev.mysql.com/doc/refman/8.0/en/>. Accedido 26-08-2018.
- [18] Steven Holzner. *PHP : manual de referencia*.
- [19] Dan Mantyla Ved Antani, Simon Timms. *JavaScript : functional programming for JavaScript developers*.
- [20] David Flanagan. *JavaScript [Recurso electrónico] : the definitive guide*.

Anexo I

Manual de instalación

Para acceder a la aplicación suministrada necesitamos disponer de diferentes herramientas. En nuestro caso necesitamos disponer un servidor local para acceder a la aplicación

Debemos disponer de diferentes *frameworks* como WAMPServer, a continuación se explicará la instalación de estos *frameworks*.

Comenzamos con la instalación de WAMPServer, para ello primero debemos descargarnos el instalador desde la página web de WAMPServer [6]. Tras instalarlo en los archivos proporcionados podemos encontrar el archivo comprimido “codigofuente.zip”, que como su nombre indica contiene el código fuente del proyecto. Debemos descomprimir dicho archivo en el directorio “C:\wamp64\www”.

A continuación, debemos de importar la base de datos proporcionada. En los archivos proporcionados podemos encontrar el archivo “bbdd.sql”, que contiene la base de datos. Para importarla debemos ir al navegador y escribir “localhost/phpmyadmin”, y nos dirigirá a la página inicial de phpMyAdmin, que como ya se mencionó en el capítulo “Implementación” es la herramienta que hemos usado para gestionar la base de datos.

En la figura I.1 se muestra la página inicial de phpMyAdmin, para entrar debemos de usar el usuario “root”. Una vez que iniciamos sesión en phpMyAdmin, nos dirigimos en el menú izquierdo a “Nueva”, “Importar”, seleccionamos el archivo de la base de datos antes mencionado “bbdd.sql” y “Continuar”. En este punto ya nos debería aparecer en el menú izquierdo la base de datos “bbdd”.



Bienvenido a phpMyAdmin

Idioma - *Language*

Español - Spanish

Iniciar sesión ⓘ

Usuario: root

Contraseña:

Elección del servidor: MySQL

Continuar

Figura I.1: Página inicial phpmyadmin

Una vez realizados estos pasos podemos entrar en el sistema, para ello debemos iniciar WAMPServer y en el navegador escribiremos “localhost”. Tras esto el navegador nos debería dirigir a la página principal de la aplicación, que en este caso es la página de inicio de sesión.

Anexo II

Manual de Usuario

Este manual de usuario le ayudará a gestionar de manera ágil y sencilla nuestra aplicación web para la gestión del software de los laboratorios del Dpto. de Informática.

Entrada a la aplicación

The image shows a login interface for a system titled "Gestor de Software UJA". The interface is contained within a white rectangular box with a light gray border, set against a larger light gray background. At the top of the box, the title "Gestor de Software UJA" is displayed in a dark gray font. Below the title, the text "Inicia sesión" is centered. Underneath, there are two input fields: the first is labeled "Correo electrónico" and the second is labeled "Contraseña". Both fields have a light gray border and a small icon on the right side. Below these fields is a blue button with the text "Iniciar sesión" in white, centered on the button.

Figura II.1: Página de inicio de sesión

La primera vez que accedemos nos encontramos con la página de inicio de sesión. Existen dos tipos de usuarios en el sistema que mostrarán diferentes vistas, para ello se proporcionan dos usuarios de ejemplo para acceder a la aplicación.

Usuario Profesor:

- Correo electrónico: profesor@correo.com
- Contraseña: profesor

Usuario Administrador:

- Correo electrónico: administrador@correo.com
- Contraseña: administrador

Vistas comunes

Para los dos tipos de usuarios hay una parte en común, la barra de navegación o “navbar”, dónde podemos encontrar el nombre de la aplicación a un lado y al otro el nombre del usuario. También dispone de un menú con las opciones “Editar perfil” y “Cerrar sesión”. El nombre de la aplicación a su vez es un enlace a la página principal del usuario, que dependerá del rol que tenga asignado.



Figura II.2: Página de inicio de sesión

Editar perfil

En este apartado tenemos dos opciones: editar los datos personales (nombre, apellidos o teléfono) o modificar la contraseña.

The screenshot shows the 'Gestor de Software UJA' interface. At the top, there is a header bar with the text 'Gestor de Software UJA' on the left and a dropdown menu 'Apellidos, Nombre' on the right. Below the header, there are two tabs: 'Datos personales' (active) and 'Cambiar contraseña'. The 'Datos personales' tab contains four input fields: 'Nombre', 'Apellidos', 'Correo' (with the value 'profesor@correo.com'), and 'Teléfono'. Below these fields is a blue 'Guardar' button.

Figura II.3: Modificar los datos personales

The screenshot shows the 'Gestor de Software UJA' interface. At the top, there is a header bar with the text 'Gestor de Software UJA' on the left and a dropdown menu 'Apellidos, Nombre' on the right. Below the header, there are two tabs: 'Datos personales' and 'Cambiar contraseña' (active). The 'Cambiar contraseña' tab contains two input fields: 'Nueva contraseña' and 'Repetir contraseña'. Below these fields is a blue 'Guardar' button.

Figura II.4: Cambiar contraseña

Vista Profesor

The screenshot shows the 'Gestor de Software UJA' interface for the 'Profesor' user. At the top, there is a header bar with the text 'Gestor de Software UJA' on the left and a dropdown menu 'Apellidos, Nombre' on the right. Below the header, there are three tabs: 'Asignaturas' (active), 'Peticiónes', and 'Laboratorios'. Below the tabs, there is a blue 'Añadir Asignatura' button. At the bottom, there is a light blue message box with an information icon and the text 'Todavía no has añadido asignaturas.'

Figura II.5: Página principal para el usuario “Profesor”

Como podemos ver el usuario tipo “Profesor” tiene un menú en el que puede elegir entre “Asignaturas”, “Peticiónes puntuales” y “Laboratorios”.

Añadir asignatura

El proceso para añadir una asignatura es muy simple, tan solo ha de seleccionar una de la asignatura de la lista.

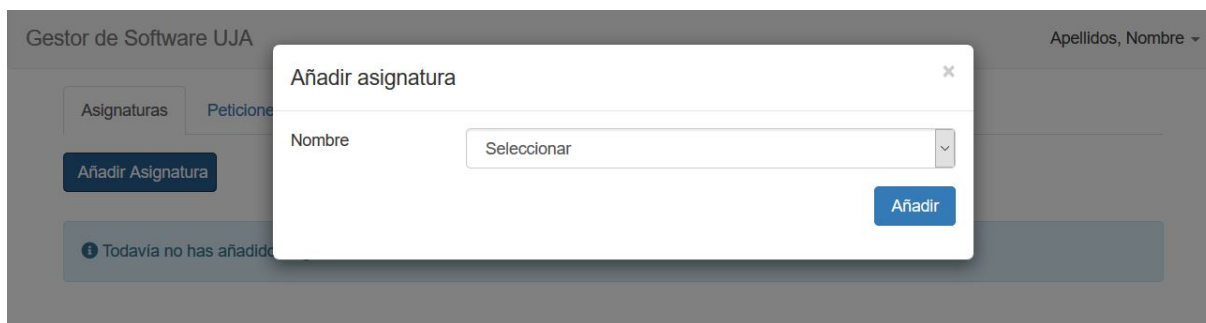


Figura II.6: Añadir una asignatura a la lista del usuario

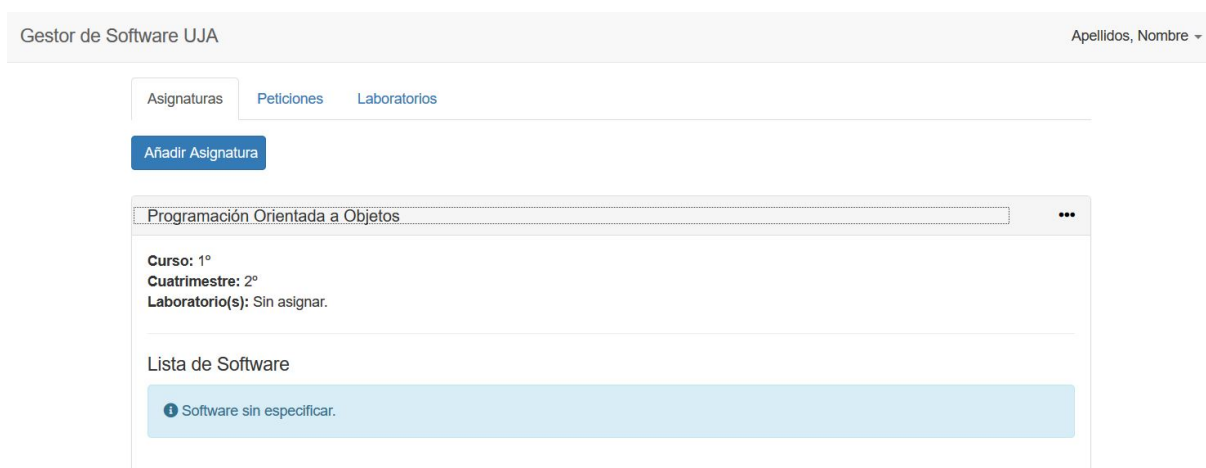


Figura II.7: Datos de la asignatura

Tras añadir una nueva asignatura se nos mostrará como el ejemplo de la Figura II.7. Dónde aparecen los diversos datos de la asignatura, nombre, curso, cuatrimestre y laboratorios. Y, bajo estos, aparecerá la lista del software solicitado para la asignatura.

Gestionar asignatura

Para cada asignatura disponemos de un menú con las diferentes acciones que podemos realizar, “Enviar ticket”, “Eliminar” y “Añadir software”.

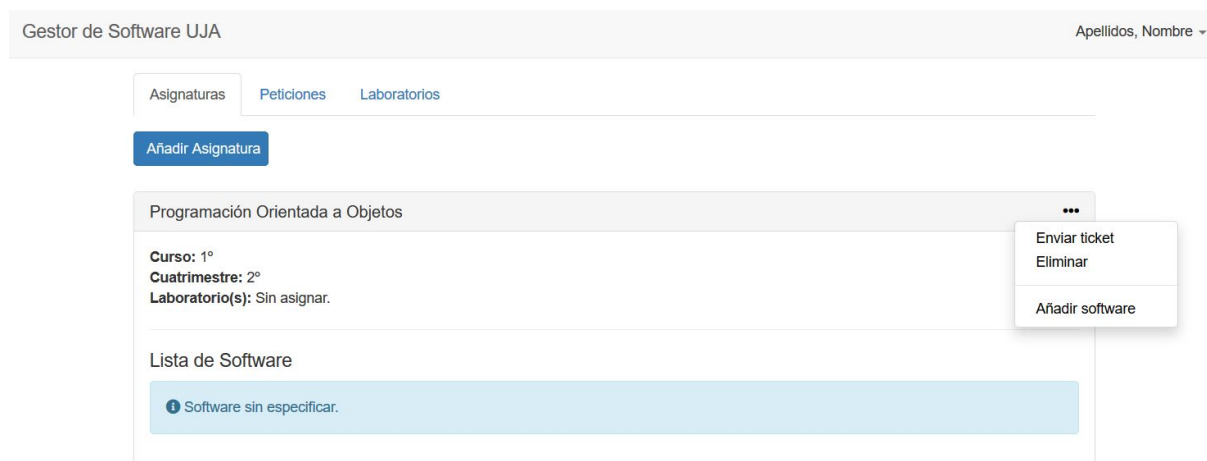


Figura II.8: Menú de una asignatura

Añadir software a una asignatura

Podemos añadir software a una asignatura de dos formas diferentes: seleccionando un software o añadiendo uno no existente.

Si añadimos un software no existente en el sistema, éste se creará y se asignará a la asignatura.

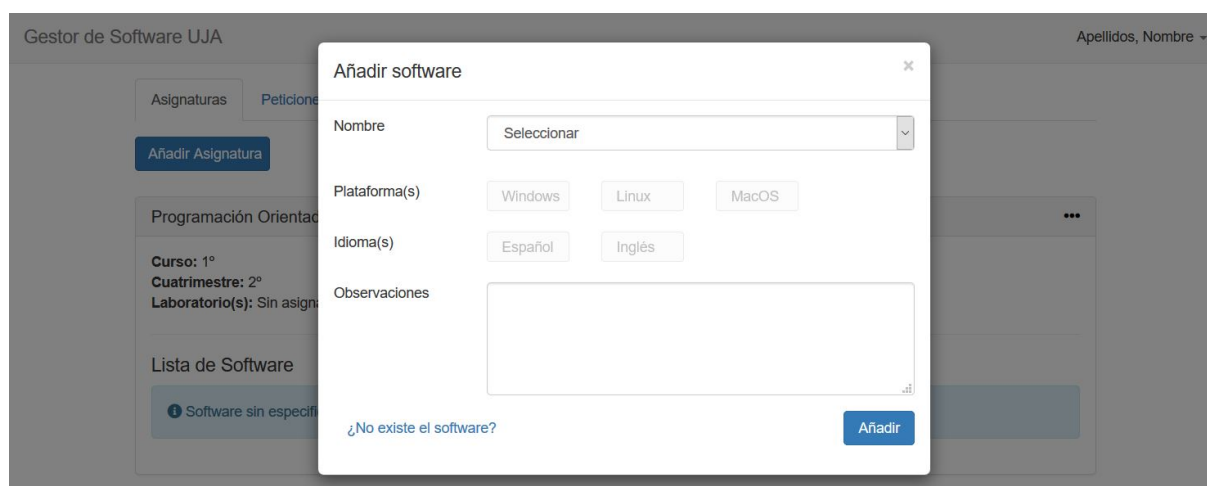


Figura II.9: Añadir software existente a una asignatura

En la Figura II.11 se muestra un ejemplo de una asignatura tras añadirle un software.

Figura II.10: Añadir software nuevo a una asignatura

Nombre	Versión	Plataforma(s)	Idioma	Licencia
Visual Studio Community	2015	Windows	Español	Gratuito

Figura II.11: Asignatura con software añadido

Gestionar Peticiones puntuales

Otra de las funcionalidades que puede realizar un usuario tipo “Profesor” es gestionar “Peticiones puntuales”. Estas peticiones permiten a los usuarios solicitar software que no esté asociado a ninguna asignatura, como cursos, exposiciones, etc.

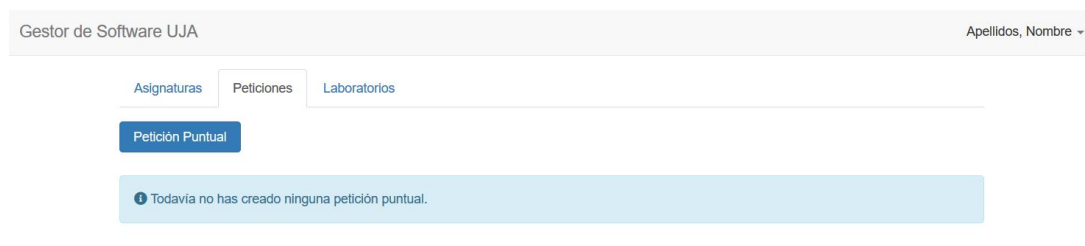


Figura II.12: Pestaña peticiones puntuales

Añadir petición puntual

Para añadir una petición puntual se deben de completar los datos y asignar los laboratorios dónde se va a impartir, aunque esto último no es obligatorio y puede ser modificado posteriormente.

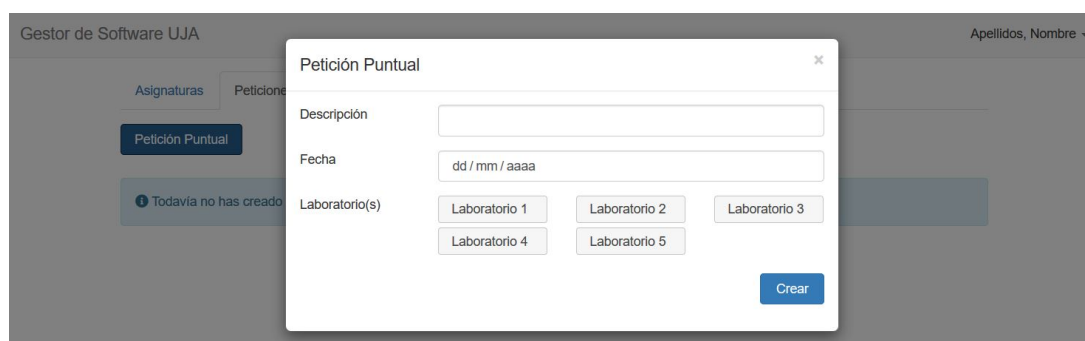


Figura II.13: Añadir petición puntual

Una vez añadida la petición nos aparecerá como en el ejemplo de la Figura II.14.

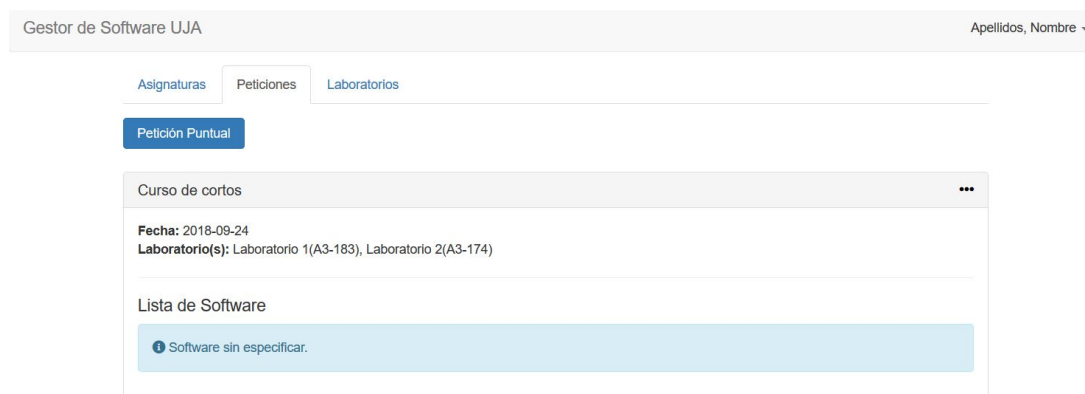


Figura II.14: Ejemplo de petición puntual

En la Figura II.15 podemos ver las diversas opciones que podemos realizar con una petición puntual:

- Enviar ticket: crear ticket para la petición puntual.
- Editar: modificar los datos de la petición puntual.
- Eliminar: eliminar la petición puntual.
- Añadir software: añadir software a la petición. Se realiza como en el caso de la asignatura



Figura II.15: Menú para una petición puntual

Gestionar laboratorios

En el caso del usuario “Profesor” sólo puede consultar los datos de los laboratorios (Figura II.20).

Vista Administrador

Al iniciar sesión con un usuario de tipo “Administrador” podremos gestionar los diferentes elementos del sistema como asignaturas, software, etc. Aparecerá un menú en la parte superior dividido en “pestañas” para dirigirnos a los diferentes apartados.

Gestor de Software UJA Apellidos, Nombre ▾

Tareas Pendientes Tareas Finalizadas **Laboratorios** Asignaturas Software Usuarios

Añadir Laboratorio

Laboratorio 1 (A3-183) ...

Numero de puestos: 20

Laboratorio 2 (A3-174) ...

Numero de puestos: 20

Laboratorio 3 (A3-172) ...

Numero de puestos: 20

Laboratorio 4 (A3-170) ...

Numero de puestos: 20

Laboratorio 5 (A3-176) ...

Numero de puestos: 20

Figura II.16: Pestaña laboratorios

Tareas pendientes

La pestaña “Tareas pendientes” muestra las diferentes tareas pendientes por realizar, con los datos necesarios como la asignatura, software, etc.

Gestor de Software UJA Apellidos, Nombre ▾

Tareas Pendientes Tareas Finalizadas Laboratorios Asignaturas Software Usuarios

Laboratorio 1 Nombre Apellidos

30-08-2018 15-02-2018

Petición para: Programación Orientada a Objetos
Curso: 1º
Cuatrimestre: 2º

Lista de Software

Nombre	Versión	Plataforma	Idioma	Licencia
Visual Studio Community	2015	Windows	Español	Gratuito

Finalizar

Filtros

Todos los cursos ▾

Todos los cuatrimestres ▾

Todos los laboratorios ▾

Asignaturas y Peticiones ▾

Todos los solicitantes ▾

Fecha Solicitud ASC. ▾

Figura II.17: Ejemplo de Tarea pendiente

Finalizar tarea

Las tareas pueden ser marcadas como finalizadas una vez que los técnicos de laboratorio instalen el software solicitado. Al finalizar la tarea el sistema se encargará de enviar un correo electrónico ¹ al usuario que solicitó el software, informándolo de que dicho software ya está instalado.

¹ Si se ejecuta la aplicación en local el servicio de mensajería, encargado de enviar los correos electrónicos no podrá ser ejecutado y se lanzará un error.

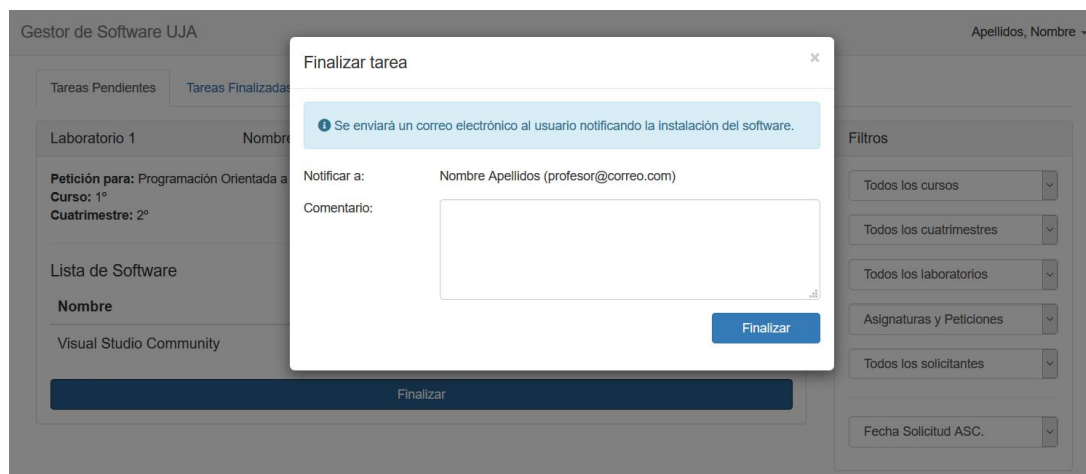


Figura II.18: Finalizar tarea pendiente

Tareas finalizadas

La pestaña “Tareas finalizadas”, como su nombre indica, muestra las tareas marcadas como finalizadas. Añade información a la tarea como la fecha de finalización o el usuario que ha finalizado dicha tarea.

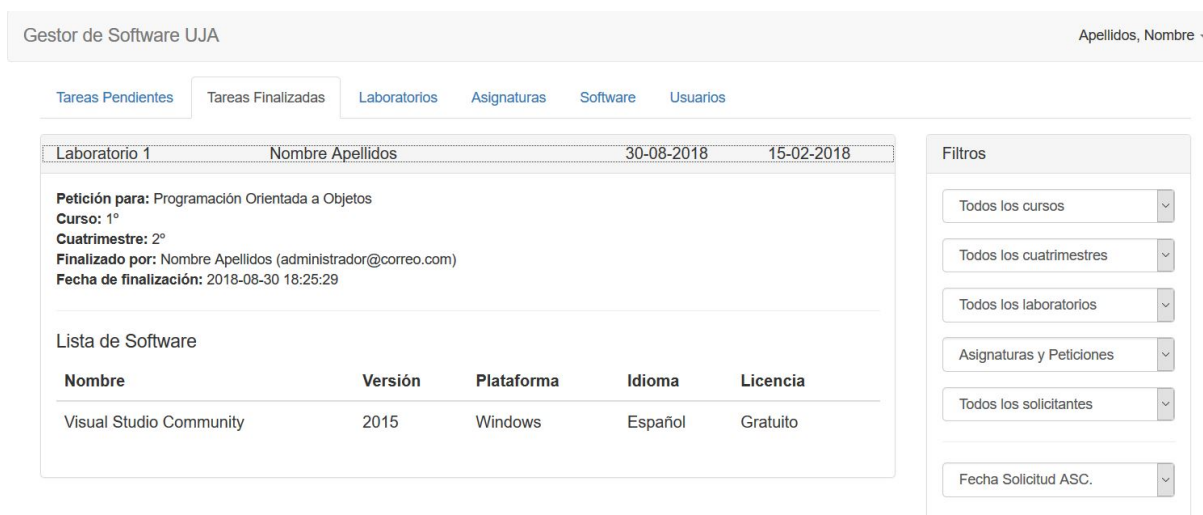


Figura II.19: Ejemplo tarea finalizada

Gestionar Laboratorios

La pestaña “Laboratorios” nos permite gestionar los laboratorios existentes en la base de datos, así como mostrar el software instalado en dichos laboratorios.

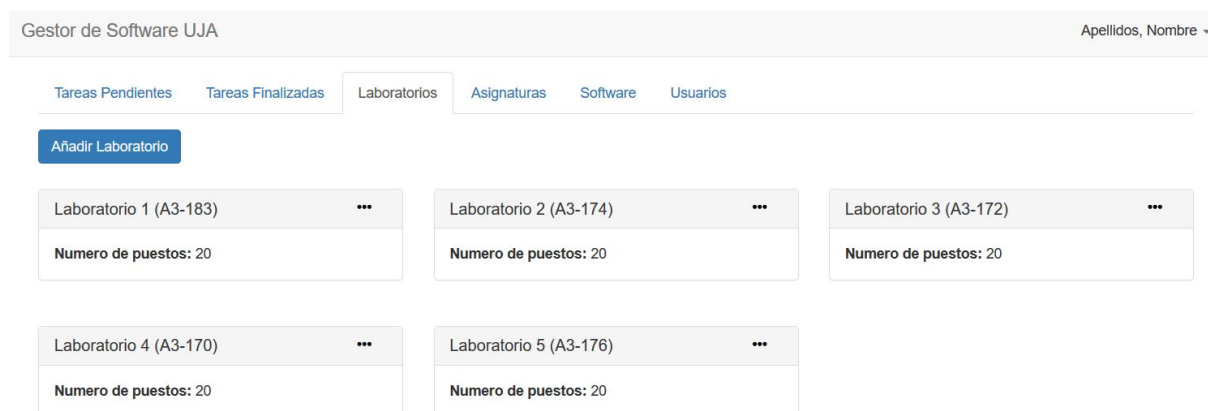


Figura II.20: Pestaña Laboratorios

Para cada laboratorio disponemos de un menú con diferentes opciones, “Editar”, “Software instalado”.

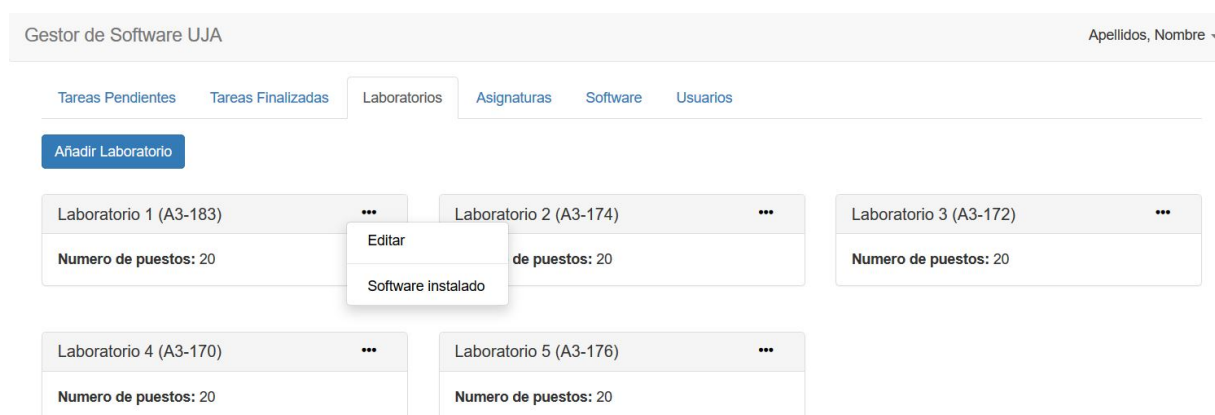


Figura II.21: Menú laboratorio

Añadir laboratorio

Para añadir un laboratorio sólo se deben completar los campos. El campo “Observaciones” permite añadir información adicional sobre el laboratorio como módulos a instalar, página web de descarga para el software, etc.

The screenshot shows a web application titled 'Gestor de Software UJA'. In the background, there are tabs for 'Tareas Pendientes' and 'Tareas Finalizadas', and a list of laboratories. A modal window titled 'Añadir Laboratorio' is open in the center. It contains the following fields: 'Nombre' (text input), 'Dependencia' (text input), 'Nº de puestos' (number input with a spinner), and 'Observaciones' (text area). A blue 'Añadir' button is at the bottom right of the modal.

Figura II.22: Añadir laboratorio

Editar laboratorio

Permite modificar los datos del laboratorio o eliminarlo.

The screenshot shows the same web application. The modal window is now titled 'Editar Laboratorio'. The fields are pre-filled with data: 'Nombre' is 'Laboratorio 1', 'Dependencia' is 'A3-183', and 'Nº de puestos' is '20'. The 'Observaciones' field is empty. At the bottom of the modal, there are two buttons: a red 'Eliminar' button on the left and a blue 'Guardar' button on the right.

Figura II.23: Editar laboratorio

Software instalado

Muestra una lista del software que está instalado en el laboratorio, la cual podemos modificar eliminando los elementos disponibles.

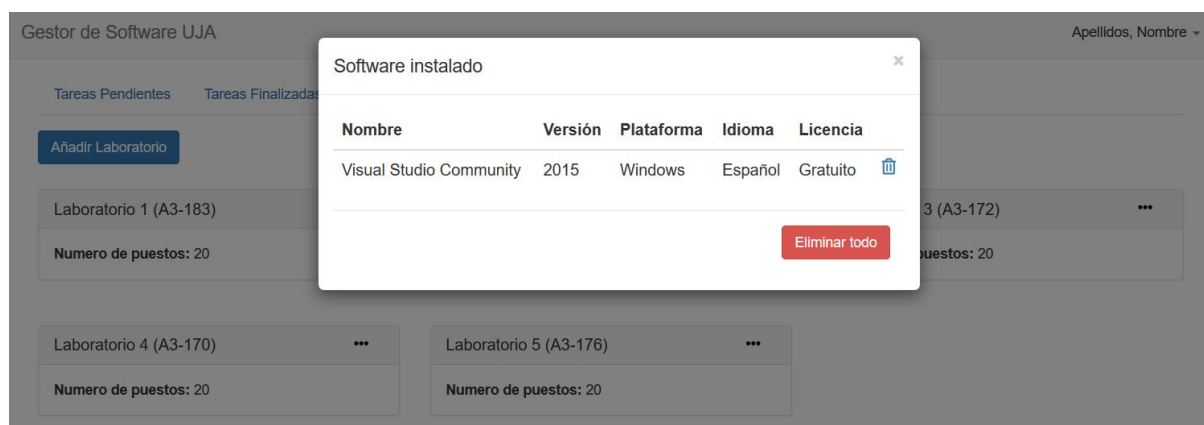


Figura II.24: Ejemplo Software instalado en laboratorio

Gestionar Asignaturas

La pestaña “Asignatura” muestra una tabla con las diferentes asignaturas y los datos de estas (nombre, curso, cuatrimestre y laboratorio).

Gestor de Software UJA					Apellidos, Nombre ▾
Tareas Pendientes	Tareas Finalizadas	Laboratorios	Asignaturas	Software	Usuarios
Añadir Asignatura Asignar Laboratorios					
Nombre	Curso	Cuatrimestre	Laboratorio		
Álgebra	1º	2º	Sin asignar	✎	
Análisis y métodos numéricos	1º	1º	Sin asignar	✎	
Arquitectura de Computadores	2º	1º	Sin asignar	✎	
Auditoría Informática	3º	2º	Sin asignar	✎	
Bases de datos Distribuidas	3º	2º	Sin asignar	✎	
Calidad del Software	3º	2º	Sin asignar	✎	

Figura II.25: Pestaña “Asignaturas”

Añadir asignatura

Para añadir una nueva asignatura al sistema sólo se ha de completar los datos, los laboratorios no son necesarios completarlos en un principio.

The screenshot shows a web application titled 'Gestor de Software UJA'. In the background, there is a table with columns for 'Nombre', 'Curso', 'Cuatrimestre', and 'Laboratorio'. The 'Nombre' column lists subjects like 'Álgebra', 'Análisis y métodos numéricos', 'Arquitectura de Computadores', 'Auditoría Informática', 'Bases de datos Distribuidas', and 'Calidad del Software'. The 'Curso' column shows '3º' and '2º'. The 'Cuatrimestre' column shows 'Sin asignar'. The 'Laboratorio' column has edit icons. Overlaid on this is a modal dialog box titled 'Añadir Asignatura'. It contains the following fields: 'Nombre' (text input), 'Curso' (dropdown menu with 'Seleccionar' selected), 'Cuatrimestre' (dropdown menu with 'Seleccionar' selected), and 'Laboratorio' (a grid of five buttons labeled 'Laboratorio 1' through 'Laboratorio 5'). At the bottom right of the dialog is a blue 'Añadir' button.

Figura II.26: Añadir asignatura

Editar asignatura

En la Figura II.27 se muestra un ejemplo de la vista de edición de una asignatura. En este caso la opción de eliminar la asignatura aparece en dicha vista.

This screenshot shows the 'Editar Asignatura' modal dialog box. The fields are pre-filled with data: 'Nombre' is 'Álgebra', 'Curso' is '1º', and 'Cuatrimestre' is '2º'. The 'Laboratorio' section still has the five buttons. At the bottom of the dialog, there are two buttons: a red 'Eliminar' button on the left and a blue 'Guardar' button on the right.

Figura II.27: Ejemplo de modificación de una asignatura

Asignar laboratorios

Esta funcionalidad nos permite asignar los laboratorios dónde se va a impartir una asignatura de una forma muy intuitiva y más rápida que editar para cada asignatura los laboratorios donde se imparte.

Gestor de Software UJA

Apellidos, Nombre ▾

Laboratorios

Seleccionar Laboratorio ▾

Asignaturas

¡Primero debes de seleccionar un laboratorio!

Figura II.28: Vista para Asignar laboratorios

Gestor de Software UJA

Apellidos, Nombre ▾

Laboratorios

Laboratorio 1 (A3-183) ▾

Programación Orientada a Objetos

Asignaturas

- Álgebra
- Análisis y métodos numéricos
- Arquitectura de Computadores
- Auditoría Informática
- Bases de datos Distribuidas
- Calidad del Software
- Desarrollo Ágil
- Desarrollo de Aplicaciones Empresariales
- Desarrollo de Aplicaciones Web
- Desarrollo de Software para Dispositivos Móviles
- Diseño de Algoritmos
- Diseño e Implantación de Servidores
- Electrónica Digital
- Estadística
- Estructuras de Datos
- Fundamentos de Arquitectura de Computadores

Figura II.29: Ejemplo de asignación de laboratorios

La Figura II.29 muestra un ejemplo de la vista para la asignación de laboratorios. En ella se puede observar que al seleccionar asignaturas de la lista a la derecha, se activa la flecha hacia la izquierda. Pulsando sobre ella podemos asignar las asignaturas seleccionadas a el laboratorio seleccionado.

Gestionar Software

La pestaña “Software” muestra una lista del software existente en modo de tabla así como sus datos.

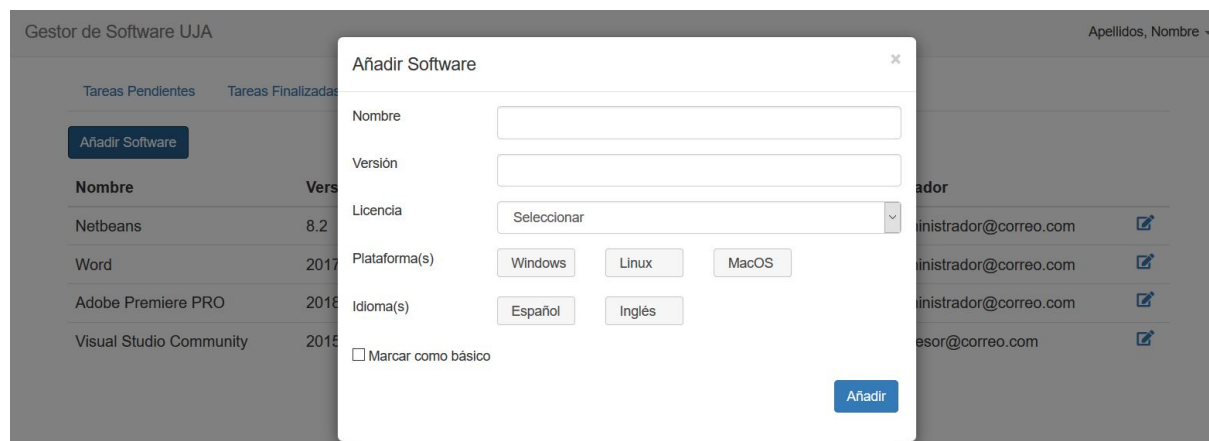


Nombre	Versión	Plataforma	Idioma	Licencia	Básico	Creador
Netbeans	8.2	Windows	Español, Inglés	Gratuito	No	administrador@correo.com
Word	2017	Windows, MacOS	Español	De pago	No	administrador@correo.com
Adobe Premiere PRO	2018	Windows, MacOS	Español, Inglés	De pago	No	administrador@correo.com
Visual Studio Community	2015	Windows	Español	Gratuito	No	profesor@correo.com

Figura II.30: Pestaña “Software”

Añadir software

Para añadir un nuevo software al sistema sólo debemos completar los datos que aparecen en pantalla (Figura II.31).



Nombre

Versión

Licencia

Plataforma(s)

Idioma(s)

☐ Marcar como básico

Añadir

Figura II.31: Añadir software al sistema

Editar software

En la Figura II.32 se muestra un ejemplo dónde se muestra un ejemplo de la vista de edición de un software.

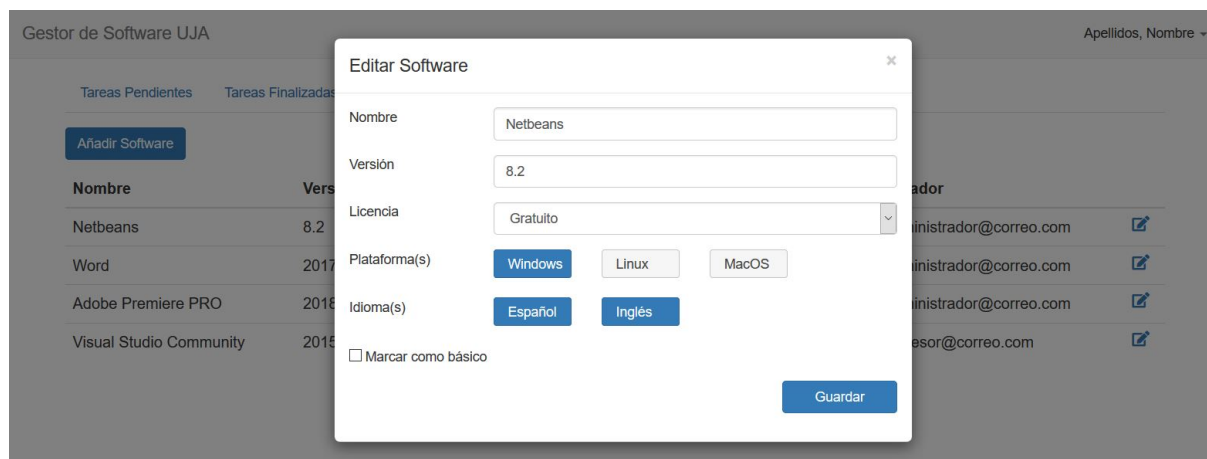


Figura II.32: Ejemplo de modificación de software

Gestionar usuarios

En la pestaña “Usuarios” se muestra una lista en forma de tabla de los diferentes usuarios así como de sus datos.

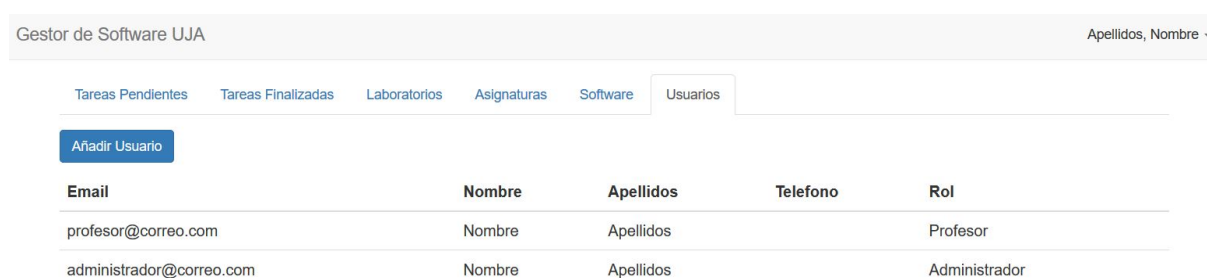


Figura II.33: Pestaña “Usuarios”

Añadir usuarios

Para añadir un nuevo usuario al sistema tan solo es necesario el correo electrónico con el que dicho usuario iniciará sesión.

El sistema se encargará de generar una contraseña segura y aleatoria y mandará un correo electrónico ² al nuevo usuario con los datos necesarios para que inicie sesión. Posteriormente, el nuevo usuario podrá modificar la contraseña por otra de su agrado.

²Si se ejecuta la aplicación en local el servicio de mensajería, encargado de enviar los correos electrónicos no podrá ser ejecutado y se lanzará un error.

The screenshot shows a web application interface for 'Gestor de Software UJA'. A modal window titled 'Añadir Usuario' is open in the center. The modal contains a light blue information box with a message: 'Se enviará un correo al usuario con la contraseña para iniciar sesión.' Below this, there is a text input field containing 'usuario@example.com' and a dropdown menu labeled 'Seleccionar tipo de usuario'. At the bottom right of the modal is a blue button labeled 'Añadir'. In the background, the application's sidebar is visible with a 'Añadir Usuario' button and a table with email addresses. The main content area shows a table with roles 'Profesor' and 'Administrador'.

Gestor de Software UJA	
Tareas Pendientes	Tareas Finalizadas
Añadir Usuario	
Email	
profesor@correo.com	
administrador@correo.com	

Apellidos, Nombre	
Rol	
Profesor	
Administrador	

Figura II.34: Añadir usuario