

---

This is the **published version** of the bachelor thesis:

Sánchez Ruiz, Ingrid; Pons Aroztegui, Jordi, tut. Eina web de suport als responsables acadèmics. 2025. (Enginyeria Informàtica)

---

This version is available at <https://ddd.uab.cat/record/317524>

under the terms of the  license

# Herramienta web de soporte a responsables académicos

Ingrid Sánchez Ruiz

30 de junio de 2025

**Resumen–** En las facultades y escuelas universitarias, algunos profesores asumen cargos de gestión con diversas responsabilidades académicas. Mantener al día todas estas tareas y obligaciones puede resultar una labor compleja, especialmente cuando el cargo cuenta con muchas responsabilidades. Por ello, surgió la idea de este proyecto. La herramienta web de soporte a responsables académicos es una plataforma diseñada para facilitar la labor de estos cargos, permitiéndoles compartir información tanto entre ellos como con futuros responsables, además de gestionar sus tareas de forma más cómoda y eficiente. Esta gestión se realiza mediante la publicación de tareas con fechas asociadas y un buscador que permite filtrar por períodos temporales de realización. El desarrollo del proyecto se ha llevado a cabo utilizando Vue.js para el frontend y Node.js junto con Express para el backend, todo ello siguiendo una metodología de cascada iterativa.

**Paraules clau–** Actividades, Coordinación, Express, Gestión, Node.js, Plazos, Roles, Soporte, Universidad, Vue.js

**Abstract–** At university faculties and schools, some professors take on management roles with various academic responsibilities. Keeping all these tasks and duties up to date can be a complex endeavor, especially when the role involves multiple responsibilities. This is what led to the idea behind this project. The web-based support tool for academic coordinators is a platform designed to facilitate the work of those in such roles, enabling them to share information both among themselves and with future coordinators, as well as manage their tasks in a more convenient and efficient way. Task management is carried out through the publication of tasks with associated dates and a search function that allows filtering by time periods. The project has been developed using Vue.js for the frontend and Node.js together with Express for the backend, all following an iterative waterfall methodology.

**Keywords–** Activities, Coordination, Express, Management, Node.js, Deadlines, Roles, Support, University, Vue.js

---

## 1 INTRODUCCIÓN - CONTEXTO DEL TRABAJO

CUALQUIER trabajo es complejo cuando empiezas desde cero, sin conocimientos o posiblemente sin la experiencia suficiente. Pero en el caso de puestos donde solo se hace cargo una persona, como ocurre en el caso de las coordinaciones de estudios y otros cargos

- E-mail de contacto: [ingrid.sanchezr@autonoma.cat](mailto:ingrid.sanchezr@autonoma.cat)
- Mención realizada: Tecnologías de la Información
- Trabajo tutorizado por: Jordi Pons Aróstegui (DEIC)
- Curso 2024/25

académicos de gestión, esto puede llegar a ser mucho más tedioso.

Si a esto le sumamos la escasa información que hay referente a las tareas que se deben realizar, que algunas veces se describen en normativas relacionadas con el rol, y la cantidad de renovaciones que hay debido a la duración en estos cargos, los cuales suelen ser de tres años, con posibilidad de renovación por tres años más, se vuelve más complicado. Por estos motivos, el tiempo que se emplea en recopilar la información necesaria es más elevado del que debería y ocurre de manera recurrente, generando que el nuevo responsable acabe dedicando parte de su tiempo a entender su función, en lugar de centrarse en las tareas que ya deberían estar en marcha.

Por otro lado, existe la dificultad de recordar todas las tareas repetitivas que deben realizarse a lo largo del curso académico, algunas con fechas específicas de realización y otras que requieren compartir documentos. Recordarlas todas se vuelve complicado, especialmente debido a las limitaciones que presentan las plataformas existentes para gestionar estas actividades de forma eficiente.

Por todas estas casuísticas surgió la motivación de desarrollar una herramienta web de soporte a responsables académicos, la cual facilite tanto el traspaso de información como el control de las tareas a realizar.

Dada la complejidad en la gestión de tareas entre los distintos cargos académicos, se tomó la decisión de organizar las actividades y tareas por roles, ya que hay algunos docentes que comparten funciones a pesar de ocupar cargos diferentes. De esta manera se evita duplicidad de información, se facilita tanto su búsqueda como su consulta y se permite la colaboración.

En este documento se explicarán los objetivos del proyecto, el estado del arte, los requisitos, el estado actual de la herramienta, la metodología y tecnologías utilizadas, los resultados obtenidos y las conclusiones basadas en ellos, incluyendo posibles líneas futuras de mejora sobre la herramienta desarrollada.

## 2 OBJETIVOS

Se ha considerado conveniente dividir los objetivos de este Trabajo de Fin de Grado en dos categorías, una referente a los objetivos de la propia herramienta y otros referentes a la evolución personal que se pretendía conseguir con este proyecto, así como los conocimientos a desarrollar.

### 2.1. Objetivos funcionales de la herramienta

Los objetivos que se buscaban alcanzar con la herramienta web de gestión de tareas son los siguientes:

- Facilitar el traspaso de información entre diferentes cargos, reduciendo así el tiempo de adaptación.
- Optimizar el seguimiento de tareas a realizar, reduciendo así los errores humanos a causa de olvidar tareas o falta de información.
- Redistribuir las tareas de un rol a otro de manera rápida y sencilla.
- Mejorar el rendimiento en la gestión de tareas y responsabilidades académicas.
- Compartir información de buenas prácticas entre responsables académicos que ocupan los mismos cargos en distintos ámbitos.

### 2.2. Objetivos de desarrollo personal

Los objetivos que se buscaban alcanzar relativos al crecimiento personal y profesional con este proyecto son los siguientes:

- Mejorar las habilidades de desarrollo web.
- Mejorar la capacidad de planificación y cumplimiento de la misma.
- Experimentar cada uno de los procesos de desarrollo de un proyecto (recopilación de requisitos, creación de la base de datos, desarrollo de Backend y Frontend y testeo de la página) de forma seguida y desde cero.

## 3 ESTADO DEL ARTE

Actualmente, existen diversas plataformas y aplicaciones para la gestión de tareas, las cuales se dividen principalmente en dos tipos: gestión de tareas basada en un calendario y gestión de tareas basada en una *checklist*.

En el caso de plataformas o aplicaciones de gestión de tareas basadas en un calendario tenemos el clásico Google Calendar. Este tipo de herramientas permiten añadir tareas de manera temporal, ya sea con un periodo de realización asignado o una fecha concreta. Se pueden compartir las tareas de manera grupal, lo cual cuadra con la necesidad de distribuir las tareas en roles. También permite adjuntar archivos, facilitando la difusión de recursos entre los usuarios. Sin embargo, no cubre la posibilidad de crear tareas sin fechas específicas, lo cual es necesario para el tipo de tareas repetitivas, las cuales van surgiendo de manera desperdigada en el tiempo, como por ejemplo la gestión de un contrato de prácticas externas.

Por contrapartida, plataformas o aplicaciones de gestión de tareas basadas en una *checklist* como Microsoft Planner sí que se ajustan más a las necesidades de los responsables académicos. Cuenta con una creación de tareas detallada, donde se pueden agregar archivos o fechas de realización pero no cuentan con períodos de realización con fechas específicas. Estas tareas se pueden dividir en grupos, lo cual simularía los roles. Además, cuenta con una funcionalidad de comentarios sobre las tareas. Sin embargo, estos comentarios solo pueden ser públicos, por lo tanto anotaciones personales deben de ir en otra plataforma. Este tipo de plataformas tampoco permiten la diferenciación de privilegios, de manera que cualquier usuario puede manipular los grupos y las tareas.

Por lo general, estas plataformas cuentan con bastantes recursos útiles para la gestión de tareas, especialmente las del segundo grupo. Sin embargo, la falta de personalización en estas herramientas puede dificultar la forma en la que se transmite la información. A esto se suma la ausencia de control sobre los privilegios de los usuarios, lo que puede derivar en errores como la eliminación de información esencial para el correcto desarrollo del trabajo, o el acceso no autorizado a información por parte de usuarios que no deberían haber sido añadidos al grupo. Además, no permiten realizar anotaciones de manera privada, teniendo que recurrir a otra plataforma o aplicación, ya sea del mismo grupo (Microsoft o Google) o externa.

## 4 ANÁLISIS DE LA HERRAMIENTA

A continuación, se procederá a analizar los conceptos generales que definen el funcionamiento de la herramienta, incorporando en esta explicación los detalles más relevantes que permiten comprender su estructura y finalidad.

En la plataforma, los **roles** determinan la distribución de las actividades y tareas. Un usuario puede tener asignados uno o varios roles, según sus diferentes responsabilidades, y cada rol conlleva un conjunto específico de actividades y tareas asociadas. Por ejemplo, un usuario con el rol de “Coordinador de prácticas” tendrá tareas distintas a las de otros roles.

Las **actividades** son la agrupación lógica de tareas. Cada rol contiene un número determinado de actividades, las cuales permiten agrupar las tareas que tienen un tema en común. Por ejemplo, la actividad “Prácticas en empresa” incluye distintas tareas relativas a esa temática.

A su vez, las **tareas** constituyen la parte esencial de esta herramienta. Cada tarea contiene la información necesaria sobre qué se debe hacer, junto con la posibilidad de añadir detalles adicionales sobre cómo y cuándo realizarla.

Finalmente, están los **períodos**, que establecen el intervalo temporal durante el cual se deben llevar a cabo ciertas tareas. Estos períodos ayudan a organizar y planificar las actividades, aunque no todas las tareas cuentan con un período específico asignado, permitiendo así cierta flexibilidad en su ejecución.

En resumen, podemos decir que un cargo académico tiene, como mínimo, un **rol** asignado que conlleva la responsabilidad sobre un conjunto de **actividades** que se materializan mediante **tareas** concretas que tienen lugar en **períodos** de tiempo establecidos.

Por otra parte, se han definido tres tipos de usuarios dentro de la plataforma, cada uno con funciones específicas y permisos distintos:

- **Administrador:** Este usuario controla la herramienta. Se encarga de gestionar las solicitudes de los usuarios para pertenecer a un rol y de gestionar los roles de los usuarios. Además también se encarga de eliminar usuarios de la plataforma y de crear, modificar o eliminar los períodos predefinidos.
- **Administrador del rol:** Este usuario es quien publica las tareas del rol en el que sea administrador. Puede crear actividades y eliminar, modificar y añadir tareas y comentarios de cualquier usuario, siempre que estos comentarios sean públicos.
- **Usuario común:** Este usuario solo puede añadir comentarios a tareas ya creadas y solo podrá modificar o eliminar sus propios comentarios, nunca los de otros usuarios.

Cualquier usuario tendrá la posibilidad de solicitar al administrador de la herramienta diversas acciones relacionadas con la gestión de roles. Entre estas, podrá pedir la crea-

ción de un nuevo rol y su correspondiente asignación al mismo, así como solicitar la desvinculación de un rol que tenga actualmente asignado, o bien la vinculación a un rol ya existente dentro del sistema.

## 5 METODOLOGÍA Y PLANIFICACIÓN

Para el correcto desarrollo del proyecto se ha utilizado una metodología en cascada [1], principalmente porque las que suelen ser mejores candidatas, es decir, las metodologías ágiles, se han considerado más características de un desarrollo grupal, donde se van haciendo partes del producto en cada sprint ya que la expansión es tan grande que hay muchas subpartes por hacer.

Por ello se optó por un modelo más tradicional como es el de cascada, sin embargo, era necesario algo más flexible que permitiese modificar partes del trabajo volviendo a fases anteriores, de ahí la idea de utilizar un modelo de cascada iterativa.

Un modelo de cascada iterativa [2] básicamente consta en pasar por todas las fases de desarrollo del proyecto: planificación, análisis, diseño, desarrollo y verificación, de manera secuencial, pero, a diferencia del modelo de cascada tradicional, con posibilidad de volver a fases anteriores del proyecto tras la valoración, dejando esa rigidez que pudiese llegar a dificultar el desarrollo natural.

Como se puede observar en la tabla (TABLA 1), se dedicó un día a la planificación, doce días al análisis y trece días al diseño. Esta última fase se retomó posteriormente, tras haber sido completada inicialmente, haciendo uso de la iteración propia de la metodología en cascada iterativa. A continuación, se destinaron cincuenta y tres días al desarrollo, de los cuales aproximadamente la mitad se dedicaron al desarrollo del frontend de la plataforma, mientras que gran parte de la otra mitad se centró en el backend. El menor tiempo de desarrollo se destinó a la base de datos. Finalmente, se emplearon cinco días en la fase de verificación de la aplicación.

TABLA 1: FASES DEL DESARROLLO

Fase	Inicio	Fin	Tiempo total
Planificación	27/02/2025	28/02/2025	1 día
Análisis	28/02/2025	11/03/2025	12 días
Diseño	11/03/2025	24/03/2025	13 días
Desarrollo	24/03/2025	01/05/2025	53 días
Verificación	01/05/2025	05/06/2025	5 días

## 6 DESARROLLO DE LA HERRAMIENTA

A continuación, se detalla el proceso de desarrollo de la herramienta, incluyendo el diseño de la herramienta, el diseño de la base de datos, el desarrollo del frontend y su correspondiente backend, así como un breve apartado dedicado a ciertos aspectos de seguridad implementados en la plataforma.

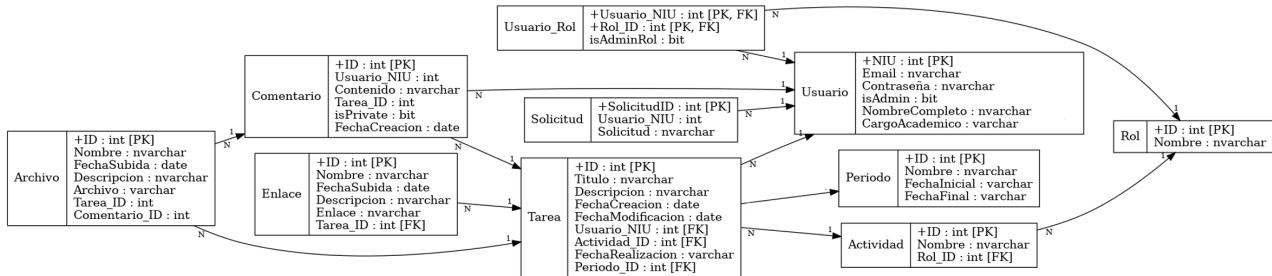


Fig. 1: Diagrama Entidad-Relación de la base de datos

## 6.1. Diseño de la herramienta

Como se muestra en las imágenes del apéndice, la interfaz de usuario de la herramienta presenta un diseño visual limpio y coherente, orientado a facilitar la navegación y la gestión de tareas de forma intuitiva. Para ello se emplea una paleta de colores dominada por el verde institucional de la UAB, lo que refuerza la identidad visual de la plataforma.

En la parte superior de la pantalla se encuentra el *navbar*, de estilo minimalista, que organiza de forma clara los elementos funcionales esenciales. Esta barra incluye el logotipo de la universidad, el rol seleccionado por el usuario, una barra de búsqueda centrada y accesible, un menú con opciones como *Administrar*, *Tasques* y *Calendari*, un ícono de usuario desde el cual se accede a la información del perfil junto a la solicitud de gestión de roles, y otro ícono de cierre de sesión. La tipografía en color blanco contrasta adecuadamente con el fondo verde, además, los elementos están distribuidos de manera equilibrada para maximizar la legibilidad y la accesibilidad de la herramienta.

El diseño general se apoya en elementos visuales como márgenes amplios, esquinas redondeadas y botones de gran tamaño, que permiten una interacción cómoda. Los botones utilizan un tono de verde más claro o azul grisáceo, acompañado de sombras sutiles que resaltan su función dentro de la interfaz.

Junto al verde predominante, se utilizan tonalidades complementarias como el beige, el marrón y un azul claro, generando un contraste visual suficiente que facilita la identificación de los distintos elementos. Además, se emplean variaciones de color en la mayoría de los botones al pasar por encima, lo que contribuye a reforzar su función y mejorar su comprensión por parte del usuario.

Los iconos conservan un estilo minimalista que complementa la redondez del diseño y utiliza la paleta de colores de la página. Así, se logra una apariencia coherente sin perder claridad ni facilidad de comprensión para el usuario.

Además, la interfaz se presenta en catalán, el idioma propio de la universidad. Esta decisión refuerza la adecuación de la herramienta al entorno académico de la UAB y garantiza una experiencia alineada con el resto de servicios institucionales.

En definitiva, la herramienta presenta una estética moderna, minimalista y amigable, que prioriza la claridad visual, la facilidad de uso y la identidad institucional.

## 6.2. Base de datos

En esta subsección se presenta el diseño final de la base de datos mediante un diagrama entidad-relación (Fig. 1), incluyendo detalles sobre las decisiones adoptadas y el funcionamiento de la misma.

La base de datos consta de diez tablas, de las cuales una de ellas muestra una relación de muchos a muchos (Usuario.Rol). El resto de las tablas se relacionan a través de campos que las conectan, como puede ser el campo Rol\_ID en la tabla Actividad, el cual conecta dicha tabla con la tabla Rol, esto sucede por una relación de uno a muchos, ya que una actividad pertenece a un solo rol, pero un rol puede tener n tareas asignadas.

En esta base de datos se han usado restricciones para poder validar el correcto funcionamiento de la web. Con la restricción SQL CHECK aseguramos que el elemento archivo pertenece a un comentario o a una tarea (está completo el campo Tarea.ID o el campo Comentario.ID), en ningún momento pertenece a ambos o a ninguno a la vez, ya que un archivo está asociado a un título y una descripción, las cuales prácticamente en ninguna ocasión se verán duplicadas. Por ello se ha recurrido a tratar un Archivo como un objeto que solo pertenece a una tarea o a un comentario.

También se ha hecho uso de restricciones de clave foránea [3] para enlazar las diferentes tablas, de manera que, por ejemplo, Usuario.NIU de Tarea tenga una relación con NIU de la tabla Usuario, creando así la relación de la base de datos, gracias a la cual se evita la inserción de tareas vinculadas a usuarios inexistentes, y se facilita la realización de consultas que integren información de ambas tablas de manera coherente.

Todas las fechas que se guardan respectivas a la realización de las tareas (las fechas de los períodos y la fecha de realización) son *varchar*, esta decisión surge de la necesidad de guardar las fechas sin año (DD/MM), dado que, como se ha comentado anteriormente, guardar un año de realización no tiene lógica si lo que se requiere son tareas que puedan ser de utilidad para cualquier curso académico.

### 6.3. Frontend

La plataforma cuenta con un frontend, el cual está basado principalmente en la herramienta de Vue.js, utilizando *route* [4] para poder crear el flujo de vistas en la herramienta web. Consta de diecisiete vistas, tres componentes y ocho rutas (Fig. 2), de las cuales dos tienen rutas hijas, una con cinco rutas hijas y otra con cuatro rutas hijas, además de un App.vue principal.

```
const routes = [
  { path: '/', component: Tasques },
  { path: '/calendari', component: Calendari },
  { path: '/perfil', component: Perfil },
  { path: '/afegir-tasca/:activitatId', component: AfegirTasca, props: true },
  { path: '/modificar-tasca/:tascaId', component: ModificarTasca, props: true },
  { path: '/resultats', name: 'Resultats', component: Resultats },
  {
    path: '/tasca/:id',
    component: TascaDetail,
    props: true,
    children: [
      { path: '', name: 'tasca-info', component: TascaInfo, props: true },
      { path: 'links', name: 'tasca-links', component: Link, props: true },
      { path: 'arxius', name: 'tasca-arxius', component: Arxius, props: true },
      { path: 'comentaris', name: 'tasca-comentaris', component: Comentaris, props: true },
      { path: 'comentari', name: 'tasca-comentari', component: AfegirComentari, props: true }
    ]
  },
  {
    path: '/admin',
    component: AdminDetail,
    children: [
      { path: '', component: AdminSolicitudes },
      { path: 'períodes', component: AdminPeríodes },
      { path: 'usuaris', component: AdminUsuaris },
      { path: 'falsificar', component: AdminFalsificar }
    ]
  }
]
```

Fig. 2: Organización de rutas del frontend

App.vue es el esqueleto principal, donde se monta el navbar [5] de la herramienta y se dispone un router-link, el cual muestra las diferentes rutas definidas. De esta manera se evita tener que copiar y pegar el código relativo al navbar, reduciendo así posibles errores y problemáticas.

Desde el navbar se puede acceder a la vista Perfil.vue, la cual muestra los datos del usuario junto a los roles a los que pertenece y, a su vez, permite enviar una solicitud de modificación, agregación o eliminación de rol al administrador principal de la herramienta web.

Por otra parte está la **ruta por defecto** ('/') la cual corresponde a la visualización principal de actividades y tareas. En esta vista se utiliza el componente Tasca.vue, el cual es un desplegable de tareas en base a la actividad enviada mediante props de la vista principal, de manera que se desplegarán tantos componentes como actividades haya en el rol seleccionado.

También está la ruta **/calendari**, la cual muestra una serie de filtros que se utilizan como buscador de tareas, de manera que si pulsas en “filtrar” aparecerán las actividades que cumplan con los filtros seleccionados. Existen diversas restricciones explicadas con anterioridad en el funcionamiento, como el hecho de no poder filtrar únicamente por rol ya que obliga a seleccionar primero uno de los dos filtros temporales antes, o que solo se pueda seleccionar uno de estos filtros, desactivando la opción de seleccionar el otro.

La ruta **/afegir-tasca/:activitat** muestra el formulario para poder añadir una tarea a la actividad donde se ha pul-

sado “afegir tasca”, un botón que es parte del componente anteriormente mencionado, Tasca.vue. Se le pasa mediante props el ID de la actividad donde agregar la tarea para el correcto funcionamiento del backend. Esta vista cuenta con dos componentes, AfegirArxiu.vue y AfegirLink.vue, los cuales permiten subir un archivo o enlace, introducir un título a estos objetos y, si se desea, una descripción. Está permitido añadir un máximo de diez archivos, el mismo límite que para los enlaces.

La ruta **/modificar-tasca/:tascaId** muestra el mismo formulario que la ruta anterior, con la diferencia de que al entrar a la sección ya se muestran los campos rellenados. Esto se debe a que muestra la información de la tarea a modificar que ya estaba completa, permitiendo modificar esa información, añadirla o eliminarla. También muestra los archivos subidos, junto a sus títulos y descripciones, así como los enlaces.

La ruta **/resultats** se encarga de mostrar por pantalla los resultados de una búsqueda realizada a través de la barra de búsqueda del *navbar*. En dichos resultados se indica el rol al que pertenece la coincidencia, la actividad y, finalmente, la tarea que cumple con el criterio de búsqueda. Este criterio consiste en tomar la cadena de texto introducida en la barra de búsqueda y mostrar las tareas cuyo título contenga dicha cadena en alguna parte. De cada tarea se muestran el título, el autor (en caso de tener uno) y la fecha de modificación o creación. Estas tareas permiten ser seleccionadas para mostrar más detalles de la misma en caso de así desecharlo.

Después se desarrollan las rutas padre, la primera de ellas es **/tasca:id**, la cual se encarga de mostrar las diferentes pantallas que tiene una tarea. Está formado por seis vistas y cinco hijos [6]. La vista padre es TascaDetail.vue, que corresponde al título, creador y fecha de modificación o creación de la tarea. Esta parte actúa igual que el App.vue, simplemente de manera añadida, puesto que esta vista padre se queda de manera permanente en cada uno de sus hijos.

El primer hijo, el cual es por defecto (**/tasca:id**) es la vista TascaInfo.vue, la cual muestra la información detallada de la tarea seleccionada y cuenta con tres botones para acceder al resto de hijos, los cuales representan los diferentes apartados de una tarea. Además de dos botones, uno para eliminar la tarea y otro para modificarla.

El segundo hijo (**/tasca:id/links**) muestra los enlaces asociados a la tarea, con su descripción y su título.

El tercer hijo (**/tasca:id/arxius**) hace exactamente lo mismo que el anterior, pero con los diferentes archivos.

El cuarto hijo (**/tasca:id/comentaris**) muestra todos los comentarios publicados en la tarea y los comentarios propios que sean privados, así como los archivos asociados a cada comentario y su descripción (el comentario como tal). En el caso de ser administrador del rol permite eliminarlos.

El quinto y último hijo (**/tasca:id/comentar**) muestra el formulario a completar para publicar un comentario, el cual consta del componente AfegirArxiu.vue, el comentario y un checkbox para indicar si el comentario es público o privado.

Finalmente, tenemos la segunda y última ruta padre, **/admin**. Esta ruta se encarga de gestionar todas las funcionalidades que pertenecen al administrador de la herramienta. En el *navbar* aparecerá una opción de “Administrador” en el caso de que el usuario que ha iniciado sesión sea administrador. En este caso la vista padre es AdminDetail.vue, que se encarga de mostrar un *navbar* adicional que muestre las cuatro funcionalidades de administrador.

El primer hijo, que corresponde a la ruta por defecto(**/admin**), muestra todas las solicitudes de rol no gestionadas enviadas por el resto de usuarios. Cada solicitud incluye el NIU del usuario que la ha realizado, una opción para eliminarla, y tres botones de acción:

El primer botón, “Respondré”, despliega un menú con todos los roles disponibles, una casilla para indicar si el usuario será administrador del rol, y un botón “Enviar”. Al enviarse, el usuario pasará a formar parte del rol seleccionado.

El segundo botón, “Crear rol”, permite introducir el nombre de un nuevo rol mediante un campo de texto, junto con la misma casilla de administrador y el botón “Enviar”. Si se confirma la acción, el nuevo rol se crea y el usuario pasa a pertenecer a él.

Por último, el botón “Eliminar rol” muestra un desplegable con los roles actuales del usuario solicitante y un botón “Eliminar”, que al ser pulsado elimina al usuario del rol seleccionado.

El segundo hijo, correspondiente a la ruta **/admin/periodes**, muestra el panel desde el cual el administrador puede gestionar los distintos períodos. Este panel ofrece tres funcionalidades principales:

- La primera, “*Nou període*”, permite la creación de un nuevo periodo mediante tres campos de entrada de texto: uno para el nombre del periodo y dos para especificar la fecha de inicio y de fin, ambas en formato DD/MM.
- La segunda funcionalidad, “*Modificar període*”, permite actualizar las fechas de un periodo ya existente.
- La tercera funcionalidad, “*Eliminar període*”, se encarga de suprimir un periodo previamente creado.

El tercer hijo (**/admin/usuaris**) permite eliminar un usuario, mostrando un desplegable con los NIUs de los usuarios registrados y un botón de “Eliminar”.

Y el cuarto y último hijo (**/admin/falsificar**) da la posibilidad de hacerse pasar por otro usuario con un desplegable que muestre todos los usuarios existentes y un botón de “Falsificar” el cual llevará al administrador a la página de inicio del usuario falsificado. Esta funcionalidad está hecha para el correcto testeo de la herramienta y en caso de pasar a producción deberá de ser eliminada.

## 6.4. Backend

El backend de la herramienta está realizado con Node.js junto a su framework Express. Consta de dos carpetas y de dos ficheros fuera de esas carpetas, ya que son ficheros principales, así como de un fichero .env [7], en el cual se guardan los datos necesarios y confidenciales para conectarse a la base de datos. De esta manera para cambiar la conexión solo se tendrá que manipular este fichero.

Index.js es el fichero principal, el cual inicia el servidor, crea una sesión para generar una cookie con la cual se consigue acceder a la información correcta y redirige mediante rutas y app.use a los diferentes archivos.

ConectionDB.js se encarga de establecer la conexión con la base de datos, guardando esta conexión en una pool promise, de manera que se pueda utilizar cada vez que se deba de hacer una petición a la base de datos sin tener que establecer la conexión de manera recurrente.

La primera carpeta es la de middleware, la cual solo guarda un archivo llamado iniciSessio.js, el cual es accedido cuando el usuario entra a la página web e introduce su NIU que será lo que se guarde en la cookie. Su función es colocar el NIU en una variable de manera que el resto de ficheros puedan acceder a esta información fácilmente.

La segunda carpeta es routes, la cual cuenta actualmente con ocho ficheros distintos:

- **comentaris.js**, el cual gestiona todas las llamadas al backend relativas a los comentarios. Cuenta con un *get* para recuperar la información de los comentarios de una tarea pasada por parámetro, un *post* para guardar un nuevo comentario en la base de datos (haciendo así que se publique) y un *delete* que se encarga de eliminar el comentario pasado por parámetro.
- **iniciSessio.js**, que comprueba que el NIU que se introduce para iniciar sesión esté en la base de datos y se encarga de gestionar el login y el logout.
- **perfil.js**, que se encarga de conseguir toda la información a mostrar del usuario según la variable de sesión y de enviar la solicitud de rol a la base de datos.
- **tasques.js**, el cual se encarga de gestionar todo lo relacionado con las tareas. Un *get* para conseguir la información de las actividades y tareas a partir del rol pasado por parámetro, un *post* para crear una nueva actividad, un *get* para recuperar toda la información de una tarea en específico, un *delete* que se encarga de eliminar una tarea y, por último, dos *get* que se encargan de recuperar la información de los archivos y los enlaces de la tarea junto a sus respectivos *delete*.
- **admin.js**, que se encarga de recuperar, modificar, añadir o eliminar la información correspondiente a las funcionalidades de administrador ya descritas en el frontend.
- **afegirTasca.js**, encargado de agregar una tarea y de modificarla con dos *post*.

- **buscador.js**, que consta de un *get* encargado de realizar la búsqueda de tareas según la cadena de texto enviada por parámetro.
- **calendari.js**, que se encarga de conseguir las actividades y tareas que coincidan con el criterio de búsqueda del usuario expresado a través de los filtros.

## 6.5. Aspectos de seguridad en la plataforma

Una de las principales medidas de seguridad implementadas en el sistema es la protección frente a ataques de inyección SQL, una de las vulnerabilidades más comunes en aplicaciones web que interactúan con bases de datos.

Para evitar este tipo de ataques, se utilizan consultas parametrizadas [8] a través de la librería mssql. En lugar de construir consultas dinámicamente concatenando valores directamente en la cadena SQL, el sistema define los parámetros de forma explícita, indicando su tipo y valor, como se puede observar en la Fig. 3. Esto garantiza que el contenido proporcionado por el usuario se trate estrictamente como datos, y no pueda ser ejecutado como parte del código SQL.

```
const result = await pool.request()
  .input('niu', sql.Int, niu)
  .input('mensaje', sql.NVarChar(sql.MAX), mensaje)
  .query(`
    INSERT INTO dbo.Solicitud (Usuario_NIU, Solicitud)
    OUTPUT INSERTED.SolicitudID
    VALUES (@niu, @mensaje)
  `);

const id = result.recordset[0].SolicitudID;
res.json({ success: true, id });
```

Fig. 3: Consulta parametrizada

Para garantizar la seguridad de las sesiones de usuario a través del NIU, la aplicación implementa mecanismos que impiden que las cookies utilizadas para gestionar la autenticación puedan ser accedidas o modificadas directamente desde el navegador.

Concretamente, se configura la cookie de sesión de manera que su contenido solo puede ser gestionado por el servidor. Esto se logra estableciendo propiedades que restringen el acceso desde scripts del lado del cliente, evitando así que herramientas del navegador o ataques como el Cross-Site Scripting (XSS) puedan capturar o alterar la cookie.

Además, el contenido de la cookie está firmado digitalmente mediante una clave secreta, lo que asegura que cualquier intento de modificación externa sea detectado y rechazado automáticamente por el servidor. De este modo, incluso si un atacante intentara alterar la cookie manualmente, esta no sería válida.

Estas medidas contribuyen a mantener la integridad de la sesión del usuario y a proteger el sistema contra accesos no autorizados o manipulaciones maliciosas desde el entorno del cliente.

## 6.6. Implantación

La implantación de la herramienta en el servidor institucional se contempló en una primera fase del proyecto, con el objetivo principal de facilitar su uso directo dentro del entorno académico.

No obstante, durante el proceso surgieron problemas de incompatibilidad entre el backend en Node.js y la configuración del servidor, orientado a PHP. Dado el tiempo limitado para resolverlos, se decidió posponer la implementación en el servidor y dejarla como una línea futura de trabajo.

## 7 DETALLES DEL FUNCIONAMIENTO

A continuación se profundizará en el funcionamiento específico de la herramienta, aportando detalles relevantes para su entendimiento.

En primer lugar, se decidió que cuando un usuario sea dado de baja por el administrador, se eliminaran todos sus comentarios, su información personal y el campo de autor de las tareas que haya creado. De este modo, se garantiza la eliminación de sus datos personales manteniendo el contenido aportado, el cual podrá seguir siendo útil para el resto de usuarios.

Las actividades y tareas están organizadas por roles, por lo que solo se visualizan las correspondientes al rol seleccionado. El usuario puede elegir qué rol desea consultar (solo si forma parte), y en función de esta selección, la pantalla principal se actualizará para mostrar únicamente las actividades y tareas asociadas a dicho rol.

Las tareas contienen los siguientes campos (Fig. 4):

- **Título y descripción**, los cuales son obligatorios.
- **Periodo de realización**, el cual está predefinido por el administrador.
- **Fecha de realización**, la cual indica cuando debe realizarse la tarea. Esta cuenta con el formato DD/MM ya que para que sea válido en cualquier curso académico no debe de contener año de realización.
- **Archivos y enlaces**, estos cuentan con un título, el cual se deberá de completar obligatoriamente, y una descripción. Se pueden añadir un máximo de diez archivos y diez enlaces por tarea.

Fig. 4: Página de creación de nueva tarea

Cabe destacar que una tarea no puede tener al mismo tiempo un periodo y una fecha de realización, de manera que la propia plataforma en el momento de selección de uno de los dos desactivará el campo contrario.

La tarea, al ser publicada, muestra la fecha de creación, la fecha de modificación (en el caso de que no haya sido modificada esta será la misma que la de creación) y el autor de la tarea. Este último solamente se mostrará si el usuario que creó la tarea sigue dado de alta en la plataforma; en caso de ser dado de baja, dicho campo simplemente se omitirá.

Todas las tareas cuentan con una sección de comentarios en la que cualquier usuario puede añadir anotaciones o plantear dudas. Estos comentarios pueden ser públicos o privados, y disponen de un campo de texto obligatorio, además de la posibilidad de adjuntar archivos. Los archivos seguirán el mismo formato que en el caso de las tareas, incluyendo campos de título y descripción, con un límite máximo de diez archivos por comentario. Estos comentarios (Fig. 5) se muestran con la fecha de publicación, el usuario que lo ha realizado y, en el caso de que el comentario sea privado, un icono que así lo indique.

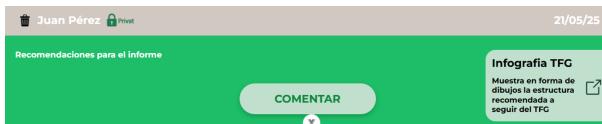


Fig. 5: Comentario privado

La plataforma incluye una pantalla de perfil de usuario, donde se muestra su información de contacto (nombre, NIU, correo corporativo y cargo académico), así como los roles que tiene asignados, indicando en cada caso si actúa como administrador del rol. Además, se dispone de un botón de 'Solicitud de rol' que permite abrir una ventana de texto desde la cual el usuario puede enviar una petición al administrador para ser añadido o eliminado de un rol.

En la *navbar* se encuentra una barra de búsqueda que permite localizar tareas por coincidencia en el título. Los resultados mostrados indicarán a qué actividad pertenece cada tarea y el rol al que están asociadas.

Por último, la herramienta cuenta con un apartado llamado "Calendari", en el que se pueden filtrar las tareas de forma temporal. Esto permite al usuario consultar qué tareas debe realizar en el espacio de tiempo seleccionado. Esta funcionalidad solo muestra tareas que cuenten con un periodo o una fecha de realización.

Dispone de tres filtros distintos: periodo de realización, donde permite seleccionar uno de los períodos definidos por el administrador, intervalo temporal, el cual permite especificar un rango de fechas concreto seleccionado por el usuario, y los roles a visualizar, que determina qué roles se incluirán en el filtrado.

La selección de roles solo estará disponible una vez se haya activado uno de los dos filtros temporales. Además, no es posible seleccionar simultáneamente un periodo y un

intervalo, con el objetivo de evitar conflictos de uso.

Una vez aplicados los filtros y presionado el botón de filtraje, se mostrarán todas las tareas cuyo periodo o fecha de realización coincidan con el intervalo o periodo seleccionado, junto con la información básica de cada tarea: título, rol asignado, usuario que la creó, actividad a la que pertenece y fecha o intervalo de realización.

Tanto en el caso de la barra de búsqueda como en el filtrado temporal, las tareas resultantes son seleccionables, lo que permite acceder directamente a la pantalla de detalles de la tarea.

## 8 RESULTADOS OBTENIDOS

A continuación, se presentan los objetivos planteados con su correspondiente resultado, analizando su grado de cumplimiento y justificando si han sido alcanzados o no.

### 8.1. Resultados funcionales

- *Facilitar el traspaso de información entre diferentes cargos, reduciendo así el tiempo de adaptación:* Se considera que se ha conseguido lograr este objetivo. A pesar de tener diferentes cargos, por ejemplo un profesor de una asignatura y un responsable de una mención, pueden optar a los mismos roles, lo cual les permitirá llegar a la misma información. También, en el caso de ser nuevo en el cargo o que le asignen un nuevo rol, no parte de cero, ya que al asignarle ese rol la herramienta detecta que forma parte de él y muestra las actividades y tareas correspondientes al rol. De esta manera puede acceder a toda la información del rol desde el primer momento de la incorporación, conociendo las tareas y actividades (Fig. 6) que han sido introducidas por el resto de compañeros, partiendo de una base mucho más amplia de información desde la primera toma de contacto.

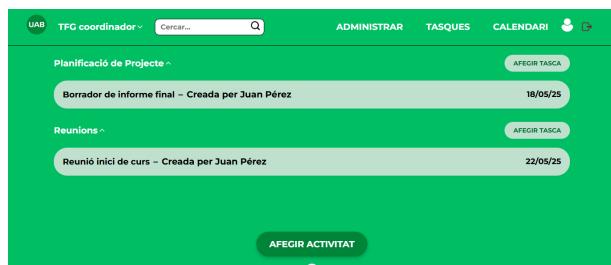


Fig. 6: Estructura de actividades y tareas

- *Optimizar el seguimiento de tareas a realizar, reduciendo así los errores humanos a causa de olvidar tareas o falta de información:* Se considera que se ha cumplido con el objetivo. Se pueden conocer las tareas que se deben de hacer según el rol (Fig. 6), facilitando el objetivo de no olvidarse de alguna tarea o información importante. Además, con el apartado relativo al calendario (Fig. 7), por el cual se pueden filtrar las tareas temporalmente, se pueden observar con más facilidad las tareas a hacer en un periodo de tiempo

próximo, conociendo la fecha límite para hacerlas en ese momento, como por ejemplo lo que se debe realizar ese mes, semana o periodo, haciendo más sencillo que no se olviden las tareas a realizar.

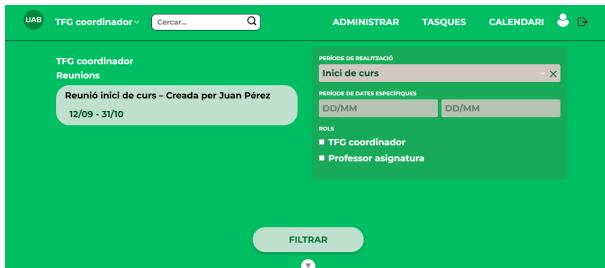


Fig. 7: Funcionalidad de Calendario

- *Redistribuir las tareas de un rol a otro de manera rápida y sencilla:* Este objetivo se decidió dejar como línea futura de la herramienta web, puesto que compartir tareas se consideró que no era tan esencial y realmente era añadir una complejidad extra que se realizaría en el caso de que hubiese tiempo adicional, el cual finalmente no se ha acabado dando.
- *Mejorar el rendimiento en la gestión de tareas y responsabilidades académicas:* Se considera el objetivo alcanzado, puesto que esta herramienta, al permitir añadir tareas y actividades y además desglosar las responsabilidades en forma de roles facilita mucho la gestión de tareas y responsabilidades, ya que están ordenadas por rol y dentro de ese rol además se desglosan en actividades, teniendo así una vista más concreta de las tareas que se deben hacer. Esto es posible debido a las funcionalidades de añadir actividad y añadir tarea (Fig. 4) implementadas en la herramienta.
- *Compartir información de buenas prácticas entre responsables académicos que ocupan los mismos cargos en distintos ámbitos:* Esto es posible si los responsables así lo desean, gracias a la sección de comentarios (Fig. 8), donde pueden formular preguntas, compartir buenas prácticas o publicar recomendaciones de forma directa. Además, tienen la opción de adjuntar archivos que complementen la información. Dado que esta funcionalidad está completamente desarrollada y operativa, se considera que el objetivo se ha cumplido satisfactoriamente.



Fig. 8: Apartado de comentarios

## 8.2. Resultados de desarrollo personal

- *Mejorar las habilidades de desarrollo web:* Sin duda este objetivo se ha cumplido. En la carrera se llevó a cabo una página web con HTML y PHP, pero cuando tocó aprender Node.js y Vue.js lo que realmente se realizó fue un juego, y sinceramente siempre había quedado la duda de cómo se desarrolla una página web convencional (como las que se usan en el día a día) con estas herramientas, por eso se decidió utilizarlas para desarrollar este proyecto. A día de hoy se han adquirido conocimientos que en su día no se adquirieron, como utilizar rutas para ir a las diferentes pantallas de la página web, o incluso el uso de padres e hijos en estas rutas, conociendo mucho mejor la forma de estructurar el frontend con Vue.js. Para Node.js se destaca sin duda el aprendizaje de procesar archivos, lo cual siempre ha parecido bastante complejo, utilizar variables de sesión como las cookies o cosas más sencillas como establecer conexiones con la base de datos y hacer peticiones.
- *Mejorar las capacidades de planificación y cumplimiento de ésta:* Este objetivo no se ha alcanzado por completo, ya que no se logró ejecutar la planificación del proyecto de manera estricta. La elevada carga de trabajo externo, ajena al desarrollo de esta herramienta, dificultó considerablemente el proceso de organización y distribución del tiempo. No obstante, esta experiencia ha resultado especialmente valiosa a nivel personal y profesional, ya que ha permitido identificar errores importantes de gestión que servirán como aprendizaje para el futuro. Se ha comprendido la importancia de establecer prioridades claras, incluso cuando los recursos temporales son limitados, así como la necesidad de mantener una visión realista sobre las propias capacidades. A pesar de las dificultades, el proceso ha demostrado que, con determinación y esfuerzo, es posible avanzar en múltiples tareas, por muy exigentes que puedan parecer.
- *Experimentar cada uno de los procesos de un proyecto (recopilación de requisitos, creación de la base de datos, desarrollo de Backend y Frontend y testeo de la página) de forma seguida, desde cero:* Se considera que el objetivo está cumplido, ya que se ha logrado experimentar de forma completa y ordenada cada una de las fases que conforman el desarrollo de un proyecto software desde cero. El proceso comenzó con la recopilación de requisitos, lo cual permitió establecer una base sólida sobre la que construir el resto del sistema. A partir de esta información, se diseñó y creó la base de datos, definiendo su estructura en función de las necesidades identificadas. Posteriormente, se abordó el desarrollo del frontend, siguiendo con el desarrollo del backend y finalmente, se llevó a cabo un proceso de testeo que permitió verificar el correcto funcionamiento del sistema y garantizar la coherencia entre todas las partes desarrolladas. Este último proceso estando presente al final del desarrollo de cada una de las funcionalidades.

## 9 CONCLUSIONES

Tras finalizar el desarrollo de la herramienta web de soporte a responsables académicos, se puede decir que el resultado ha sido satisfactorio. Se ha conseguido construir una plataforma funcional que, aunque es una primera versión, cumple con los objetivos marcados desde el inicio. Está pensada para facilitar la gestión de tareas y mejorar la coordinación entre distintos cargos, por lo que puede aportar un valor real dentro del entorno universitario si se decide implementarla.

Llevar a cabo un proyecto de este tipo de forma individual no ha sido fácil. Ha hecho falta aprender a organizarse bien, especialmente en los momentos en los que coincidía con otras responsabilidades. Eso ha hecho que no siempre se pudiera seguir la planificación inicial al pie de la letra, pero también ha servido para aprender a priorizar mejor y a ser más realista con los tiempos.

Otro aspecto positivo ha sido trabajar sobre un proyecto abierto, sin estar completamente definido desde el principio. A medida que el desarrollo avanzaba, iban surgiendo nuevas ideas y funcionalidades que se han podido añadir, lo que ha hecho que el proyecto evolucionara de forma natural y mantuviera el interés hasta el final. También ha sido clave mantener cierta flexibilidad para poder adaptarse sin desviarse de los objetivos principales.

También conviene señalar algunas líneas de trabajo futuras que podrían contribuir a mejorar la herramienta. En primer lugar, la implantación definitiva en el servidor institucional que permitiría poner en marcha la plataforma y empezar a obtener los beneficios que se han buscado desde el inicio del proyecto.

Asimismo, se propone como mejora la incorporación de un sistema de notificaciones. Esta funcionalidad permitiría alertar a los usuarios cuando se publique una nueva actividad o tarea, cuando se acerque la fecha o período de realización de una tarea, cuando se resuelvan sus peticiones de rol o cuando se añada un comentario en alguna de las tareas visibles para su perfil. La inclusión de este sistema no solo mejoraría la comunicación dentro de la plataforma, sino que también reforzaría el seguimiento activo de las responsabilidades asignadas a cada usuario.

En resumen, ha sido una experiencia muy completa que ha permitido aplicar lo aprendido durante la carrera, adquirir nuevos conocimientos técnicos y, sobre todo, entender cómo se desarrolla una herramienta web desde cero hasta tener algo funcional.

## AGRADECIMIENTOS

Me gustaría agradecer principalmente a mi tutor, quien me ha echado una mano siempre que lo he necesitado y me ha tenido paciencia cuando las cosas no salían como se habían planeado.

También agradecer a mi familia por apoyarme en momentos de estrés, cuando todas las responsabilidades se me

venían encima y simplemente necesitaba desconectar para volver a reconectar.

A mis amigos, especialmente aquellos que he conocido durante estos cuatro años de carrera. Su apoyo ha sido indispensable para mí en todo momento: compartiendo los mismos problemas, enseñándonos lo que no lográbamos comprender, riéndonos de nuestra mala gestión y conociéndonos en este camino tan importante.

Y, finalmente, a mi pareja, quien me ha apoyado en todo momento y con quien he trabajado codo a codo. Gracias por recordarme que soy capaz de grandes cosas y que los límites solo están en mi cabeza.

## REFERENCIAS

- [1] Asana, T. (2025) 'Las 12 metodologías más populares para la gestión de proyectos [2025] • Asana', Asana, 19 febrero. <https://asana.com/es/resources/project-management-methodologies> (Accedido: 27 de febrero de 2025).
- [2] Variaciones comunes de modelos de cascada (sin fecha). <https://fastercapital.com/es/tema/variaciones-comunes-de-modelos-de-cascada.html> (Accedido: 27 de febrero de 2025).
- [3] Como crear relaciones entre tablas SQL, (sin fecha). Stack Overflow. Disponible en: <https://es.stackoverflow.com/questions/84613/como-crear-relaciones-entre-tablas-sql> (Accedido: 14 de abril de 2025)
- [4] ¿Qué es Vue Router y Por qué es Necesario?, (sin fecha). Cursos para Empresas — Cursos Bonificados FUNDAE. Disponible en: <https://imagineformacion.com/tutoriales/que-es-vue-router-y-por-que-es-necesario> (Accedido: 23 de mayo de 2025)
- [5] JuanDavid\_Dev, (2021). Cómo crear una navbar responsive con HTML y CSS Flexbox. Youtube. Disponible en: <https://www.youtube.com/watch?v=jeQiN0gPIZw> (Accedido: 17 de abril de 2025)
- [6] Nested Routes — Vue Router. (sin fecha). Vue Router — The official Router for Vue.js. Disponible en: <https://router.vuejs.org/guide/essentials/nested-routes> (Accedido: 6 de mayo de 2025)
- [7] Moreno, K., (2022). Introducción a los archivos .env. DEV Community. Disponible en: <https://dev.to/khriztianmoreno/introduccion-a-los-archivos-env-3e2f> (Accedido: 13 de mayo de 2025)
- [8] Dayanand, A., (2023). Simple Steps for Securely Inserting Data into SQL — Best Practices [en línea]. Medium. Disponible en: <https://blog.devgenius.io/securely-insert-data-into-sql-4f98daafdf0> (Accedido: 22 de mayo de 2025)

## APÉNDICE

A continuación se mostrarán las pantallas de la herramienta que no han sido mostradas en el informe.

UAB TFG coordinador de

ADMINISTRAR TASQUES CALENDARI

**RESULTATS DE LA CERCA**

**TFG coordinador**

**Planificació de Projecte**

**Borrador de informe final – Creada per Juan Pérez** 19/5/2025

**Reunions**

**Reunió inici de curs – Creada per Juan Pérez** 22/5/2025

Fig. 9: Resultados de una búsqueda

UAB TFG coordinador de

ADMINISTRAR TASQUES CALENDARI

**MODIFICAR TASCA**

← Nom de la tasca \* Reunió inici de curs Data de realització (DD/MM) DD/MM

Descripció \* Primer contacte del curs Període de realització Inici de curs

**AFEGEIX ARXIU** EngInformatica.pdf ELIMINAR PERÍODE

**AFEGEIX ENLLAÇ** Primer afegeix un enllaç &+

**Arxiu sobre les bases de l'enginyeria informàtica**

**ELIMINAR ARXIU** MODIFICAR TASCA

Fig. 10: Funcionalidad de modificar tarea

UAB TFG coordinador de

ADMINISTRAR TASQUES CALENDARI

**Juan Pérez**

**NIU: 1631092**  
**ADREÇA: 1631092@uab.cat**  
**CÀRREC ACADÈMIC: Profesor asociado**

**ROLS**

TFG coordinador Administrador

Professor asignatura

**SOL-LICITAR ROL**

Fig. 11: Apartado de perfil de usuario

Borrador de informe final - Creada per Juan Pérez

19/5/2025

Creada el 18/5/2025

Asegurar-se de que es preparari i lliurari el borrador de informe final a temps per a la data de realització definida

MODIFICAR

ELIMINAR

Enllaços

Arxius

Comentaris

Data de realització: 05/06  
Període: No hi té

Fig. 12: Visualización de la tarea

ENLLAÇOS

Web del grau

URL on està tota l'informació sobre el grau

Fig. 13: Apartado de enlaces asociados a la tarea

ARXIUS

Bases de la enginyeria informàtica

Diagrama amb les característiques principals de l'enginyeria informàtica

Fig. 14: Apartado de archivos asociados a la tarea

Fig. 15: Funcionalidad de gestión de solicitudes del Administrador

Fig. 16: Funcionalidad de gestión de períodos del Administrador

Fig. 17: Funcionalidad de gestión de usuarios del Administrador

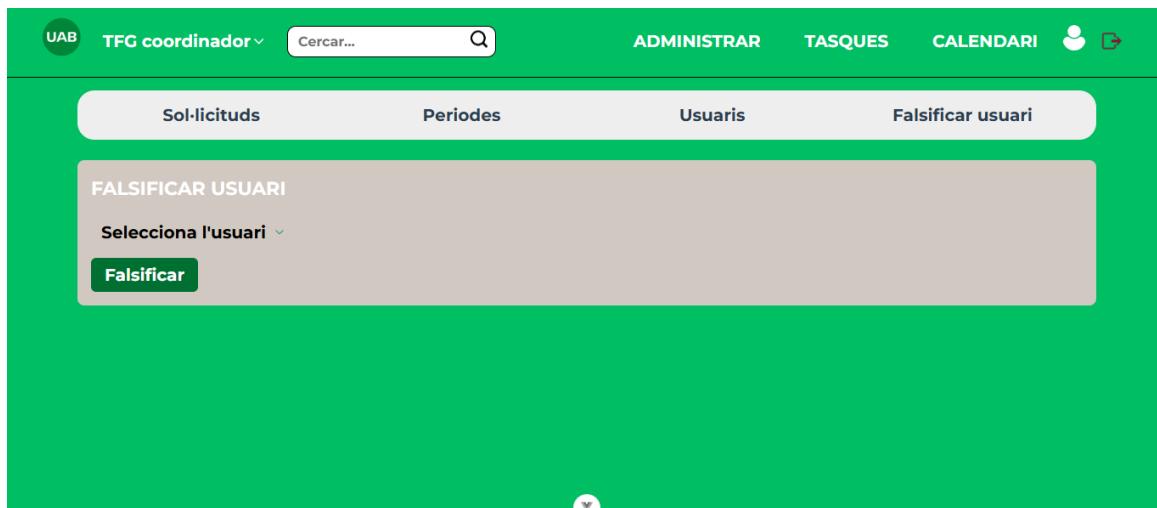


Fig. 18: Apartado de falsificar usuario del Administrador