LICEO LA PAZ

PROYECTO FIN DE CICLO

C.P.R LICEO LA PAZ

DESARROLLO DE APLICACIONES MULTIPLATAFORMA

NEREA PENA CARBAJALES

TUTOR: JESÚS ÁNGEL PÉREZ ROCA FERNÁNDEZ

Resumen

TaskFlow es un proyecto de gestión de tareas que tiene como objetivo principal ofrecer un programa digital, simple, intuitivo y fácil de usar que permite a sus usuarios gestionar sus tareas de manera eficaz. La aplicación ofrecerá funcionalidades como la creación de listas de tareas, asignación de categorías y determinar un estado para ellas.

Abstract

TaskFlow is a task management project with the primary objective of offering a digital program that is simple, intuitive and easy to use, allowing its users to effectively manager their task. The application will provide functionalities such as task list creation, categorization assignment, and status determination.

Palabras Clave

- **Gestión de tareas:** El proyecto se enfoca en mejorar la organización, seguimiento y finalización de las tareas.
- **JavaFX:** Se emplea *JavaFX*, un conjunto de bibliotecas y herramientas de interfaz de usuario de Java, para desarrollar correctamente la interfaz grafica necesaria para el programa.
- **Scene Builder:** Ofrece una herramienta visual de diseño de interfaces que permite crear y personalizar la apariencia del programa.
- **Base de datos (MySQL):** Se emplea una base de datos para guardar y recuperar información relacionada con las tareas, tales como las categorías, los usuarios que tienen tareas o información detallada de las mismas.

A Martin <3

Sumario

Introducción/motivación

Elegí realizar este proyecto de gestión de tareas porque identifiqué una necesidad en mi forma de trabajar de crear mi propia aplicación de gestión de tareas en la que pudiese añadir lo que yo mas necesito.

Me di cuenta que otras aplicaciones de gestión de tareas tenían muchísimas funcionalidades, pero cuando yo hacia uso de ellas, sentía que la mayoría de esas funcionalidades no les veía un uso.

Al embarcarme en este proyecto, considere mis necesidades y que funcionalidades eran las que mas empleaba yo y cuales no necesitaba en mi día a día. Sabia que al implementar todas mis ideas en este proyecto crearía una aplicación en la que yo me sentiría cómoda y estaría aprovechando el 100% de sus funcionalidades.

En cuanto a objetivos que quiero lograr con este proyecto es mejorar la productividad de la gente que usa esta aplicación y la eficiencia de los mismos. Quiero proporcionar una herramienta sencilla y efectiva que ayude a optimizar el tiempo que se le dedica a asignar las tareas, gracias a que contiene lo necesario para su uso perfecto.

Quise crear una interfaz intuitiva y fácil de usar para así ofrecer una experiencia agradable al usuario. Como es costumbre en mis proyectos, los colores pastel y el minimalismo perduran en *TaskFlow*, permitiendo el uso efectivo por parte de cualquier usuario.

En resumen, mi objetivo principal con este proyecto es desarrollar una herramienta eficiente y personalizada que ofrezca una buena productividad en la ejecución de las tareas y brindando una mayor visibilidad y control sobre el progreso de los objetivos.

Estado del arte

Como he mencionado antes, *TaskFlow* es una aplicación que se centra en brindar a sus usuarios una experiencia de uso con el 100% de funcionalidades operativas.

Existen varias aplicaciones de gestión de tareas en el mercado. Se mencionan algunas de ellas junto con sus puntos fuertes y débiles en comparación con *TaskFlow:*

- ***Trello***.
  - Puntos fuertes: *Trello* es una herramienta popular y ampliamente utilizada que permite crear tableros y listas para organizar tareas. Es muy visual y fácil de utilizar, generándole accesible para diferentes usuario.
  - Puntos débiles: En su versión gratuita tiene ciertas limitaciones en cuanto a la cantidad de tableros y funciones disponibles.
- ***Todoist***.
  - Puntos fuertes: *Todoist* es una aplicación sencilla y elegante que se enfoca en la simplicidad y facial de uso. También ofrece sincronización en tiempo real en múltiples dispositivos.
  - Puntos débiles: Resulta menos adecuada para proyectos de equipo complejos,
- ***Asana***.
  - Puntos fuertes: *Asana* ofrece funciones como el seguimiento del progreso de las tareas, fechas limite y generación de informes. Es una herramienta muy compleja que se adapta a proyectos de diferentes tamaños.
  - Puntos débiles: Algunas funciones requieren una suscripción de pago y en muchos casos la curva de aprendizaje para los usuarios menos experimentados puede ser difícil.

En cuanto a tecnologías empleadas, *TaskFlow* utiliza *JavaFX* y *Scene Builder* para desarrollar una interfaz grafica de usuario. Estas tecnologías permiten crear una interfaz visualmente atractiva y funcional, junto con el trabajo de bases de datos para el almacenamiento de datos relacionados con las tareas.

A diferencia de las aplicaciones mencionadas, *TaskFlow* se centra unicamente en que el usuario pueda aprovechar el 100% de la aplicación, unicamente utilizando las funcionalidades que considere necesarias.

Caso de estudio

Para solucionar los problemas en los casos presentados y mejorarlos al beneficio del usuario, se desarrollarían unos puntos claves en la aplicación.

- **Interfaz intuitiva:** Se plantea trabajar con una interfaz intuitiva y fácil de usar, que permite a los usuarios trabajar de manera efectiva con la aplicación y poder realizar de manera efectiva todas las funcionalidades que ofrece la aplicación. En esta aplicación se le prestará especial atención a la organización de los elementos de la interfaz para facilitar la navegación y comprensión de la aplicación.

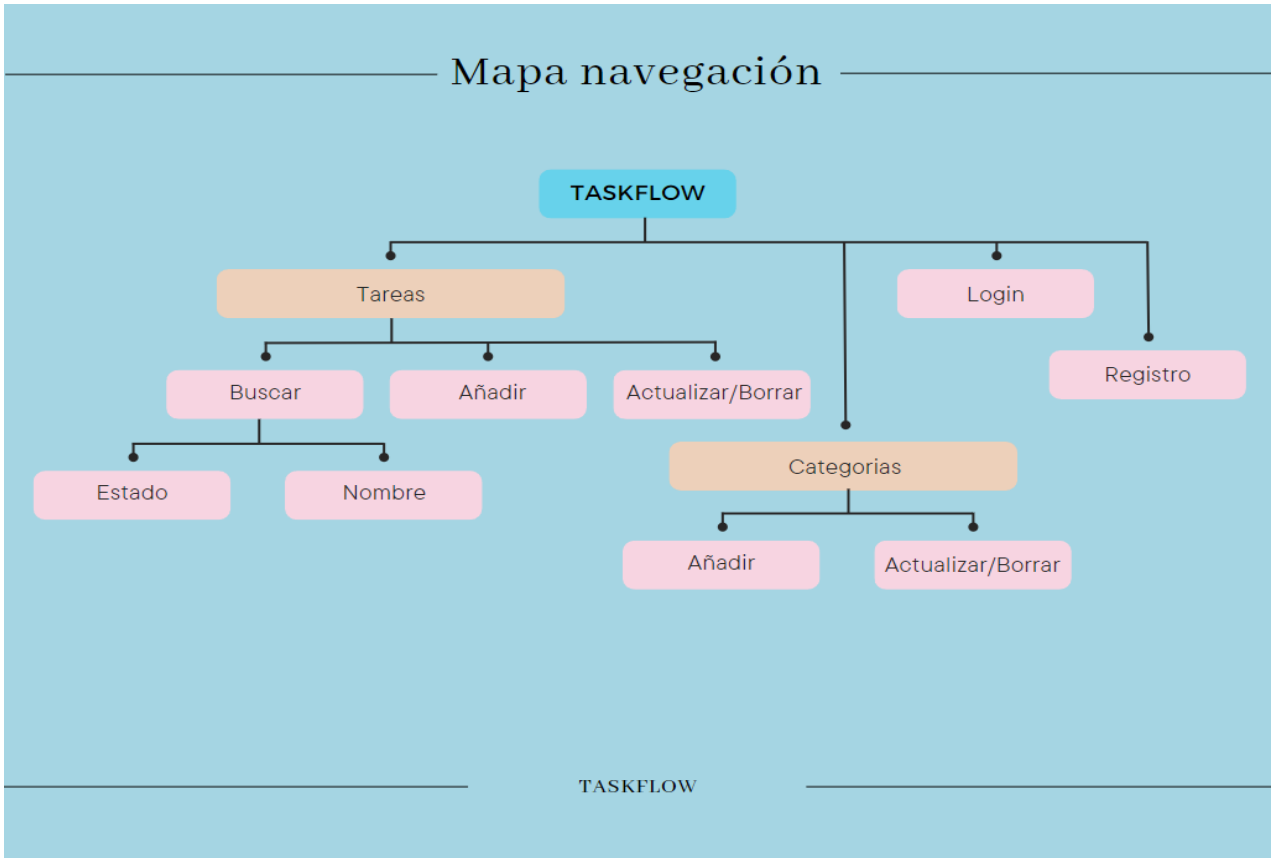
- **Creación de tareas:** Se implementara una funcionalidad para crear nuevas tareas y asignarles información relevante como su descripción o la categoría a la que pertenecen. Ademas el usuario podrá buscar en tiempo real cualquier tarea que haya creado.

- **Seguimiento del progreso:** Se desarrollara una función que permita realizar un seguimiento del progreso de cada tarea. Los usuarios podrán marcar las tareas como *“en progreso”*, *“pendientes”* o *“sin completar”*, lo que brindará una versión clara del estado actual de las tareas.

Al desarrollar estos puntos clave, se pretende mejorar la experiencia en *TaskFlow,* ofreciendo una solución completa y eficiente. *TaskFlow* se compromete a cumplir las necesidades de aquellas personas que valoran la funcionalidad y quieren las cosas claras a la hora de organizarse.

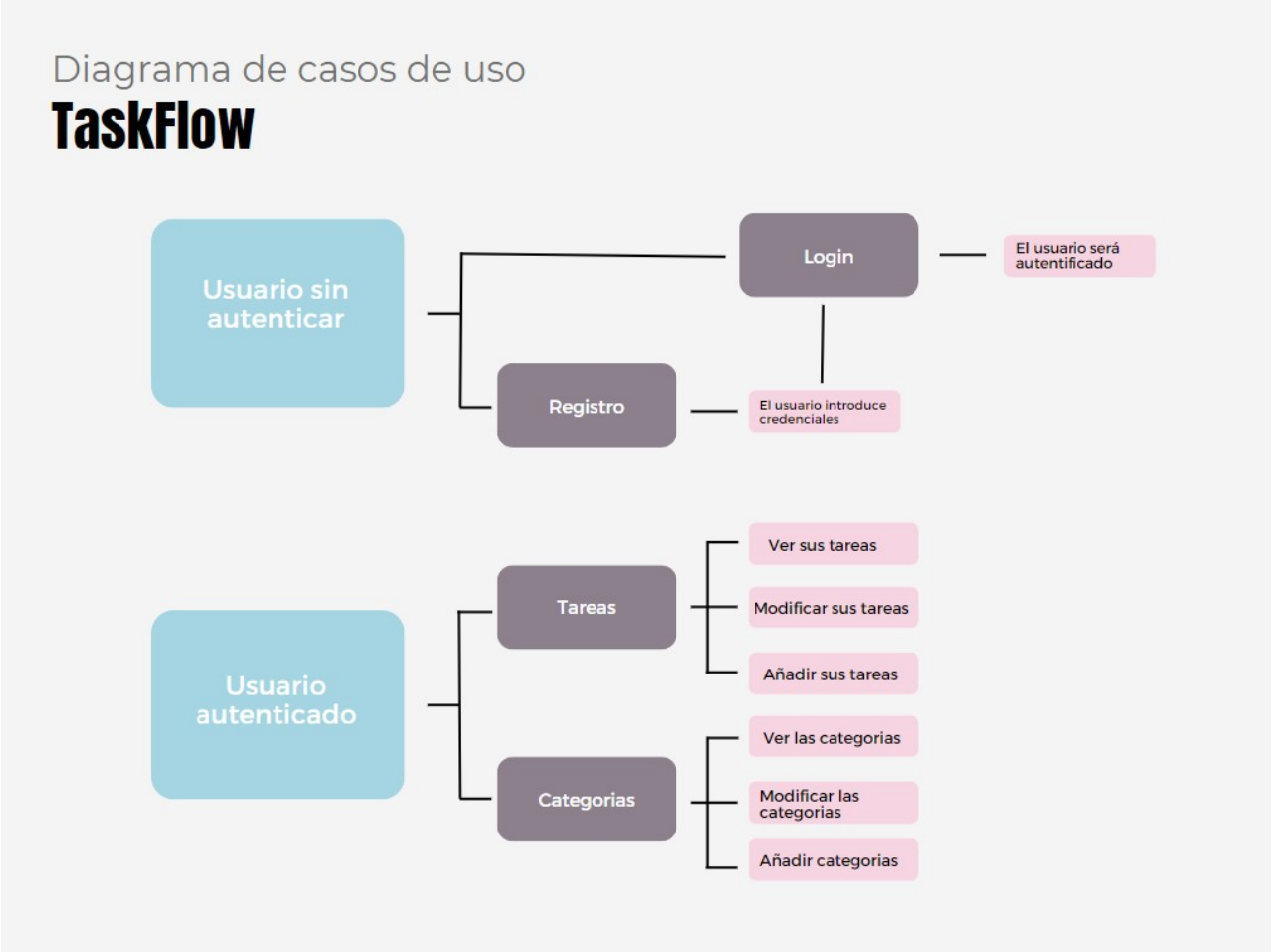
Diagramas

DIAGRAMA DE NAVEGACIÓN



Mapa de navegación en el que se presentan la estructura y organización de la aplicación. En este mapa se muestra de manera clara como se organizan las distintas ventanas (escenas) de la aplicación.

DIAGRAMA CASOS DE USO

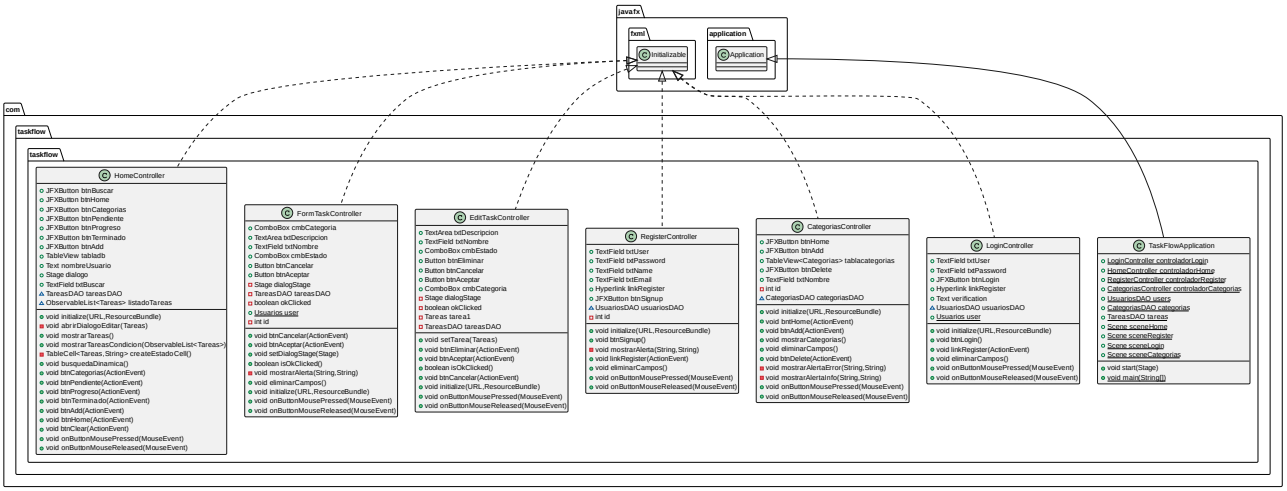


El diagrama de casos de uso aquí presentado, representa las interacciones entre los diferentes actores y la aplicación. En este diagrama, los actores principales son los usuarios autenticados y sin autenticar.

Un usuario sin autenticar no tendrá la posibilidad de acceder por completo a la aplicación, unicamente podrá iniciar sesión en la aplicación. En el caso de que no se tuviese un usuario, se le permitirá ir a una sección de registro, en la que tras introducir datos, se le podrá dar permiso para iniciar sesión y entrar en la aplicación.

Un usuario autenticado es aquel que tras iniciar sesión, puede acceder a la aplicación y disfrutar de sus funcionalidades. Es importante dejar claro que un usuario solo podrá ver y modificar sus tareas, permitiendo así que cada usuario tenga su espacio. En cambio, un usuario podrá ver todas las categorías de la aplicación y podrá modificarlas y añadir mas si cree conveniente, siendo añadidas para los demás usuarios.

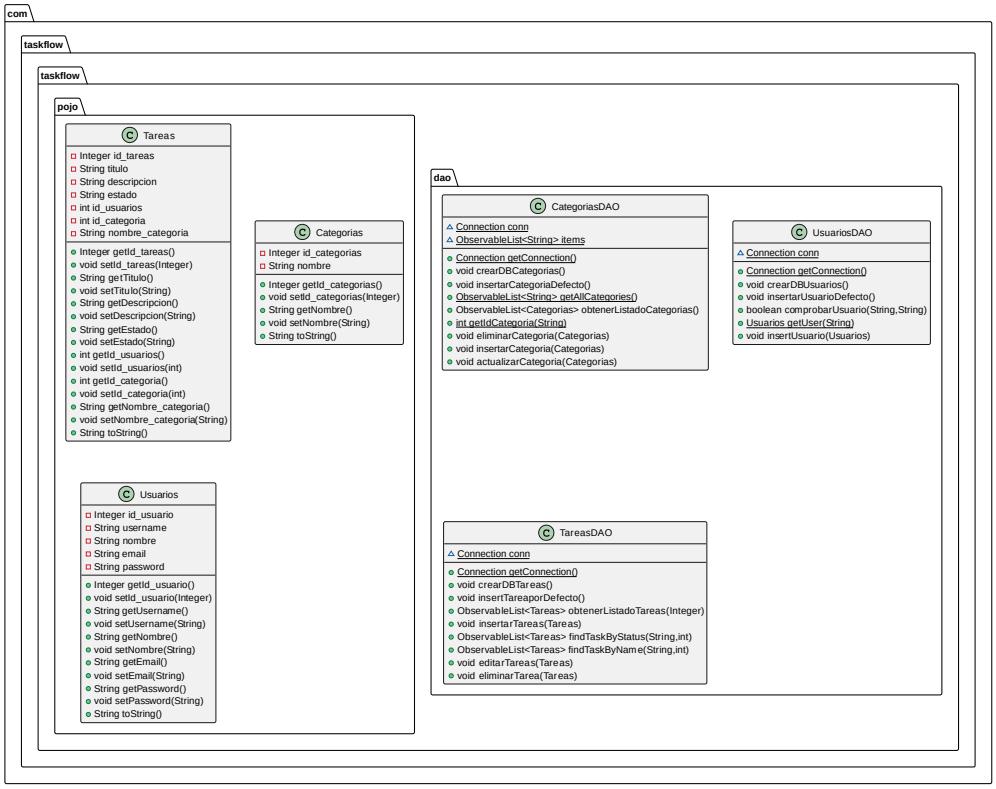
DIAGRAMA DE CLASES



El diagrama de clases representa la estructura y las relaciones entre las clases en un sistema orientado a objetos. En estos diagramas, se muestran las clases principales del programa y sus métodos asociados.

En la primera imagen se puede apreciar un diagrama de las clases *Controller* junto con la clase ejecutable.

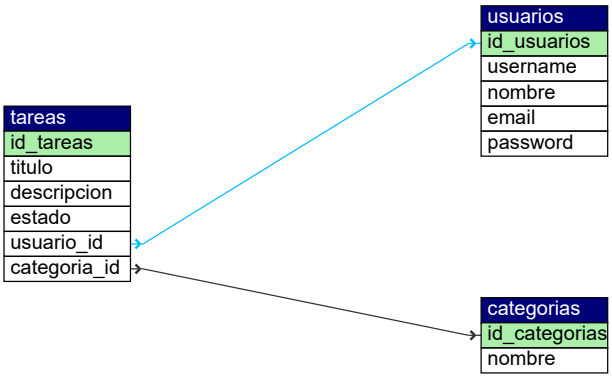
- Las clases *Controller* son las responsables de manejar la lógica de la interfaz de usuario y coordinar las interacciones entre la vista y la clase *DAO* en la aplicación. Las clases *Controller* se asocian con la clase de vista correspondiente, lo cual nos indica que el controlador es el responsable de manejar eventos y las acciones relacionadas con la vista. Los métodos de la clase *Controller* se comunican con la vista para actualizar la interfaz de usuario y con la clase *DAO* para realizar operaciones con los datos.
- Por otro lado, la clase *Application* representa la entrada principal de la aplicación y es la encargada de inicializar y configurar los componentes del sistema. Esta clase puede tener asociaciones con otras clases del programa, como las clases *Controller*, las clases *DAO*. Los métodos de la clase *Application* representan los métodos de arranque de la aplicación como start( ).



La segunda imagen contiene las clases *POJO* y las clases *DAO:*

- Las clases *POJO* representan los objetos de nuestro programa (tareas,categorías y usuarios). En cada caja se muestran los atributos de cada clase, que representan los datos que queremos almacenar en nuestro objeto. Cada atributo se muestra como una variable con su nombre y para acceder a esos atributos agregaremos los métodos *getter* (devuelven el valor del atributo) y *setter* (establecen un nuevo valor para el atributo).
- Las clases *DAO* son las clases responsables en interactuar como si fueran una base de datos. Las clases *DAO* tienen una asociación con las clases *POJO* correspondientes, lo que quiere decir que el *DAO* se encarga de manejar los objetos de esas clases en particular. Ademas las clases *DAO* contienen los métodos para realizar operaciones de persistencia, recuperación, actualización y eliminación de datos en la base de datos.

DIAGRAMA BASE DE DATOS

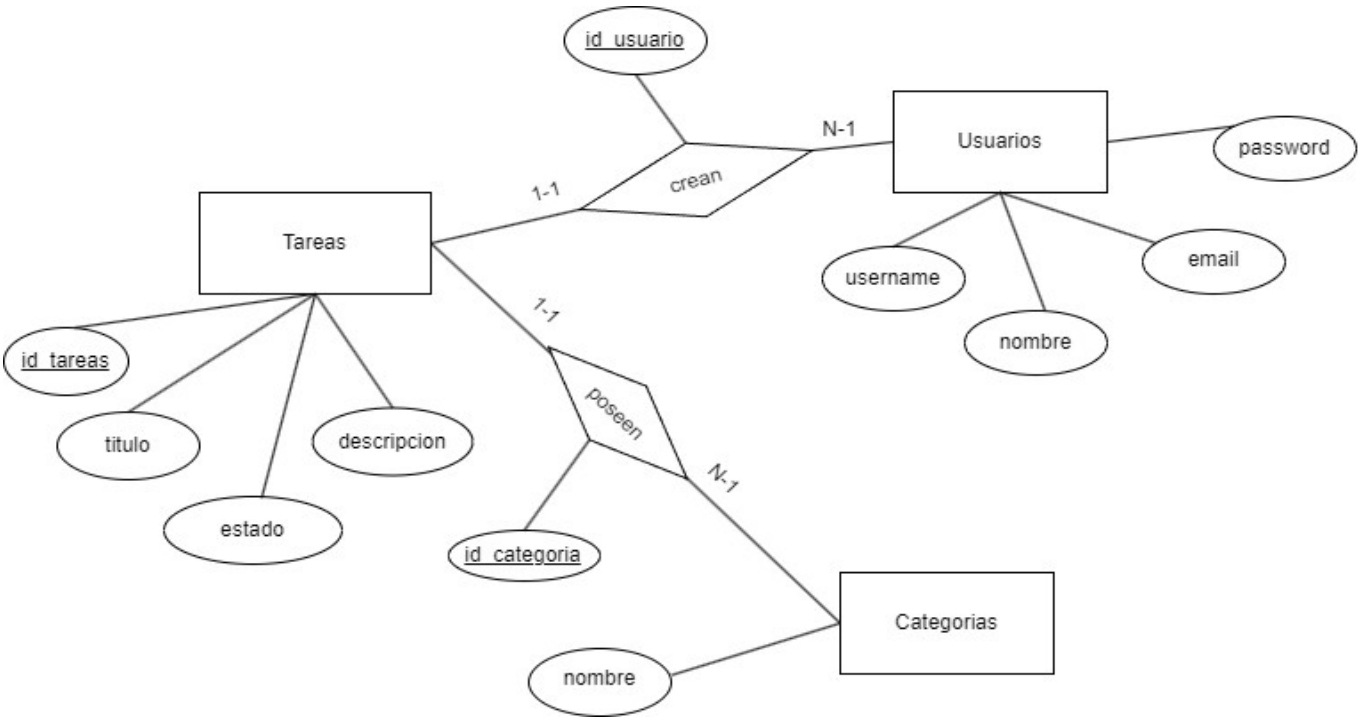


Un diagrama de base de datos es una representación visual de la estructura y relaciones de la base de datos. Un correcto diagrama nos proporciona una visión clara de como se organizan y conectan las tablas, las entidades y atributos.

Los elementos clave de este diagrama de base de datos son:

- **Tablas**: Las tablas son las entidades principales en la base de datos. En este diagrama se observan tres tablas (***tareas***,***usuarios*** y ***categorías***).
- **Relaciones**: Nos muestran como se relacionan las tablas entre si. Por ejemplo, en este diagrama, una tarea puede estar asignada a un usuario y un usuario puede tener muchas tareas (*relación 1-N*) o una categoría puede tener múltiples tareas (*relación 1-N*).
- **Claves primarias:** Nos permiten identificar de manera única cada registro en una tabla. Se representan resaltando la columna que actuá como clave primaria.
- **Claves foráneas:** Son las encargadas de representar las relaciones entre las tablas. Vincula una tabla con otra mediante una columna en común. Por ejemplo, la columna ***usuario_id*** de la tabla tareas se comunica con ***id_usuarios*** en la tabla usuarios.

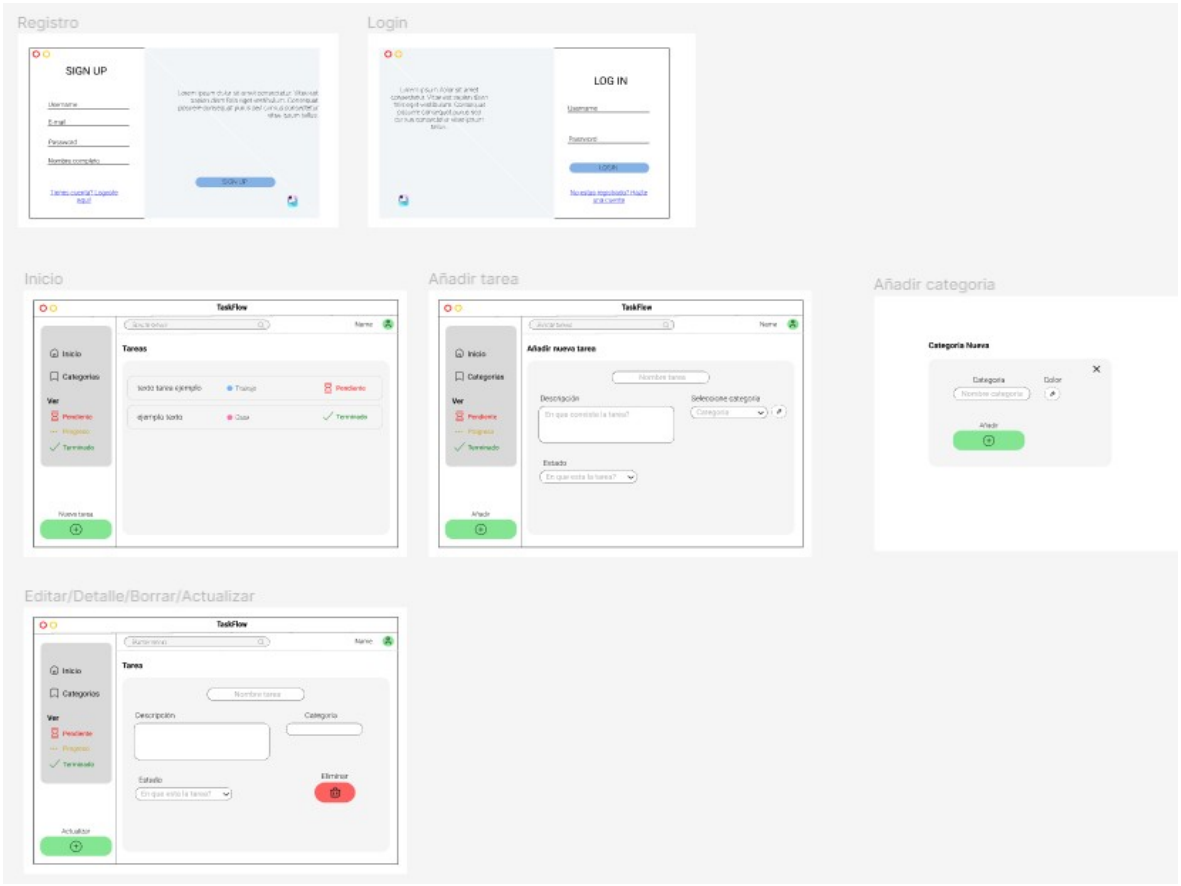
DIAGRAMA ENTIDAD-RELACIÓN



Un diagrama entidad relación (*E-R*) es una representación visual de las entidades, atributos y las relaciones entre ellas.

- **Entidades**.
  - **Principales**.
    - **Tareas**: se representan a las tareas de la aplicación y puede tener atributos como *id_tareas*, *titulo*, *descripción*,*estado*,*id_usuario* y *id_categoría*.
    - **Usuarios**: se representan a los usuarios de la aplicación y tiene atributos como *id_usuario*, *nombre*, *username*, *email* y *password*.
  - **Adicionales.**
    - **Categorías**: se representa la información sobre las categorías posibles, con atributos como *id_categoría* y *nombre*.
- **Relaciones.**
  - **Tareas-Usuarios:** una relación entre Usuarios y Tareas indicando que usuario crea cada tarea.
  - **Tareas-Categorías:** una relación entre Tareas y Categorías que permite clasificar las tareas en diferentes categorías.

WIREFRAMES



Un wireframe es una representación visual de la aplicación en forma de esquema que muestra la estructura y disposición de los elementos en una interfaz de usuario. En este caso, estos wireframes representan la idea original de la aplicación.

Se puede observar como se plantean las diferentes ventanas de inicio, como se distribuyen las diferentes funcionalidades (el buscador, los filtros o el poder añadir las tareas). También se muestra una idea principal del login y registro, aunque posteriormente cambio un poco la distribución.

Aunque muchos de los esquemas no coincidan al 100% con los diseños finales, los wireframes son útiles para empezar a planificar las distintas ventanas y que contendrá cada una.

Desarrollo del proyecto

TaskFlow es una aplicación de gestión de tareas cuya finalidad es ofrecer una aplicación que permita a los usuarios trabajar de manera eficiente y poder organizarse sin malgastar el tiempo. El desarrollo de este proyecto se divide principalmente en varias etapas:

- **Análisis de los requisitos:** Previamente al desarrollo de la aplicación se reunieron una serie de necesidades que debería cumplir la aplicación para llegar a las expectativas. Esos requisitos forman parte de un análisis del mercado en el que se fueron viendo los diferentes puntos fuertes y débiles de otras aplicaciones similares.
- **Diseño de la interfaz:** En esta etapa del desarrollo es cuando la interfaz de usuario se diseña. Se crean pequeños esquemas básicos (*wireframes*) y se plantea una estructura de la aplicación, incluyendo una gran mayoría de las disposiciones de los elementos en la interfaz.
- **Implementación**: Una vez planteada la interfaz se implementaría la lógica de la aplicación. Se crean las clases y los métodos necesarios para gestionar la funcionalidad de la aplicación.
- **Integración de la base de datos:** Tras haber implementado la lógica, se llegaría a la parte en la que se integra la aplicación con la base de datos para almacenar y recuperar información sobre las tareas. Se utiliza *JDBC* para poder interactuar con la base de datos de manera eficiente.
- **Pruebas y mejoras:** Una vez planteado todos los pasos anteriores, se realizarian pruebas en la aplicación para mejorar los posibles errores que se vayan encontrando en ella.

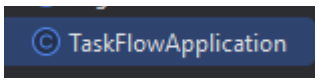
Manual Administrador

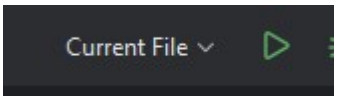
Para un correcto funcionamiento de la aplicación es necesario cumplir con ciertos requisitos previos:

- Tener instalado un gestor de base de datos (puede valer tanto *XAMPP* como *WampServer*, pero en este ejemplo se empleará *WampServer*).
  - Una vez instalado *WampServer* tendremos que lanzar el servidor, que consiste en ejecutar la aplicación.

  - Tras haber lanzado el servidor, en la barra de tareas aparecerá nuestra aplicación que tendremos que abrir con el click derecho y seleccionando p*hpMyAdmin*.

  - Una vez dentro lo único que tendremos que hacer es crear una nueva base de datos de nombre "***gestion_tareas***".
- Es importante tener también instalado ***IntelliJ IDEA***, ya que es un IDE con mucha potencia y con capacidad de buscar actualizaciones disponibles en el caso de que no tengamos en nuestro ordenador el *.jdk* necesario.
- Lanzado el servidor y la base de datos creada lo único que quedaría sería clonar el repositorio y abrirlo con el IDE ***IntelliJ.*** Buscaríamos la clase ejecutable y lanzaríamos el programa.

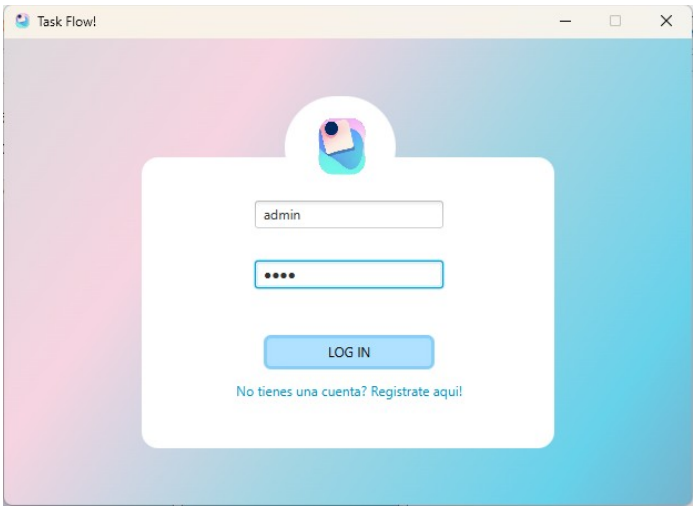




Manual Usuario

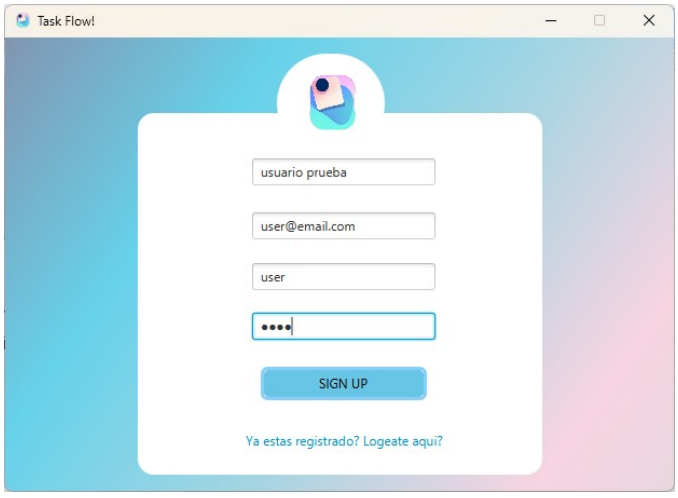
Tras haber explicado como se instala y se pone apunto el programa, se procede a explicar el completo funcionamiento de la aplicación con **casos de prueba**.

Inicializaremos la aplicación en la clase ejecutable ***TaskFlowApplication.java*** y tras hacer eso la aplicación lanzará la ventana de **Login**.

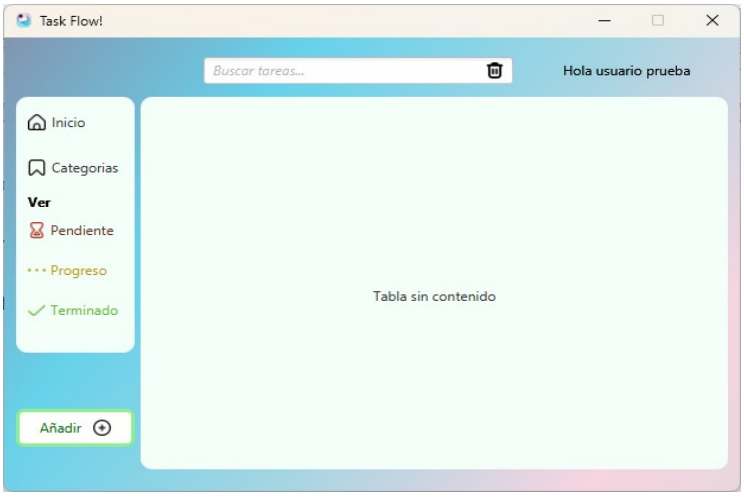


En esta ventana, el usuario si se encuentra registrado, introducirá sus credenciales y el programa procederá a validar si se encuentran en la base de datos, si existe ese usuario en la base de datos, el programa le redirigirá a la ventana de home.

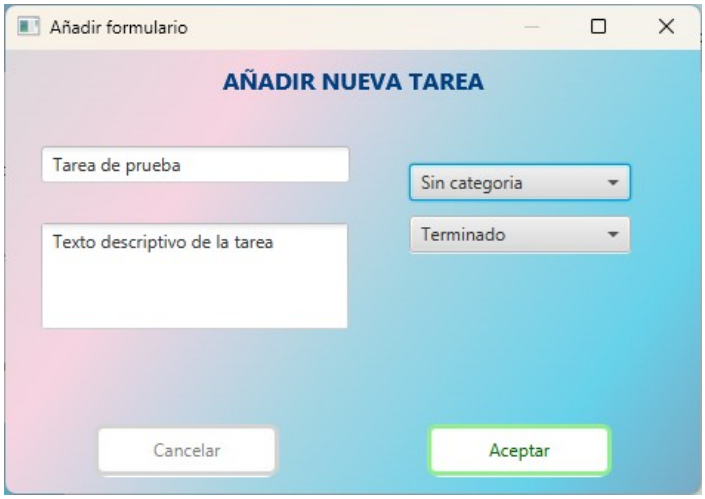
En el caso de que el usuario carezca de credenciales para entrar, en la propia ventana aparece un hipervínculo en el que recomienda al usuario crearse una cuenta. Se le redirigirá a una nueva ventana de ***Registro***, en la que podrá introducir sus datos para poder tener una cuenta de usuario. Una vez aceptado, se le volverá a redirigir a la ventana de ***Login*** para que complete su registro.



Tras haber entrado, la aplicación lo redirigirá a la ventana de **Home** en la que podrá gestionar sus tareas desde ahí. Como ha sido un usuario nuevo, éste carecerá de tareas y por tanto, tendrá que crear una. La ventana de Home se vería así:

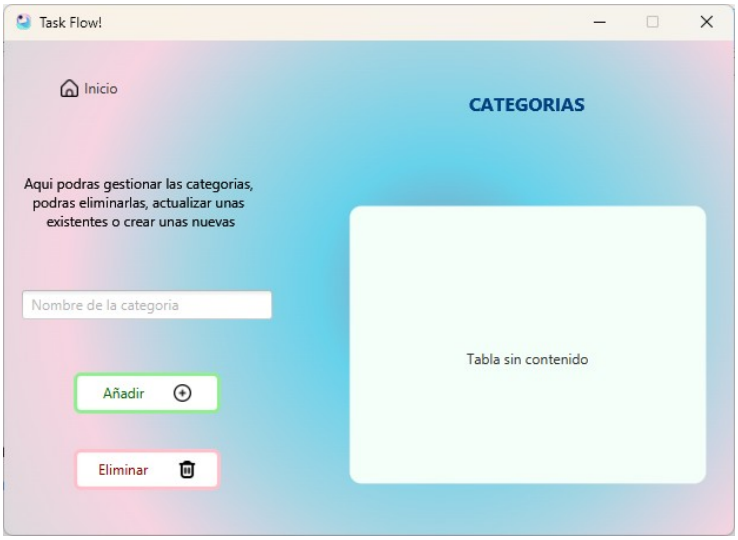


Para añadir una tarea nueva, el usuario encontrará un botón abajo a la izquierda que dice "***AÑADIR***" en el que si pulsa en él, podrá crear una nueva tarea.



Como es un usuario recién registrado, tampoco tiene categorías a las que asignar la nueva tarea. Pero por defecto existirá una categoría llamada "***SIN CATEGORÍA***" que luego podrá modificarse. Una vez pulsado el botón de aceptar volveremos a nuestra ventana de **Home** con la tarea actualizada.

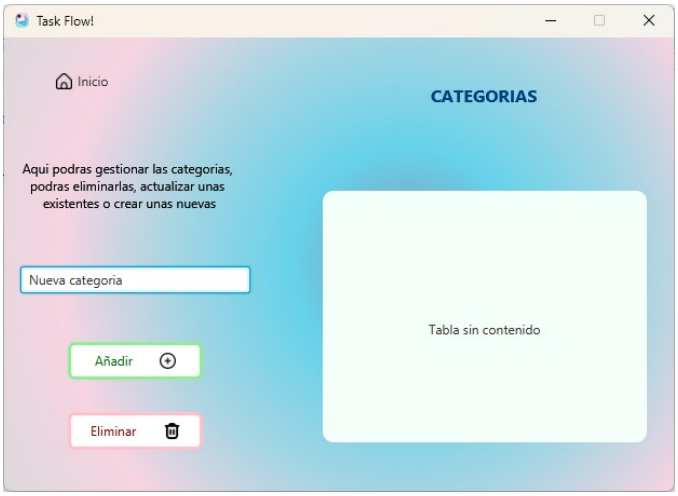
Ahora ya que tenemos una tarea creada, procederemos a crear una categoría que asignar a esa tarea nueva. En la ventana de **Home** tenemos un panel lateral en el cual encontraremos diversos botones, entre los que destacar el de ***CATEGORÍAS.***

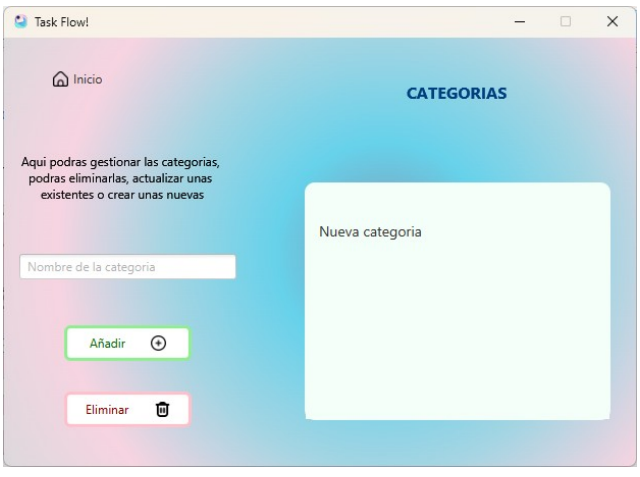


En esta ventana, el usuario podrá gestionar las tareas que hay en toda la aplicación. Estas son comunes para todos los usuarios, no como las tareas que son únicas e independientes para cada uno. Por defecto la lista aparecerá vacía ya que la categoría "***SIN CATEGORÍA***" esta oculta.

El usuario podrá ver dos botones que indican acciones sobre las categorías, en este caso querremos insertar una nueva categoría para ponérsela a nuestra nueva tarea.

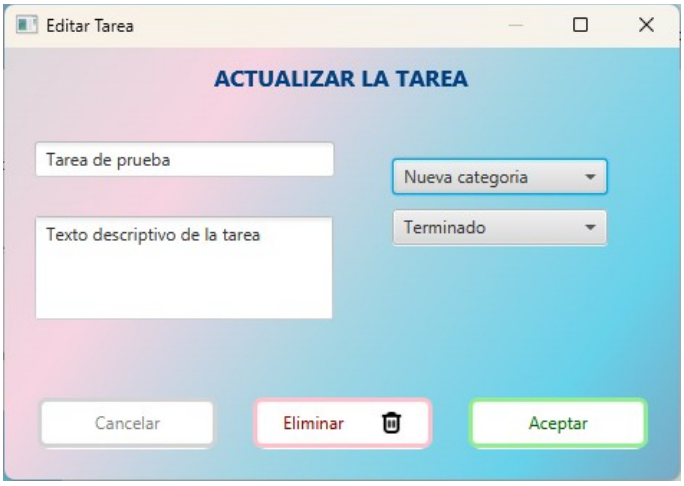
Para ello, escribiremos en el campo de texto el nombre de la categoría que querremos añadir y le pulsaremos al botón de añadir. Y automáticamente aparecerá en nuestra lista.





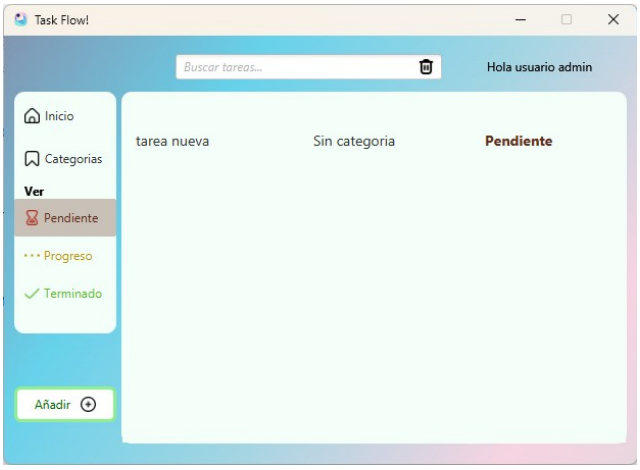
Una vez tenemos nuestra categoría creada, pulsaremos el botón de inicio y nos redirigirá a la ventana Home, donde procederemos a editar nuestra tarea para añadirle esta nueva categoría.

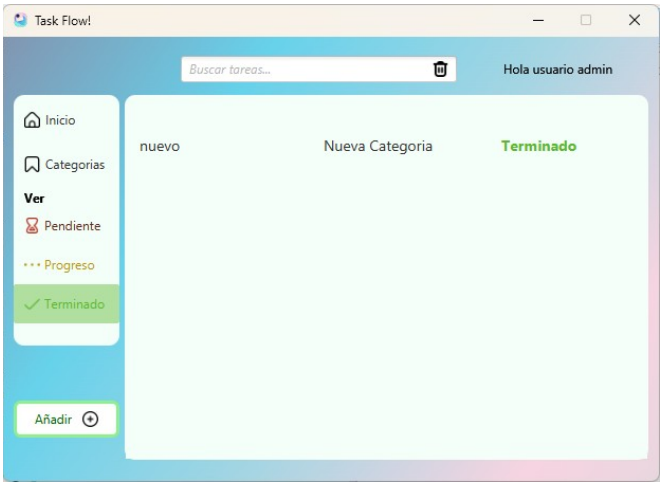
Para editar una tarea, pulsaremos dos veces sobre nuestra tarea y se abrirá una ventana con los datos de esta, y desde esa ventana nosotros podremos hacer los cambios que queramos, funcionando igual que cuando creamos una tarea. Seleccionamos nuestra categoría y le damos al botón de aceptar.



Tras haber actualizado nuestra tarea, nos aparecerá la nueva categoría como la categoría de nuestra tarea de prueba.

Como funcionalidades a mayores, se puede filtrar por los estados de las tareas en el menú lateral de ***Home,*** para demostrarlo hemos creado otras tareas





Viabilidad tecno-económica

A la hora de plantear un proyecto es importante analizar los posibles costes y evaluar la rentabilidad. Se plantea un resumen de los aspectos a tener en cuenta a la hora de ver su viabilidad:

- **Viabilidad técnica.**
  - Hay que hacer una evaluación de los recursos tecnológicos que se necesitarán para desarrollar e implementar el proyecto y tener en cuenta si están disponibles o se tienen que adquirir.
  - Habría que analizar si el equipo encargado del desarrollo del proyecto tiene los conocimientos necesarios para llevar a cabo el trabajo o si es necesario dar formación.
- **Viabilidad económica.**
  - Se tendrían que tener en cuenta los costos que pueda tener el desarrollo del programa, ademas de los costos del personal encargado del desarrollo del mismo.
  - Análisis de riesgos. Es importante en cualquier proyecto identificar los posibles riesgos y desafíos que puedan afectar a la viabilidad económica del proyecto.
  - Beneficios esperados. Se tendría que tener en cuenta si el proyecto es capaz de mejorar la productividad y la eficiencia de los trabajadores, se podrá ahorrar tiempo y recursos.

Trabajo futuro

Existen varias mejoras que se pueden considerar a la hora de mejorar la aplicación.

- **Notificaciones:** Se podría implementar un sistema de notificaciones para enviar recordatorios o alertas a los usuarios sobre las tareas pendientes. Un buen control de alertas ayudaría a mejorar la eficiencia y garantizaría que ninguna tarea importante se pase por alto.
- **Colaboración:** Seria una gran mejora para la aplicación permitir en un futuro la posible colaboración entre usuarios al asignar tareas a otros miembros. Estos cambios fomentarían una mejora en la gestión de los proyectos en equipo que se pueden llevar a cabo en *TaskFlow*.
- **Interfaz mas intuitiva:** Para hacer la experiencia del usuario aun mas cómoda y eficaz, se pueden optimizar ciertas funcionalidades, añadir atajos de teclado para hacer mas rápidas las acciones…
- **Integración de calendarios:** En un futuro se podría emplearse un uso bidireccional con calendarios externos, como Google Calendar o Outlook, para que se pueden mostrar las tareas y los plazos directamente en el calendario del usuario.

El planteamiento de estas mejoras tiene como finalidad enriquecer la funcionalidad de la aplicación, mejorar la experiencia de usuario y ser capaz de adaptarse a las necesidades cambiantes.

Conclusiones.

Durante el desarrollo de este proyecto, muchos de los objetivos planteados al comienzo fueron resueltos. *TaskFlow* es una aplicación que ha llegado para mejorar la organización y productividad a la hora de gestionar tareas. Se implementado funcionalidades clave, como la creación de tareas,la asignación de estados para una mayor organización, una búsqueda efectiva y la categorización de las tareas.

Aunque muchos de los objetivos fueron resueltos, existen también otros objetivos los cuales no pudieron ser llevados a cabo, debido a limitaciones de tiempo,recursos o la complejidad que suponían.

En cuanto a posibles lineas de investigación futura, girarían en torno a las posibles mejoras planteadas anteriormente, como puede ser la integración de calendarios para permitir una mayor sincronización y facilidad de uso para los usuarios.

Como resumen, este proyecto sería una solución para mejorar la gestión de tareas tal y como la conocemos ofreciendo una interfaz intuitiva y funcionalidades que ayudan a los usuarios a ser mucho mas organizados y productivos. Aplicando las mejorar futuras, *TaskFlow* podría convertirse en una herramienta eficiente y satisfactoria para la gestion de tareas.

Biblioteca de recursos web y referencias.

Durante este desarrollo de proyecto se han consultado diferentes recursos web.

- Documentación oficial de JavaFx (https://docs.oracle.com/javafx/2/)
- Scene Builder (https://docs.oracle.com/javafx/scenebuilder/1/user_guide/jsbpub-user_guide.htm)
- Configurar Scene Builder en IntelliJ (https://www.jetbrains.com/help/idea/opening-fxml-files-in-javafx-scene-builder.html)
- Selección de colores (https://www.w3schools.com/colors/colors_names.asp)
- Cambios en los diseños.
  - http://www.usabilidad-ux.com/usabilidad-experiencia-de-usuario-formularios-web/
  - https://uxplanet.org/5-best-practices-for-designing-effective-buttons-71a4a33e6d26
  - https://uxplanet.org/7-basic-rules-for-button-design-63dcdf5676b4
- Github profiles
  - https://gist.github.com/pethaniakshay/302072fda98098a24ce382a361bdf477
  - https://github.com/MartinDeibe02/ProyectoFinDeCiclo/blob/main/src/main/java/edu/martin/dida/proyectofinciclo/App.java
- Recursos extras
  - https://gist.github.com/jujogracu/5239976
  - https://es.stackoverflow.com/questions/345496/pulsar-boton-con-enter-en-javafx
  - https://www.youtube.com/watch?v=WQEEjD3hwjA&list=PLHdQfj3Nty4VulBDyryc5M9zfXwWo49Mc&index=12

Anexos.

Todos los recursos aquí presentados, se encuentran en la carpeta de *Documentación/Diagramas* con mayor calidad.