

TRABAJO FINAL DE CURSO



TareApp

CENTRO: Integrado de Formación Profesional nº1 de Ceuta

CURSO: Grado Superior de Desarrollo de Aplicaciones Multiplataforma

AUTOR: Francisco Rodríguez Belmonte

TUTORA: Carmen Domínguez González

CURSO: 2024/2025

RESUMEN

La aplicación TareApp permite a los usuarios crear y administrar sus tareas de forma sencilla. A lo largo de este proyecto se han propuesto objetivos que se han logrado para la gestión de usuarios, listas, tareas, notas e idioma.

Se ha desarrollado usando un modelo en cascada, que permite una planificación estructurada y progresiva a través de distintas fases: diseño, creación de vistas, desarrollo de modelos y controladores, y realización de pruebas.

Para el desarrollo también se han usado NetBeans y Android Studio, así como el patrón Modelo Vista Controlador para estructurar el código. Además, se utilizó una API web para manejar las consultas a la base de datos en la nube.

Palabras clave

Tareas, listas, notas, organizar, prioridad, agenda, planificador, app, aplicación, Android, gestor, Java, web, PHP, XML, JSON, idiomas, español, inglés, francés y diario.

ABSTRACT

The TareApp application allows users to create and manage their tasks easily. Throughout this project, objectives have been proposed and achieved for the management of users, lists, tasks, notes, and language.

It has been developed using a waterfall model, which allows for structured and progressive planning through different phases: design, creation of views, development of models and controllers, and testing.

For the development, NetBeans and Android Studio have also been used, as well as the Model-View-Controller pattern to structure the code. A web API was also used to handle queries to the cloud database.

Keywords

Tasks, lists, notes, organize, priority, agenda, planner, app, application, Android, manager, Java, web, PHP, XML, JSON, languages, Spanish, English, French, and journal.

ÍNDICE

INTRODUCCIÓN.....	1
JUSTIFICACIÓN DEL PROYECTO.....	2
OBJETIVOS.....	3
ALTERNATIVAS.....	4
1. Google Keep.....	4
2. Microsoft To Do.....	4
3. Todoist.....	5
ANÁLISIS DAFO.....	6
REQUISITOS FUNCIONALES.....	7
REQUISITOS NO FUNCIONALES.....	8
DIAGRAMA DE CLASES.....	9
DIAGRAMA DE FLUJO.....	10
DIAGRAMA DE GANTT.....	11
DISEÑO DE LA BASE DE DATOS.....	12
Modelo Entidad Relación.....	12
Modelo Relacional.....	12
Normalización de la base de datos.....	13
DISEÑO DE INTERFACES.....	15
Paleta de colores.....	15
Tipografía.....	16
Iconos y otros elementos.....	16
Mockups.....	17
DESARROLLO.....	19
Aspectos relevantes del desarrollo.....	19
Metodologías.....	20
CONCLUSIÓN.....	21
BIBLIOGRAFÍA.....	22
ANEXOS.....	23

INTRODUCCIÓN

TareApp es una aplicación para la gestión de tareas que permite a los usuarios crear, editar, borrar, organizar y priorizar sus tareas pendientes de forma eficiente. Facilita el seguimiento de actividades diarias mediante filtros por prioridad, fecha o estado, e incluye un sistema de notas rápidas para apuntar información importante.

Una vez registrado, el usuario puede crear listas y, dentro de ellas, añadir tareas con título, fecha, prioridad y descripción. Las tareas pueden marcarse como completadas o incompletas y visualizarse en función de distintos filtros, lo que permite mantener una organización clara y adaptada a las necesidades de cada momento.

JUSTIFICACIÓN DEL PROYECTO

La idea principal de crear esta aplicación es para poder tener todas nuestras tareas apuntadas de forma fácil y rápida, permitiendo ver nuestras tareas tanto en móvil como en ordenador.

Esta aplicación está dirigida sobre todo a un público como estudiantes o cualquier persona que necesite organizar sus tareas. Al ser una aplicación sencilla hace que cualquier persona pueda utilizar la aplicación.

La aplicación cubre la posibilidad de apuntar tareas mediante listas y notas rápidas.

OBJETIVOS

- **Gestionar Usuarios**

Este apartado nos permite poder registrarnos e iniciar sesión, la idea es poder almacenar todos los datos de la aplicación a una cuenta, para que pueda tener sus datos en distintos dispositivos.

- **Gestionar Listas de tareas**

La idea es tener todas las tareas ordenadas en listas para que así el usuario pueda llevar un mejor control de sus tareas. Las listas cuentan solo con un título para saber qué tareas se almacenarán dentro.

- **Gestionar Tareas**

Para las tareas se puede añadir un título para saber qué tarea era, una fecha para saber cuando se entrega la tarea, una prioridad para saber lo importante que es la tarea y una descripción para explicar con más detalle la tarea.

- **Gestionar notas**

El apartado de notas sirve para que un usuario pueda apuntarse algo rápido, en este apartado solo se añade un texto de la nota y un color, la idea es que se apunte algo como recordatorio, sin prioridad, ni fecha, etc.

- **Uso de idioma**

La aplicación tiene varios idiomas como Español, inglés y francés para que así llegar a un mayor público.

ALTERNATIVAS

1. Google Keep

Descripción: Es una aplicación de Google para almacenar notas. Es muy sencilla y con una interfaz muy intuitiva.

Ventajas:

- Uso de las herramientas de Google (Gmail, Calendar, Drive, etc.).
- Interfaz limpia y fácil de usar.
- Soporte multiplataforma (web, Android, iOS).
- Notas rápidas con colores y etiquetas.

Desventajas:

- No tiene gestión de tareas y listas.
- No permite el trabajo offline

2. Microsoft To Do

Descripción: Herramienta gratuita de Microsoft para gestionar tareas personales y de trabajo.

Ventajas:

- Uso de las herramientas de Microsoft (Outlook, Microsoft One, etc.).
- Permite crear listas, tareas, subtareas, recordatorios y fechas límite.
- Interfaz limpia y sincronización en tiempo real.
- Soporte multiplataforma (web, Android, iOS).

Desventajas:

- Enfoque para proyectos más cortos
- Algunas funciones requieren cuenta Microsoft.

3. Todoist

Descripción: Aplicación de gestión de tareas y proyectos, ideal tanto para uso personal como profesional.

Ventajas:

- Sistema de etiquetas, prioridades y recordatorios.
- Posibilidad de colaborar en proyectos con otros usuarios.
- Diseño moderno y limpio.
- Disponible en múltiples plataformas.

Desventajas:

- Algunas funcionalidades están limitadas en la versión gratuita.
- No incluye una sección para notas.
- Puede resultar compleja para usuarios novatos.

ANÁLISIS DAFO

El análisis DAFO es una herramienta que ayuda a entender una situación mirando cuatro aspectos: fortalezas y debilidades internas, y oportunidades y amenazas externas, para tomar mejores decisiones.



REQUISITOS FUNCIONALES

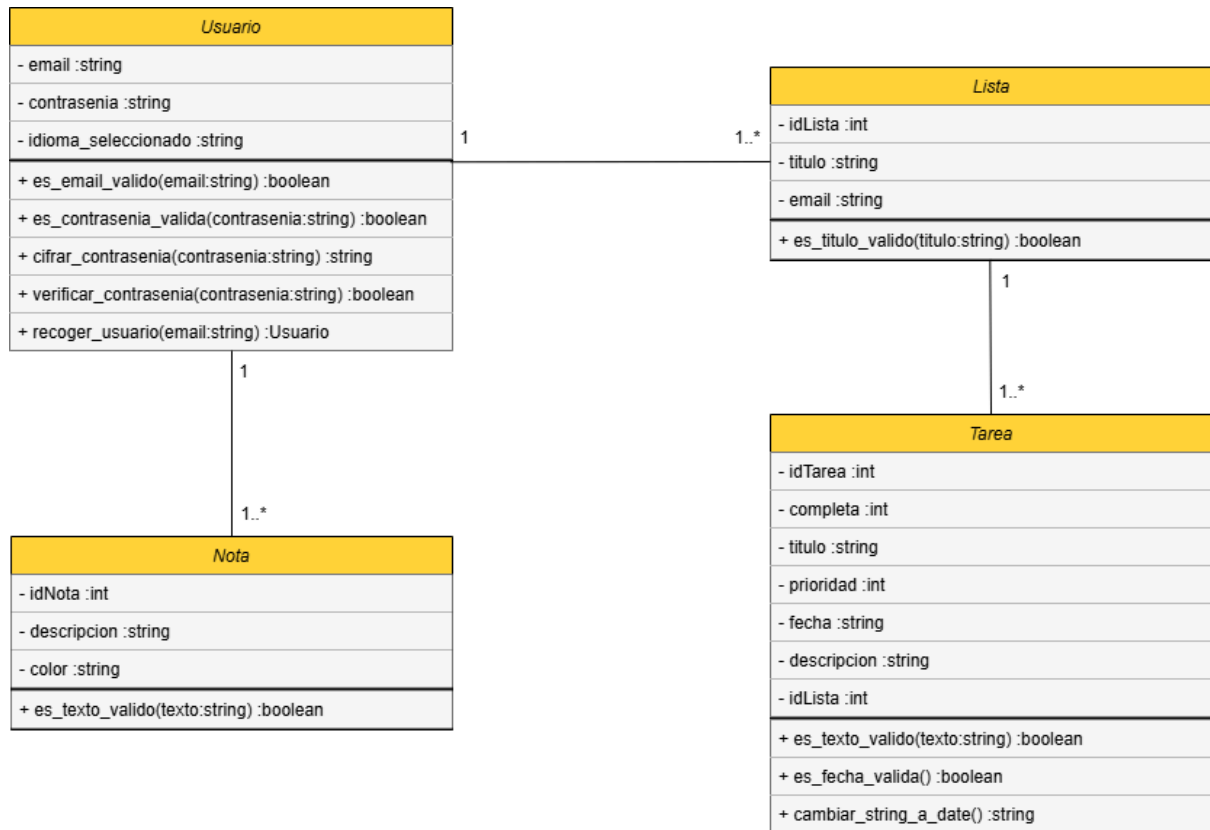
- **RF - 1. Gestión de usuarios.**
 - RF - 1.1. Ver usuarios.
 - RF - 1.2. Crear usuarios.
 - RF - 1.3. Editar usuarios.
 - RF - 1.4. Borrar usuarios.
 - RF - 1.5. Iniciar sesión.
 - RF - 1.6. Cerrar sesión.
- **RF - 2. Gestión de listas de tareas.**
 - RF - 2.1. Ver listas.
 - RF - 2.2. Crear listas.
 - RF - 2.3. Editar listas.
 - RF - 2.4. Borrar listas.
- **RF - 3. Gestión de tareas.**
 - RF - 3.1. Ver tareas.
 - RF - 3.2. Crear tareas.
 - RF - 3.3. Editar tareas.
 - RF - 3.4. Borrar tareas.
 - RF - 3.5. Filtrar tareas.
 - RF - 3.6. Marcar/Desmarcar como completadas.
- **RF - 4. Gestión de notas.**
 - RF - 4.1. Ver notas.
 - RF - 4.2. Crear notas.
 - RF - 4.3. Editar notas.
 - RF - 4.4. Borrar notas.
- **RF - 5. Uso de idiomas.**
 - RF - 5.1. Cambiar idioma.
 - RF - 5.2. Idiomas disponibles: español, francés, inglés.

REQUISITOS NO FUNCIONALES

- **RF - 1. Usabilidad.**
 - RF - 1.1. Muestra mensajes de error.
 - RF - 1.2. Intuitiva y fácil de usar.
 - RF - 1.3. Buena legibilidad.
 - RF - 1.4. Distribución y navegación intuitiva.
- **RF - 2. Funcionalidad.**
 - RF - 2.1. Para crear listas, primero hay que tener un usuario.
 - RF - 2.2. Para crear tareas, primero hay que tener una lista.
 - RF - 2.3. Para crear notas primero hay que tener un usuario.
- **RF - 3. Seguridad.**
 - RF - 3.1. Contraseñas de usuario cifradas.
 - RF - 3.2. Confirmación de correo para validar la cuenta.
- **RF - 4. Disponibilidad.**
 - RF - 4.1. Compatible con dispositivos móviles (Android) y ordenadores con Windows.

DIAGRAMA DE CLASES

Las clases representadas son sólo los modelos de datos, no se representan las demás clases ya que la aplicación está estructurada usando el patrón Modelo Vista Controlador, provocando que haya más de 40 clases, repartidas entre componentes, vistas, modelos, controladores, clases para el control del idioma.

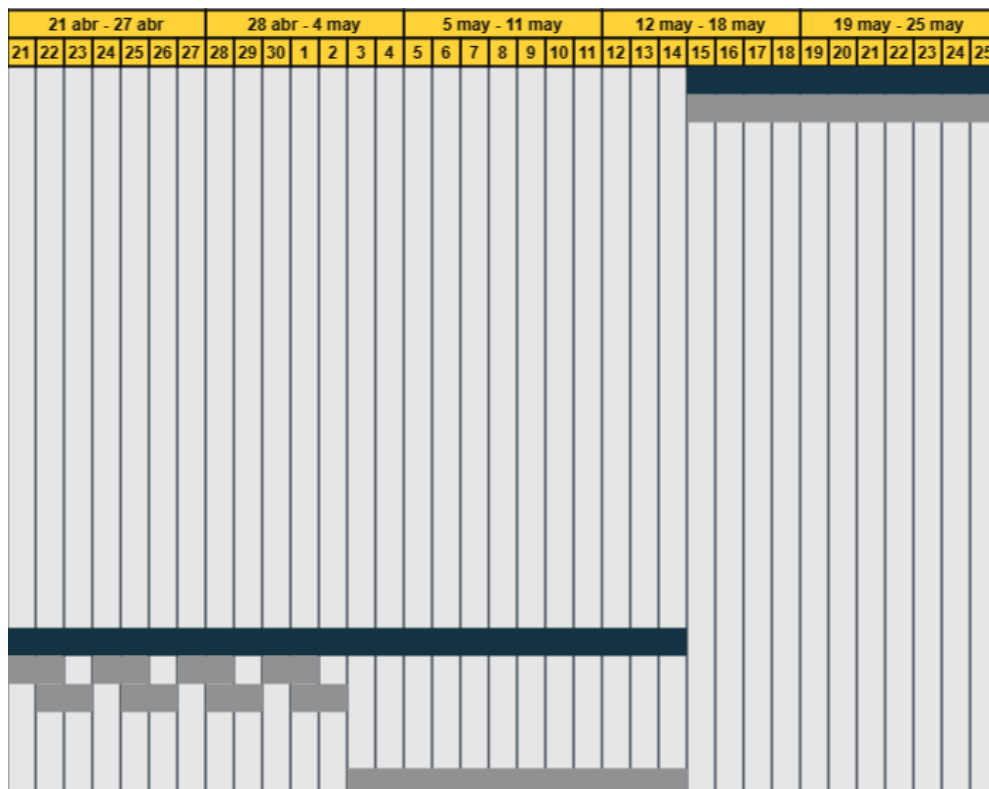
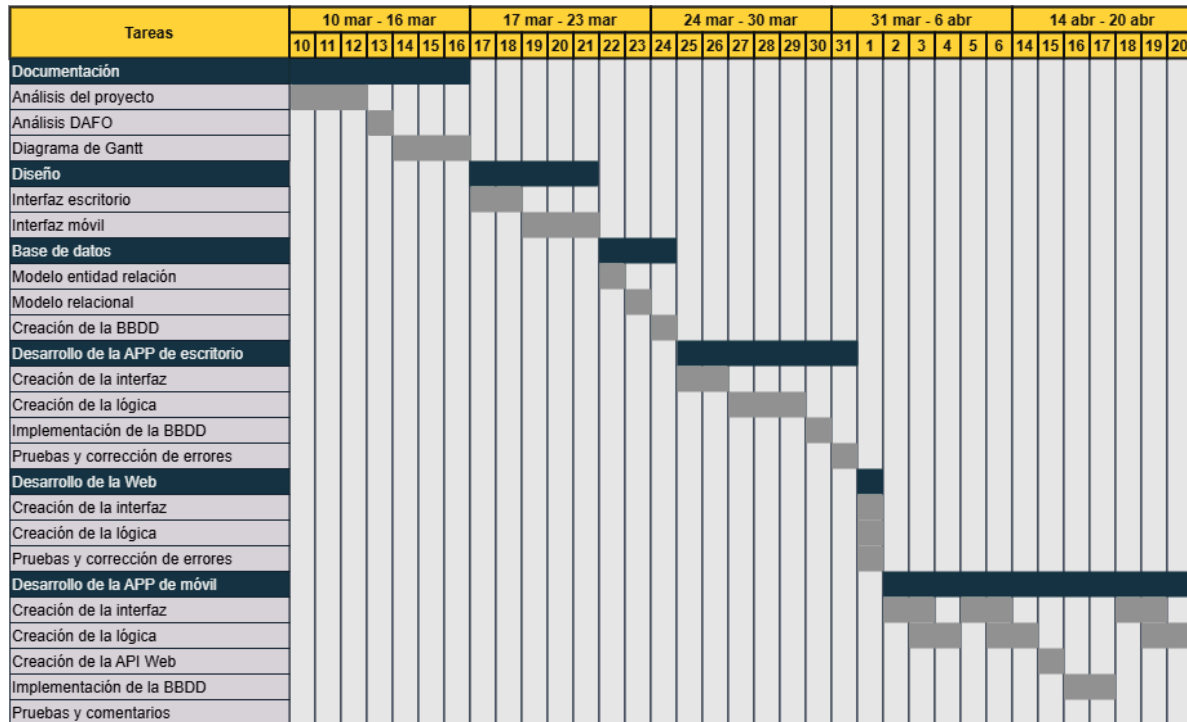


El diagrama muestra el flujo que seguiría el usuario por la aplicación, el diagrama viene adjunto en la documentación del proyecto para su correcta visualización.



DIAGRAMA DE GANTT

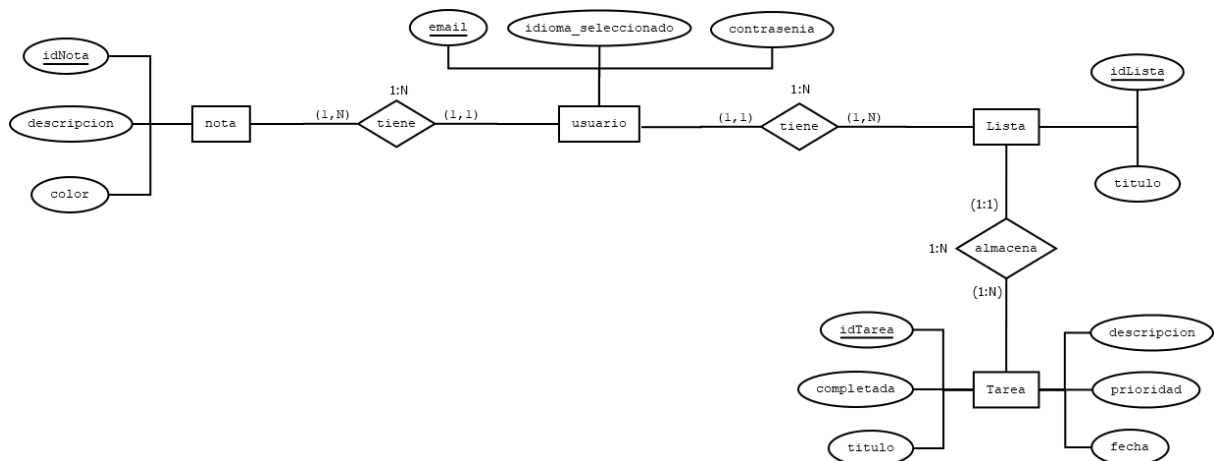
El diagrama de gantt muestra el tiempo dedicado a las diferentes partes del proyecto, el diagrama viene adjunto en la documentación del proyecto para su correcta visualización.



DISEÑO DE LA BASE DE DATOS

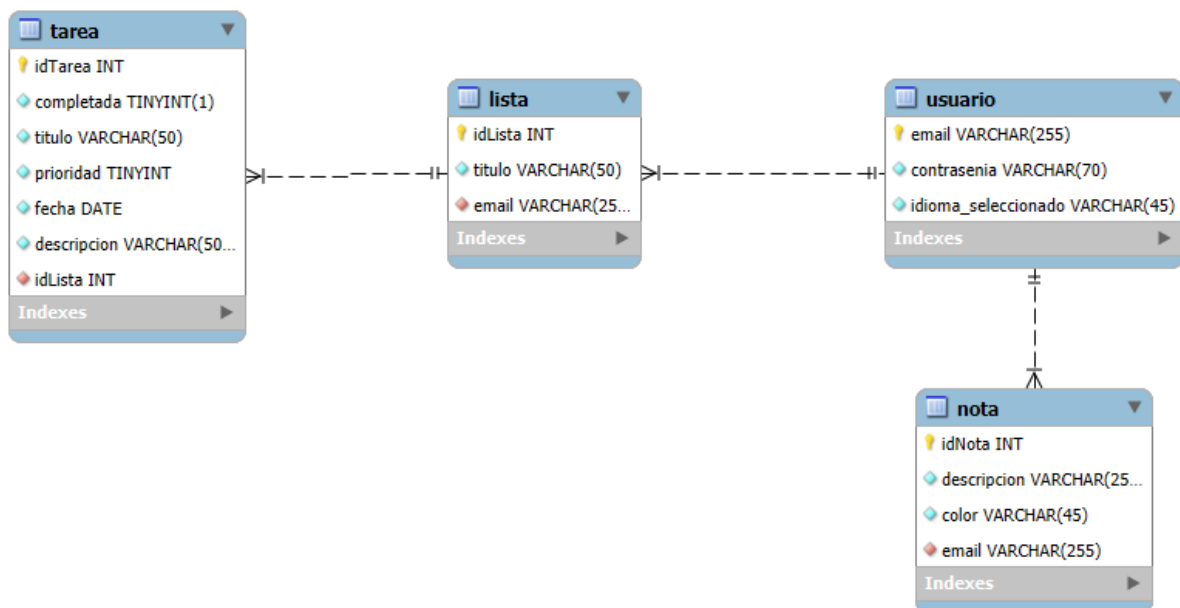
Modelo Entidad Relación

Es un diagrama que muestra cómo se relacionan los datos entre sí, representando entidades (como personas o cosas) y las relaciones que existen entre ellas.



Modelo Relacional

Es una manera de organizar los datos en tablas con filas y columnas, donde cada tabla representa una entidad y se pueden conectar entre sí mediante claves.



Normalización de la base de datos

La normalización sirve para poder minimizar la redundancia y evitar inconsistencias.

Primera Forma Normal (1FN)

- Reglas:
 - Cada atributo es atómico.
 - Cada fila tiene que tener una clave primaria.
 - Sin columnas duplicadas para un mismo tipo de dato.

1FN	Clave primaria	Átomico	Columna duplicada
Tabla Usuario	email	Sí	No
Tabla Lista	idLista	Sí	No
Tabla Tarea	idTarea	Sí	No
Tabla Nota	idNota	Sí	No

Segunda Forma Normal (2FN)

- Reglas:
 - Cumple con la 1FN.
 - Cada atributo no clave depende de toda la clave primaria
 - Tabla con clave compuesta

2FN	Cumple con la 1FN	Los atributos dependen de la clave	Clave compuesta
Tabla Usuario	Sí	Sí	No
Tabla Lista	Sí	Sí	No
Tabla Tarea	Sí	Sí	No
Tabla Nota	Sí	Sí	No

Tercera Forma Normal (3FN)

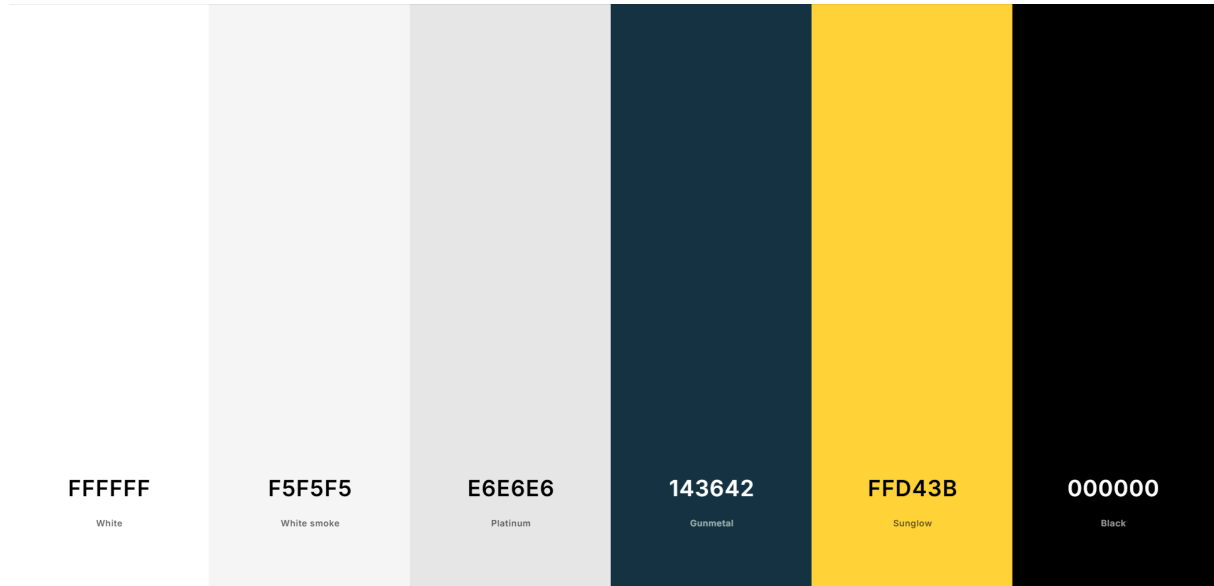
- Reglas:
 - Cumple con la 2FN.
 - Un atributo no clave no puede depender de otro no clave.

3FN	Cumple con la 2FN	Un atributo no clave depende de otro no clave
Tabla Usuario	Sí	No
Tabla Lista	Sí	No
Tabla Tarea	Sí	No
Tabla Nota	Sí	No

DISEÑO DE INTERFACES

Paleta de colores

- Colores principales



- Colores secundarios



Tipografía

Buena legibilidad y con un diseño profesional



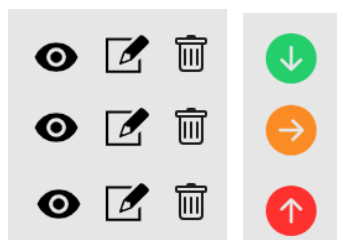
Iconos y otros elementos

En la parte superior se encuentra la cabecera donde se puede navegar por la aplicación.

En la barra azul encontramos el botón para crear la tarea y todos los iconos para los filtros.



Iconos para ver, editar y borrar elementos y los iconos para la prioridad de la tarea.



Icono para crear listas e icono para marcar y desmarcar tareas.



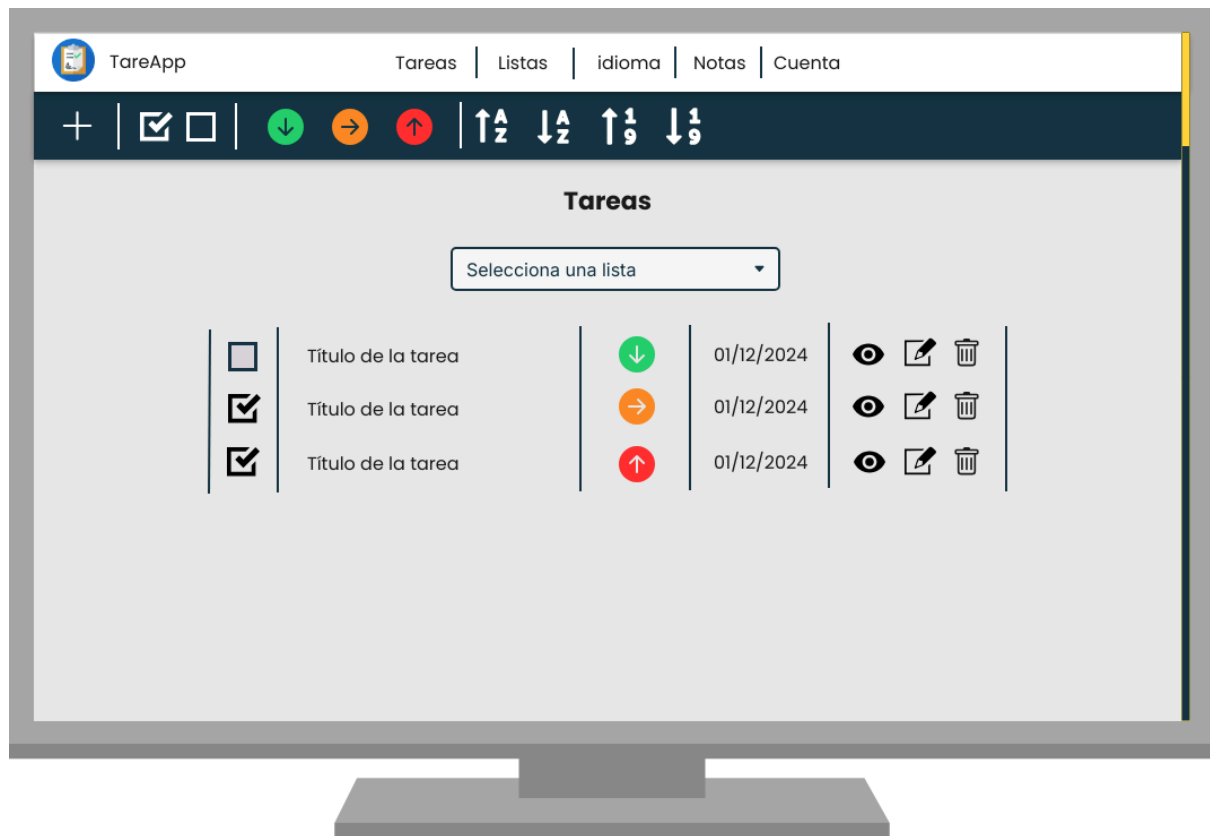
Mockups

Página tareas

En la cabecera si se pulsa en “Tareas” se accederá a la siguiente página donde se podrán ver, editar, borrar y filtrar las tareas de la lista seleccionada.

En la barra de Herramientas (barra azul oscuro) se encuentran los filtros.

En la parte inferior están las tareas, cada tarea puede ser marcada como completada o incompleta con el botón de la izquierda, se visualiza su título, prioridad y fecha y a la derecha hay iconos para poder ver, editar y borrar la tarea.



Página notas

En esta página se pueden crear notas rápidas para poder apuntarse cosas importantes que no son tareas, solo anotaciones. La vista tiene un botón para poder crear la nota y un panel donde se van mostrando todas las notas.

Al pulsar en una nota se muestra un PopUp donde vemos el texto que tiene la nota, un botón para editarla y un icono para borrarla.



Visualización de todos los diseños

Para ver todos los diseños correctamente, con una explicación más completa se ha adjuntado una guía de estilos a la que se puede acceder a través de la página Web o del pdf adjunto en los archivos de la documentación del proyecto.

[Enlace para la guía de estilos de la Web](#)

DESARROLLO

Para el desarrollo de la aplicación se han utilizado diferentes herramientas, para crear la versión de escritorio se ha utilizado **NetBeans** y para la versión móvil se ha utilizado **Android Studio**, todo esto utilizando el lenguaje de programación de java.

Para estructurar la aplicación se ha utilizado el patrón de diseño **Modelo Vista Controlador** (MVC) de esta manera se deja todo mucho más ordenado y estructurado. Las vistas se comunican con los controladores y los controladores con los modelos.

Aspectos relevantes del desarrollo

La app cuenta con una página Web donde se pueden descargar las versiones de la aplicación de escritorio y móvil, la página es "**tareapp.blog**".

Un aspecto a destacar es el uso de **confirmación por correo electrónico**. Al registrarse o al cambiar el correo asociado a la cuenta, se solicita una verificación para evitar el uso de direcciones de correo falsas.

Otro punto importante es el uso de una **base de datos en la nube**, lo que permite que tanto la versión móvil como la de escritorio compartan la misma información. De esta manera, el usuario puede acceder a sus datos desde distintas plataformas sin inconvenientes.

Además, para poder acceder a la base de datos, es necesario utilizar una **API**, ya que no es posible conectarse directamente desde Java. Por ello, se ha desarrollado una API web utilizando **PHP**, la cual permite realizar las consultas necesarias desde la aplicación.

Metodologías

Para el desarrollo de la aplicación se utilizó una **metodología en cascada**, en la cual el trabajo se organizó en fases secuenciales, permitiendo una planificación clara y estructurada. Las etapas fueron las siguientes:

1. **Diseño de la interfaz (Mockups)**

Se comenzó realizando los mockups de la aplicación para definir visualmente la estructura y el diseño de las vistas, permitiendo una mejor planificación de la interfaz de usuario.

2. **Creación de las vistas**

A partir de los diseños, se han hecho las vistas en java y en xml, intentando hacerlas lo más parecido a los mockups.

3. **Desarrollo del modelo y controlador**

Se implementaron las clases del modelo y los controladores, haciendo que las vistas se comuniquen con los controladores y los controladores con los modelos, integrando la lógica de la aplicación y realizando la conexión con la base de datos.

4. **Pruebas y comprobaciones**

Finalmente, se realizaron pruebas unitarias y pruebas de integración, así como revisiones generales, para asegurar el correcto funcionamiento de cada módulo por separado y de la aplicación en su conjunto.

CONCLUSIÓN

A lo largo del desarrollo de este proyecto se ha logrado cumplir los objetivos, creando así una aplicación multiplataforma (escritorio y móvil) para la creación de usuario, tareas, notas y listas, así como la implementación del idioma.

Se ha conseguido utilizar el patrón de diseño Modelo Vista Controlador para organizar mejor el código. Además, para un mejor uso de la aplicación se ha conseguido implementar una base de datos en la nube para que ambas aplicaciones tengan los mismos datos.

Se han cumplido todas las fases de la metodología en cascada con éxito, haciendo que el desarrollo de la aplicación sea sencillo y cómodo.

Aunque la aplicación cuente con lo imprescindible para la administración de tareas, siempre se pueden añadir mejoras en un futuro como por ejemplo:

- Añadir compatibilidad con IOS.
- Cambiar la tecnología a alguna más moderna como Flutter, sobre todo para la versión de escritorio.
- Poder añadir un apartado con un calendario donde se visualicen todas las tareas en el calendario.
- Añadir notificaciones para el aviso de la finalización de la tarea.
- Añadir la posibilidad de compartir tareas.

En conclusión, el proyecto se ha completado conforme a lo esperado, cumpliendo los objetivos establecidos y logrando un funcionamiento sólido, validado mediante numerosas pruebas que garantizan la calidad de la aplicación.

BIBLIOGRAFÍA

Alternativas de TareApp

Alternativa Google Keep

Alternativa Microsoft To Do

Alternativa Todoist

Clase SMTP para enviar correos

Stack overflow

Apuntes de clase

ANEXOS

Se adjunta un documento PDF con la guía de estilos completa.

Se adjunta un documento PDF con la presentación del proyecto.

Se adjunta una imagen con el diagrama de clases.

Se adjunta una imagen con el diagrama de flujo.

Se adjunta una imagen con el diagrama de Gantt.

Se adjuntan imágenes con todos los diseños de la aplicación.

Se adjunta una imagen con el modelo entidad-relación.

Se adjunta una imagen con el modelo relacional.