



[Home](#)

[About](#)

[Content](#)

[Others](#)

PROGRAMMING PROYECTO PROGRAMACIÓN

JUAN DÍAZ
DAVID MARTINEZ
DANIEL BENÍTEZ

INTRODUCCIÓN

Nuestro proyecto consiste en un programa de escritorio en Java diseñado para gestionar tareas de forma organizada y sencilla. El usuario puede crear tareas, agruparlas por categorías, editarlas, eliminarlas y visualizarlas mediante una interfaz simple basada en JOptionPane. Toda la información se guarda en un archivo local en formato JSON y además se sincroniza automáticamente con Google Drive, lo que garantiza un respaldo confiable en la nube. Cuando el programa inicia intenta cargar los datos locales y si no están disponibles recupera una copia directamente desde Google Drive.



```
"container">
class="row">
v class="col-md-6 col-lg-8"> <!--
<nav id="nav" role="navigation">
  <ul>
    <li><a href="index.html">Home</a><
    <li><a href="home-events.html">Hom
    <li><a href="multi-col-menu.html">
    <li class="has-children"> <a href=
      <ul>
        <li><a href="tall-button-he
        <li><a href="image-logo.htm
        <li class="active"><a href=
      </ul>
    </li>
    <li class="has-children"> <a href=
      <ul>
        <li><a href="variable-width-
```


OBJETIVO GENERAL

- Diseñar un programa que permita organizar actividades y guardar la información localmente y en la nube mediante Google Drive.

OBJETIVOS ESPECÍFICOS

- Implementar estructuras de datos para tareas y grupos
- Realizar persistencia local con JSON
- Integrar sincronización con Google Drive
- Crear interfaz con JOptionPane
- Dividir el proyecto en clases y paquetes bien organizados
- Utilizar Maven y librerías externas como Gson



OBJETIVOS DEL PROYECTO

```
17 string sInput;  
18 int iLength, iN;  
19 double dblTemp;  
20 bool again = true;  
21  
22 while (again) {  
23     iN = -1;  
24     again = false;  
25     getline(cin, sInput);  
26     system("cls");  
27     stringstream(sInput) >> dblTemp;  
28     iLength = sInput.length();  
29     if (iLength < 4) {  
30         again = true;  
31     } else if (dblTemp != '.') {
```


FUNCIONAMIENTO DEL PROGRAMA

Al ejecutarse, el programa realiza una carga automática de información intentando primero obtener los datos del archivo local data json y si este no existe o está vacío procede a buscar una copia en Google Drive. El usuario puede crear grupos, registrar tareas dentro de cada uno, visualizar toda la información organizada y también editar o eliminar elementos según sea necesario. Cada cambio se guarda tanto de forma local como en la nube, asegurando que la información esté siempre actualizada.



ESTRUCTURA DEL PROYECTO

ESTÁ DIVIDIDO EN :

- Task.java estructura de una tarea
- DataBundle.java contenedor de tareas y grupos
- LocalStorage.java manejo de guardado y carga local
- GoogleDriveStorage.java subida y descarga desde Google Drive
- SimpleTasks.java interfaz principal del usuario

```
resp_iter = self.stub.GatherResponses()  
for data in resp_iter:  
    status = Status(  
        status_id=data.id, name=data.name,  
        tuses[status.name] = status
```

DANIEL BENÍTEZ

- Manejó la creación de la estructura de las tareas.
- Ayudó a diseñar la forma de visualizar las tareas en listas. Probó el funcionamiento completo de la app.
- Hizo la parte de explicación del funcionamiento en la sustentación.

JUAN DÍAZ

- Encargado de la organización del proyecto y estructura de carpetas.
- Ayudó a preparar los métodos para eliminar tareas y grupos. Probó varios errores y ayudó a corregirlos.
- Participó en la creación del README.

DAVID MARTINEZ

- Integró Maven, estructuras, y la sincronización con Google Drive.
- Manejó los problemas de credenciales y ejecución avanzada.
- Revisó la lógica del menú e hizo mejoras del diseño.
- Coordinó la documentación final y la presentación.

TRABAJO DE LOS INTEGRANTES



RETOS ENFRENTADOS



Uno de los principales retos fue lograr que el programa guardara y cargara la información sin dañar el archivo JSON, lo cual se corrigió diseñando adecuadamente la clase DataBundle. También encontramos problemas con la visualización, ya que los saltos de línea aparecían incorrectos, y lo solucionamos usando un formateo más organizado. Otro reto fue que algunas tareas no se eliminaban aun cuando el usuario lo solicitaba, lo cual resolvimos con filtrado por streams y actualizando los guardados locales y en Drive. La integración con Google Drive resultó ser lo más complejo debido a permisos y credenciales. Además reorganizamos todo el proyecto que inicialmente estaba en un solo archivo dividiéndolo en los paquetes model storage y ui.



CONCLUSIONES Y MEJORAS FUTURAS



Conclusiones

- Aprendimos a estructurar un programa real en Java
- Utilizamos JSON Maven y la API de Google Drive
- Trabajamos en equipo con responsabilidades claras
- La IA fue apoyo técnico pero no realizó el proyecto

Mejoras Futuras

El proyecto podría evolucionar incorporando una interfaz gráfica mediante JavaFX, un sistema de login para diferentes usuarios, sincronización totalmente bidireccional con Drive, notificaciones de tareas pendientes y exportación de información a PDF o Excel.



MUCHAS GRACIAS
THANK YOU
THANK YOU