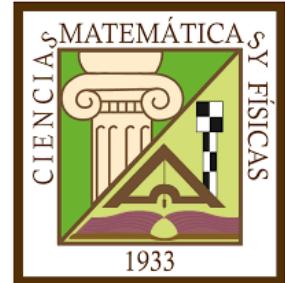




Nombre: Isaac Yépez



Curso: NO 9-5

Asignatura: GESTIÓN DE LA
CONFIGURACIÓN DEL SOFTWARE

Tema: Proyecto

Fecha: 02 de Febrero del 2026

Grupo I Integrantes:

Yépez Alonso Isaac

Didier Alejandro Arteaga Calderón

Lindao Herrera Alan Xavier

Rommel Adolfo Campuzano Mosquera

Quiñonez Lara Roy Alexander

INFORME DEL PROYECTO

En el presente proyecto se desarrolló una aplicación de consola denominada TaskManager Grupo I, cuyo objetivo principal es gestionar usuarios y tareas mediante el uso del gestor de dependencias Apache Maven. Este proyecto permitió aplicar los conceptos relacionados con la gestión de configuración del software, el ciclo de vida de desarrollo y la automatización de procesos de compilación, pruebas, empaquetado y despliegue. A través del uso de herramientas modernas, se buscó garantizar la organización, mantenibilidad y calidad del sistema desarrollado.

Objetivo del Proyecto

El objetivo principal de este proyecto fue evaluar y aplicar los conceptos de gestión de dependencias y ciclo de vida del software utilizando Maven. Para ello, se creó un proyecto estructurado que integra bibliotecas externas, pruebas unitarias y procesos automatizados. Asimismo, se buscó fortalecer las habilidades del estudiante en el uso de entornos de desarrollo integrados y herramientas de construcción profesional.

Herramientas Utilizadas

Para el desarrollo del proyecto se utilizaron diversas herramientas tecnológicas. Como entorno de desarrollo integrado se empleó Apache NetBeans, el cual facilitó la creación y administración del proyecto Maven. Se utilizó el lenguaje de programación Java para la implementación del sistema. Apache Maven fue utilizado como gestor de dependencias y automatizador del ciclo de vida del proyecto. Además, se integraron bibliotecas externas como JUnit para pruebas unitarias, Log4j para el manejo de registros, Apache Commons Lang para utilidades y Gson para la manipulación de archivos JSON.

Configuración Inicial del Proyecto

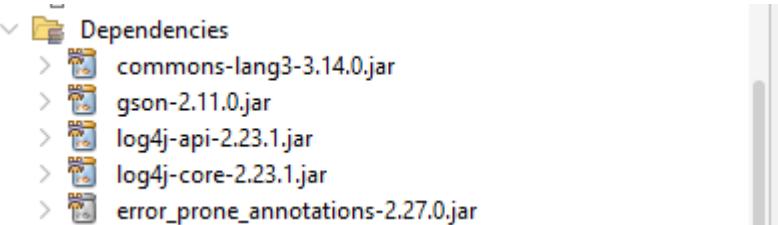
El proyecto fue creado utilizando el arquitecto maven-archetype-quickstart, lo que permitió generar automáticamente una estructura estándar. Durante esta fase se configuraron los identificadores principales como groupId, artifactId y version en el archivo pom.xml. Posteriormente, se definieron las propiedades del proyecto, como la codificación de caracteres y la versión del compilador, garantizando compatibilidad con el entorno de desarrollo.

La estructura generada permitió separar el código fuente principal, las pruebas unitarias y los recursos del sistema, facilitando el mantenimiento y escalabilidad del proyecto.

Gestión de Dependencias

La gestión de dependencias se realizó mediante el archivo pom.xml, en el cual se declararon las bibliotecas necesarias para el funcionamiento del sistema. Entre las dependencias utilizadas se encuentra JUnit, empleada para la creación y ejecución de pruebas unitarias. Log4j fue incorporado para registrar eventos, errores y procesos del sistema. Apache Commons Lang fue utilizado para facilitar operaciones sobre cadenas de texto y validaciones. Finalmente, Gson permitió almacenar y recuperar información en formato JSON.

Cada dependencia fue configurada con su respectiva versión, asegurando estabilidad y compatibilidad. Gracias a Maven, estas bibliotecas fueron descargadas automáticamente desde repositorios remotos.



```
<!-- Clase principal -->
<exec.mainClass>com.mycompany.grupoI.proyectoGrupoI</exec.mainClass>

<!-- Versiones -->
<junit.version>5.10.2</junit.version>
<log4j.version>2.23.1</log4j.version>
<commons.version>3.14.0</commons.version>
<gson.version>2.11.0</gson.version>
</properties>

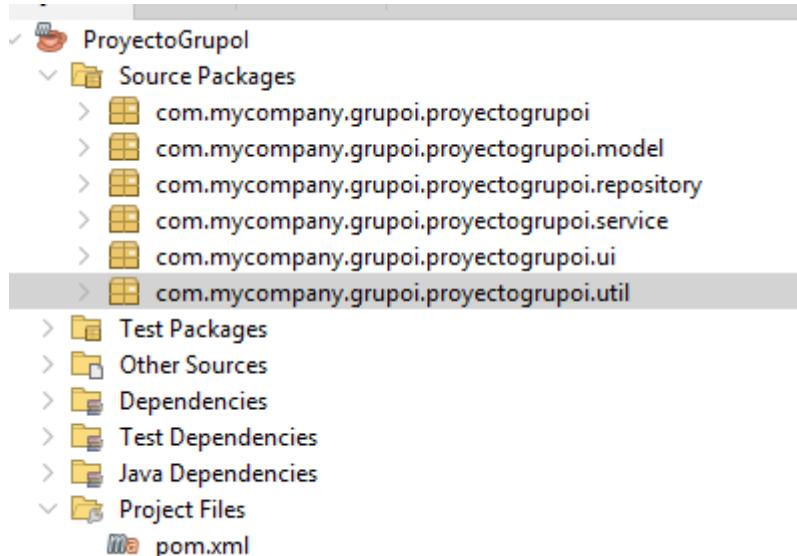
<!-- Dependencias -->
<dependencies>

    <!-- JUnit: Pruebas unitarias -->
    <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter</artifactId>
        <version>${junit.version}</version>
        <scope>test</scope>
    
```

Arquitectura del Sistema

El sistema fue desarrollado bajo una arquitectura modular basada en capas, con el fin de mejorar la organización del código. Se implementaron paquetes separados para los modelos, repositorios, servicios, utilidades y la interfaz de usuario. Los modelos representan las entidades principales como usuarios y tareas. Los repositorios se encargan del almacenamiento de datos en archivos JSON. Los servicios contienen la lógica de negocio, mientras que la interfaz de usuario gestiona la interacción con el usuario mediante un menú en consola.

Esta estructura favorece la reutilización del código, facilita la corrección de errores y permite futuras ampliaciones del sistema.



Implementación de la Aplicación

La aplicación desarrollada permite a los usuarios registrarse, iniciar sesión y gestionar sus tareas personales. Cada usuario puede crear tareas, asignarles prioridad, definir fechas límite, marcarlas como completadas o eliminarlas. Los datos se almacenan de forma persistente en archivos JSON, garantizando que la información no se pierda al cerrar el programa.

Además, se implementó un sistema de validación de datos para evitar errores en la entrada de información. El uso de funciones de hash permitió proteger las contraseñas, aumentando el nivel de seguridad del sistema.

```
C:\Users\PC PRO\Desktop\ProyectoGrupoi>java -jar target/ProyectoGrupoi-1.0.jar
10:07:44 [main] INFO com.mycompany.grupoi.proyectogrupoi.ProyectoGrupoi - Iniciando TaskManager Grupo I
```

```
== TaskManager Grupo I ==
1) Registrarse
2) Iniciar sesión
0) Salir
Opción: |
```

```
== TaskManager Grupo I ==
1) Registrarse
2) Iniciar sesión
0) Salir
Opción: 1
Usuario: Isaac
Contraseña: 123
Error: El usuario ya existe.

== TaskManager Grupo I ==
1) Registrarse
2) Iniciar sesión
0) Salir
Opción: 2
Usuario: Isaac
Contraseña: 123
Bienvenido, isaac

--- Menú de Usuario (isaac) ---
1) Crear tarea
2) Listar tareas
3) Completar tarea
4) Eliminar tarea
0) Cerrar sesión
Opción: |
```

```
--- Menú de Usuario (isaac) ---
1) Crear tarea
2) Listar tareas
3) Completar tarea
4) Eliminar tarea
9) Cerrar sesión
Opción: 2

ID | Estado | Prioridad | Fecha | Título
a9560ba5-02d3-48ec-af42-7ed97fdbd633e | OK | MEDIA | 2027-06-05 | Proyecto de aprendizaje
```

```
--- Menú de Usuario (isaac) ---
1) Crear tarea
2) Listar tareas
3) Completar tarea
4) Eliminar tarea
9) Cerrar sesión
Opción: |
```

```
2) Listar tareas
3) Completar tarea
4) Eliminar tarea
9) Cerrar sesión
Opción: 1
Título: Prueba
Descripción: Prueba de proyecto
Prioridad (BAJA/MEDIA/ALTA): Alta
Fecha límite (YYYY-MM-DD) o vacío: 2026-02-02
Tarea creada con ID: 620001d9-9222-4f45-aa99-d14aef3bed0
```

```
--- Menú de Usuario (isaac) ---
1) Crear tarea
2) Listar tareas
3) Completar tarea
4) Eliminar tarea
9) Cerrar sesión
Opción: 2
```

```
ID | Estado | Prioridad | Fecha | Título
620001d9-9222-4f45-aa99-d14aef3bed0 | PEND | ALTA | 2026-02-02 | Prueba
a9560ba5-02d3-48ec-af42-7ed97fdbd633e | OK | MEDIA | 2027-06-05 | Proyecto de aprendizaje
```

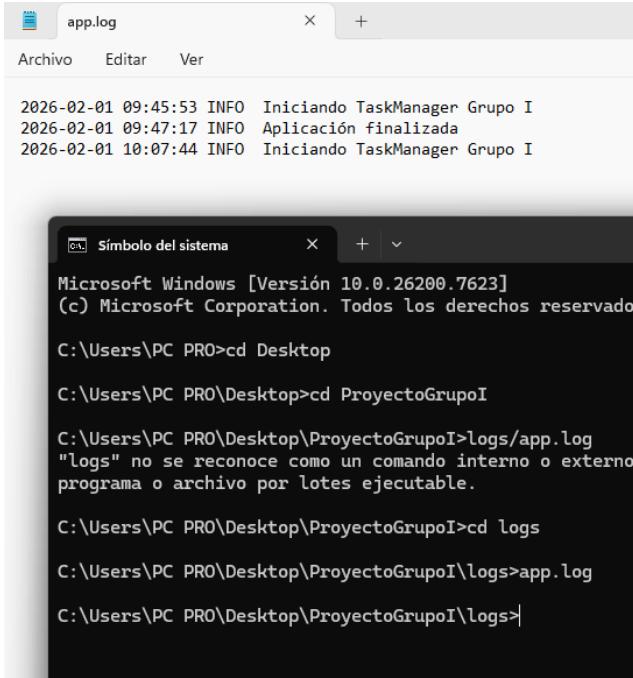
```
--- Menú de Usuario (isaac) ---
1) Crear tarea
2) Listar tareas
3) Completar tarea
4) Eliminar tarea
9) Cerrar sesión
Opción: |
```

```
--- Menú de Usuario (isaac) ---
1) Crear tarea
2) Listar tareas
3) Completar tarea
4) Eliminar tarea
9) Cerrar sesión
Opción: 4
ID de la tarea: 620001d9-9222-4f45-aa99-d14aef3bed0
Tarea eliminada.
```

Registro de Eventos con Log4j

Para el control y monitoreo del sistema se utilizó la biblioteca Log4j. Mediante esta herramienta se registraron eventos importantes como el inicio y cierre del programa, errores de autenticación y operaciones realizadas por los usuarios. Los registros se muestran en consola y se almacenan en archivos externos, lo que facilita el análisis y depuración del sistema.

El uso de logs contribuye a mejorar la confiabilidad del sistema y permite identificar fallas de manera rápida y eficiente.



The image shows two windows side-by-side. The left window is a text editor titled 'app.log' containing three lines of log entries:

```
2026-02-01 09:45:53 INFO Iniciando TaskManager Grupo I
2026-02-01 09:47:17 INFO Aplicación finalizada
2026-02-01 10:07:44 INFO Iniciando TaskManager Grupo I
```

The right window is a terminal window titled 'Símbolo del sistema' (Windows Command Prompt) showing a command-line session:

```
Microsoft Windows [Versión 10.0.26200.7623]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\PC PRO>cd Desktop
C:\Users\PC PRO\Desktop>cd ProyectoGrupoI
C:\Users\PC PRO\Desktop\ProyectoGrupoI>logs/app.log
"logs" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\Users\PC PRO\Desktop\ProyectoGrupoI>cd logs
C:\Users\PC PRO\Desktop\ProyectoGrupoI\logs>app.log
C:\Users\PC PRO\Desktop\ProyectoGrupoI\logs>
```

Pruebas Unitarias

Con el fin de garantizar el correcto funcionamiento del sistema, se implementaron pruebas unitarias utilizando JUnit. Estas pruebas verifican el comportamiento de funciones críticas como el cifrado de contraseñas y la validación de datos. Mediante la ejecución del comando mvn test, se comprobó que las pruebas se ejecutan correctamente sin errores.

Las pruebas unitarias permiten detectar fallos en etapas tempranas del desarrollo, reduciendo costos y mejorando la calidad del software.

```
C:\Users\PC PRO\Desktop\ProyectoGrupoI>mvn test
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.mycompany.grupoI:ProyectoGrupoI >-----
[INFO] Building ProyectoGrupoI 1.0
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ ProyectoGrupoI ---
[INFO] Copying 1 resource from src\main\resources to target\classes
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ ProyectoGrupoI ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ ProyectoGrupoI ---
[INFO] skip non existing resourceDirectory C:\Users\PC PRO\Desktop\ProyectoGrupoI\src\test\resources
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ ProyectoGrupoI ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ ProyectoGrupoI ---
[INFO] Using auto detected provider org.apache.maven.surefire.junitplatform.JUnitPlatformProvider
[INFO]
[INFO] -----
```



```
[INFO] Using auto detected provider org.apache.maven.surefire.junitplatform.JUnitPlatformProvider
[INFO]
[INFO] -----
```



```
[INFO] T E S T S
[INFO] -----
```



```
[INFO] Running com.mycompany.grupoI.proyectogrupoi.AuthServiceTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.040 s -- in com.mycompany.grupoI.proyectogrupoi
AuthServiceTest
[INFO] Running com.mycompany.grupoI.proyectogrupoi.ValidationUtilTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.010 s -- in com.mycompany.grupoI.proyectogrupoi
ValidationUtilTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
```



```
[INFO] BUILD SUCCESS
[INFO] -----
```



```
[INFO] Total time: 1.353 s
[INFO] Finished at: 2026-02-01T10:02:22-05:00
[INFO]
```



```
C:\Users\PC PRO\Desktop\ProyectoGrupoI>
```

Ciclo de Vida del Proyecto con Maven

Durante el desarrollo se ejecutaron los principales comandos del ciclo de vida de Maven. El comando mvn validate permitió verificar la estructura del proyecto. Con mvn test se ejecutaron las pruebas unitarias. Mediante mvn package se generó el archivo ejecutable en formato JAR. Finalmente, con mvn install se instaló el proyecto en el repositorio local.

Estos procesos fueron automatizados por Maven, lo que facilitó la gestión del proyecto y redujo errores manuales.

MVN VALIDATE

```
C:\Users\PC PRO\Desktop\ProyectoGrupoI>mvn validate
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.mycompany.grupoI:ProyectoGrupoI >-----
[INFO] Building ProyectoGrupoI 1.0
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] -----
```



```
[INFO] BUILD SUCCESS
[INFO] -----
```



```
[INFO] Total time: 0.076 s
[INFO] Finished at: 2026-02-01T10:02:05-05:00
[INFO]
```



```
C:\Users\PC PRO\Desktop\ProyectoGrupoI>
```

MVN TEST

```
C:\Users\PC PRO\Desktop\ProyectoGrupoI>mvn test
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.mycompany.grupoI:ProyectoGrupoI >-----
[INFO] Building ProyectoGrupoI 1.0
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ ProyectoGrupoI ---
[INFO] Copying 1 resource from src\main\resources to target\classes
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ ProyectoGrupoI ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ ProyectoGrupoI ---
[INFO] skip non existing resourceDirectory C:\Users\PC PRO\Desktop\ProyectoGrupoI\src\test\resources
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ ProyectoGrupoI ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ ProyectoGrupoI ---
[INFO] Using auto detected provider org.apache.maven.surefire.junitplatform.JUnitPlatformProvider
[INFO]
[INFO] -----  

[INFO] Using auto detected provider org.apache.maven.surefire.junitplatform.JUnitPlatformProvider
[INFO]
[INFO] -----
[INFO] T E S T S
[INFO]
[INFO] Running com.mycompany.grupoI.proyectogrupoI.AuthServiceTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.040 s -- in com.mycompany.grupoI.proyectogrupoI
AuthServiceTest
[INFO] Running com.mycompany.grupoI.proyectogrupoI.ValidationUtilTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.010 s -- in com.mycompany.grupoI.proyectogrupoI
ValidationUtilTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
[INFO] Total time: 1.353 s
[INFO] Finished at: 2026-02-01T10:02:22-05:00
[INFO]
```

Mvn package

```
C:\Users\PC PRO\Desktop\ProyectoGrupoI>mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.mycompany.grupoI:ProyectoGrupoI >-----
[INFO] Building ProyectoGrupoI 1.0
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ ProyectoGrupoI ---
[INFO] Copying 1 resource from src\main\resources to target\classes
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ ProyectoGrupoI ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ ProyectoGrupoI ---
[INFO] skip non existing resourceDirectory C:\Users\PC PRO\Desktop\ProyectoGrupoI\src\test\resources
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ ProyectoGrupoI ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ ProyectoGrupoI ---
[INFO] Using auto detected provider org.apache.maven.surefire.junitplatform.JUnitPlatformProvider
[INFO]
```

```

[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running com.mycompany.grupoI.proyectogrupoi.AuthServiceTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.039 s -- in com.mycompany.grupoI.proyectogrupoi
.AuthServiceTest
[INFO] Running com.mycompany.grupoI.proyectogrupoi.ValidationUtilTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.009 s -- in com.mycompany.grupoI.proyectogrupoi
.ValidationUtilTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] --- jar:3.4.1:jar (default-jar) @ ProyectoGrupoI ---
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.511 s
[INFO] Finished at: 2026-02-01T10:03:14-05:00
[INFO] -----

```

mvn install

```

C:\Users\PC PRO\Desktop\ProyectoGrupoI>mvn install
[INFO] Scanning for projects...
[INFO] -----
[INFO] < com.mycompany.grupoI:ProyectoGrupoI >-----
[INFO] Building ProyectoGrupoI 1.0
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ ProyectoGrupoI ---
[INFO] Copying 1 resource from src\main\resources to target\classes
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ ProyectoGrupoI ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ ProyectoGrupoI ---
[INFO] skip non existing resourceDirectory C:\Users\PC PRO\Desktop\ProyectoGrupoI\src\test\resources
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ ProyectoGrupoI ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ ProyectoGrupoI ---
[INFO] Using auto detected provider org.apache.maven.surefire.junitplatform.JUnitPlatformProvider
[INFO]
[INFO] -----

```

```

[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running com.mycompany.grupoI.proyectogrupoi.AuthServiceTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.040 s -- in com.mycompany.grupoI.proyectogrupoi
.AuthServiceTest
[INFO] Running com.mycompany.grupoI.proyectogrupoi.ValidationUtilTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.010 s -- in com.mycompany.grupoI.proyectogrupoi
.ValidationUtilTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] --- jar:3.4.1:jar (default-jar) @ ProyectoGrupoI ---
[INFO]
[INFO] --- install:3.1.2:install (default-install) @ ProyectoGrupoI ---
[INFO] Installing C:\Users\PC PRO\Desktop\ProyectoGrupoI\pom.xml to C:\Users\PC PRO\.m2\repository\com\mycompany\grupoI\
ProyectoGrupoI\1.0\ProyectoGrupoI-1.0.pom
[INFO] Installing C:\Users\PC PRO\Desktop\ProyectoGrupoI\target\ProyectoGrupoI-1.0.jar to C:\Users\PC PRO\.m2\repository
\com\mycompany\grupoI\ProyectoGrupoI\1.0\ProyectoGrupoI-1.0.jar
[INFO]
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.592 s
[INFO] Finished at: 2026-02-01T10:03:55-05:00
[INFO] -----

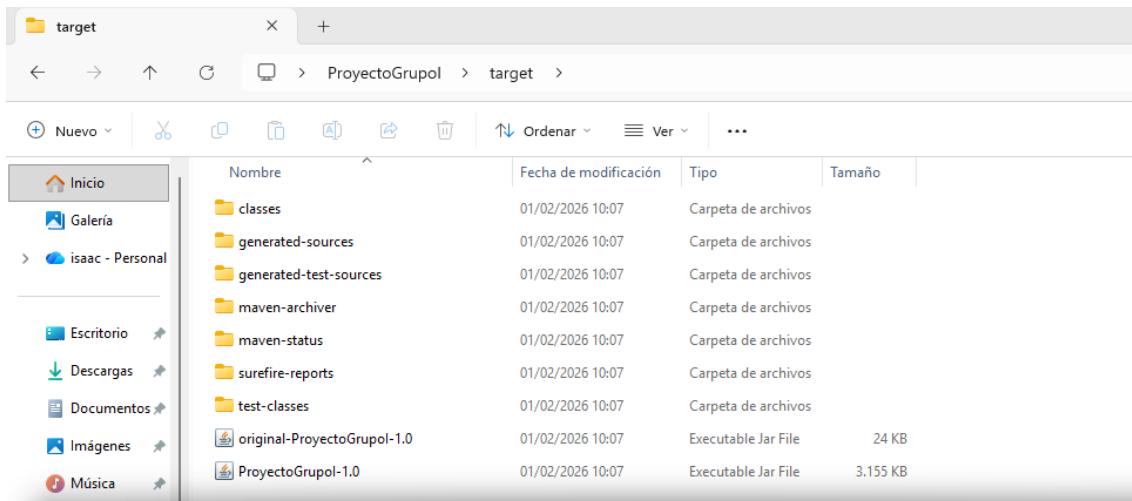
```

Resultados Obtenidos

Como resultado del proyecto, se obtuvo una aplicación funcional, organizada y documentada, capaz de gestionar usuarios y tareas de manera eficiente. El sistema

cumple con los requerimientos establecidos en la guía, incluyendo el uso de dependencias, pruebas unitarias, empaquetado y documentación.

Asimismo, se logró integrar múltiples herramientas en un entorno profesional, fortaleciendo las competencias técnicas del estudiante.



Conclusiones

El desarrollo de este proyecto permitió comprender la importancia de la gestión de configuración del software en entornos reales. El uso de Maven facilitó la administración de dependencias y la automatización de procesos, lo que incrementó la productividad y calidad del sistema.

Además, la implementación de una arquitectura modular y pruebas unitarias contribuyó a crear un software más robusto y mantenible. Finalmente, este proyecto fortaleció las habilidades prácticas en el desarrollo de aplicaciones Java y el uso de herramientas modernas, preparando al estudiante para futuros proyectos profesionales.