¡Claro! Entendido perfectamente. La confusión es mía: **tienes los códigos fuente de todo el proyecto, y no has generado ningún JAR ni tienes la librería datastructures-library como un JAR precompilado.** Solo tienes el código fuente de ambas.

Esto simplifica un poco la parte de la datastructures-library porque no hay que instalar un JAR con mvn install:install-file, sino que se construye e instala desde su propio código fuente.

Aquí te detallo cómo deberías preparar y compartir tu proyecto, enfatizando que ambos son proyectos Maven basados en código fuente:

### **Cómo Compartir el Proyecto Completo (Código Fuente de Ambos Proyectos Maven)**

Para que otra persona (o la cátedra) pueda configurar y ejecutar tu proyecto simulador-colectivos que depende de datastructures-library, la clave es que ambos proyectos se gestionen como **proyectos Maven de código fuente**.

El proceso implica que la persona que reciba los archivos deberá construir e instalar la librería datastructures-library **primero**, para que luego tu proyecto simulador-colectivos pueda encontrarla en su repositorio Maven local.

### **1. Preparar y Comprimir los Archivos para Compartir**

Debes comprimir (ej. en .zip o .tar.gz) **ambas carpetas de proyecto completas**: datastructures-library y simulador-colectivos.

**Estructura del .zip o .tar.gz que compartirías:**

📦 tu\_proyecto\_simulacion.zip (o .tar.gz)  
├── 📁 datastructures-library/ (Carpeta completa del proyecto de la librería)  
│ ├── pom.xml  
│ └── src/  
│ └── main/  
│ └── java/  
│ └── net/datastructures/ (con todos los .java de la librería)  
│ └── ... (otros archivos de proyecto de la librería)  
├── 📁 simulador-colectivos/ (Carpeta completa de tu proyecto principal)  
│ ├── pom.xml  
│ └── src/  
│ ├── main/  
│ │ ├── java/ (tus paquetes y clases)  
│ │ └── resources/ (tus .properties, .txt, .md)  
│ └── test/  
│ ├── java/ (tus tests)  
│ └── resources/ (tus archivos de test)  
│ └── ... (otros archivos de tu proyecto principal)  
└── 📄 README\_GLOBAL.md (o INSTRUCCIONES.txt - \*\*muy recomendado\*\*)

**Lo que NO necesitas compartir:**

* Las carpetas target/ dentro de cualquiera de los proyectos (Maven las recreará).
* Archivos o carpetas específicas del IDE (ej. .settings/, .project, .classpath, .idea/). Maven se encargará de configurar esto.

### **2. Instrucciones Detalladas para el Receptor (Paso a Paso)**

Esta es la parte más importante. Debes proporcionar un documento claro (como un README\_GLOBAL.md en la raíz del .zip comprimido) que guíe a la persona que va a usar tu proyecto.

**Contenido sugerido para el README\_GLOBAL.md:**

Markdown

# Proyectos del Simulador de Colectivos Urbanos  
  
Este archivo contiene dos proyectos Maven de código fuente:  
1. `datastructures-library`: La librería de estructuras de datos utilizada por el proyecto principal.  
2. `simulador-colectivos`: El proyecto principal del simulador.  
  
## Requisitos del Sistema  
  
\* **\*\*Java Development Kit (JDK):\*\*** Versión 21 o superior.  
\* **\*\*Apache Maven:\*\*** Versión 3.x o superior.  
\* **\*\*IDE (Recomendado):\*\*** Eclipse (versión reciente).  
  
## Configuración y Ejecución de los Proyectos  
  
Siga estos pasos para configurar y ejecutar la simulación. **\*\*Es crucial seguir el orden:\*\***  
  
### Paso 1: Instalar la Librería `datastructures-library`  
  
Esta librería **\*\*no se encuentra en repositorios Maven públicos\*\***. Debe ser compilada e instalada localmente en su repositorio Maven para que el proyecto principal pueda encontrarla.  
  
1. Abra una **\*\*terminal\*\*** (o el terminal integrado de su IDE, como Eclipse o VS Code).  
2. Navegue hasta la carpeta raíz del proyecto `datastructures-library` (la que acaba de descomprimir, donde se encuentra su `pom.xml`).  
 \* Ejemplo: `cd /ruta/donde/descomprimio/datastructures-library`  
3. Ejecute el siguiente comando Maven para compilar e instalar la librería:  
 ```bash  
 mvn clean install  
 ```  
 \* Esto instalará el artefacto `net.datastructures:datastructures-library:6.0.0-custom` en su repositorio Maven local (`~/.m2/repository/`).  
  
### Paso 2: Importar el Proyecto `simulador-colectivos` en Eclipse  
  
Una vez instalada la librería, puede importar el proyecto principal en su IDE.  
  
1. Abra Eclipse IDE.  
2. Vaya a `File > Import...`.  
3. Seleccione `Maven > Existing Maven Projects`. Clic en `Next`.  
4. En "Browse...", navegue hasta la carpeta raíz de su proyecto `simulador-colectivos` (la que acaba de descomprimir, donde se encuentra su `pom.xml`).  
5. Asegúrese de que el `pom.xml` del proyecto sea detectado y marcado. Clic en `Finish`.  
  
### Paso 3: Actualizar el Proyecto Maven `simulador-colectivos` en Eclipse  
  
Después de importar, es crucial forzar a Maven a resolver las dependencias, incluida la que acaba de instalar localmente.  
  
1. En el `Package Explorer` de Eclipse, haga clic derecho sobre el proyecto `simulador-colectivos`.  
2. Seleccione `Maven > Update Project...`.  
3. Asegúrese de que `simulador-colectivos` esté marcado y **\*\*active la casilla "Force Update of Snapshots/Releases"\*\***.  
4. Clic en `OK`.  
 \* Este paso es vital para que Maven reconozca la `datastructures-library` instalada localmente.  
  
### Paso 4: Ejecutar las Pruebas Unitarias (Verificación)  
  
Para verificar que el proyecto fue configurado correctamente y todas las dependencias están resueltas:  
  
1. En Eclipse, navegue a `src/test/java/ar.edu.unpsjb.ayp2.proyectointegrador.datos/LectorArchivosTest.java`.  
2. Haga clic derecho sobre el archivo en el editor o en el `Package Explorer`.  
3. Seleccione `Run As > JUnit Test`.  
 \* Debería ver una barra **\*\*verde\*\*** en la vista JUnit, indicando que todas las pruebas pasaron. Esto confirma que el proyecto está correctamente configurado.  
  
### Paso 5: Ejecutar la Aplicación Principal  
  
1. En Eclipse, navegue a `src/main/java/ar.edu.unpsjb.ayp2.proyectointegrador.interfaz/SimuladorColectivosApp.java`.  
2. Haga clic derecho sobre el archivo.  
3. Seleccione `Run As > Java Application`.  
 \* La simulación se ejecutará y mostrará la salida en la consola.  
  
---  
  
Este plan detallado guiará a cualquier persona a configurar y ejecutar tu proyecto sin problemas, manejando la dependencia local de `datastructures-library` de manera efectiva.