MANUAL TÉCNICO

Tractor Game – Sistema de Gestión de Granja – Grupo 4

Integrantes del Grupo 4

- Mateo Viuete
- Daniela Vivanco
- Jean Zambrano
- David Zambrano
- David Zapata
- Fabian Tumailli

Índice

Descripción General

Tractor Game es una aplicación escrita en Java diseñada para la gestión de una granja, permitiendo el registro y control de animales, productos, ventas, compras y reportes. El sistema está estructurado para facilitar el mantenimiento y la extensión de sus funcionalidades.

Estructura del Proyecto

La estructura típica del proyecto es:

GRANJA-GROUP04/ │ ├── src/ │ ├── (archivos .java: clases principales, modelos, controladores, vistas) │ ├── bin/ │ ├── (archivos .class compilados) │ ├── README.md ├── MANUAL_USUARIO.md ├── MANUAL_TECNICO.md └── (otros archivos de configuración)

Sugerencia de organización en src/:

- *model/*: Clases de modelo/datos
- *controller/*: Lógica de negocio y controladores

- *view/*: Interfaces gráficas o menús de consola

Requisitos Técnicos

- *Lenguaje:* Java 8 o superior
- *IDE recomendado:* Eclipse, NetBeans, IntelliJ IDEA o VS Code
- *Sistema Operativo:* Windows, Linux o MacOS
- *Herramientas adicionales:* Ninguna obligatoria, salvo que se use base de datos externa

---

Instalación y Configuración

1. *Clonar el repositorio:* bash git clone https://github.com/Mateoow/GRANJA-GROUP04.git
2. *Importar el proyecto a tu IDE favorito* (como proyecto Java estándar).
3. *Verificar dependencias:*
   Si hay librerías externas (por ejemplo, para interfaz gráfica, manejo de archivos o base de datos), deben estar incluidas en el proyecto, normalmente en una carpeta lib/ o gestionadas por Maven/Gradle.
4. *Configurar la base de datos (si aplica):*
   - Si usa archivos planos: No requiere configuración adicional.
   - Si usa una base de datos (ej.: SQLite, MySQL), configurar la conexión en el archivo de propiedades o en la clase correspondiente.

Compilación y Ejecución

1. *Compilar manualmente:* bash javac -d bin src/*.java
2. *Ejecutar el sistema:* bash java -cp bin NombreClasePrincipal
   - *Nota:* Reemplaza NombreClasePrincipal por el nombre real de la clase que contiene el método main.
3. *Desde el IDE:*
   Ejecuta el archivo principal (usualmente Main.java o App.java).

Principales Componentes y Clases

- *Clases de modelo (model):*
  Representan entidades como Animal, Producto, Venta, Compra, Usuario, etc.
- *Controladores (controller):*
  Gestionan la lógica de negocio, validaciones y manipulación de datos.
- *Vistas (view):*
  Encargadas de la interacción con el usuario (por consola o interfaz gráfica).

- *Persistencia:*
  Puede ser mediante archivos planos (txt, csv) o una base de datos relacional (según la implementación).

Ejemplo de clases típicas:

- Animal.java
- Producto.java
- Venta.java
- Compra.java
- Usuario.java
- TractorGame.java (clase principal)

Gestión de Datos

- *Archivos planos:*
  Los datos pueden almacenarse en archivos de texto ubicados en una carpeta específica (ej.: data/).
- *Base de datos:*
  Si se utiliza, la conexión y credenciales deben configurarse correctamente.
  Ejemplo de conexión a SQLite: java Connection conn =
  DriverManager.getConnection("jdbc:sqlite:granja.db");
- *Serialización:*
  En algunos casos, se pueden usar objetos serializables para guardar el estado del sistema.

Mantenimiento y Extensión

- *Agregar nuevas entidades:*
  Crear una clase de modelo, ampliar el controlador y la vista correspondiente.
- *Modificar lógica de negocio:*
  Realizar cambios en los controladores o servicios asociados.
- *Cambiar almacenamiento:*
  Si se necesita migrar de archivos a base de datos, adaptar las clases de persistencia.
- *Pruebas:*
  Se recomienda agregar pruebas unitarias para nuevas funcionalidades.

Resolución de Problemas

- *Error al compilar:*
  Verifica que la versión de Java sea la adecuada y que todas las clases estén en el directorio correcto.
- *No se abre el sistema:*
  Asegúrate de ejecutar la clase principal y que todas las dependencias estén presentes.

- *Problemas de datos:*
  Revisa los archivos de datos o la configuración de la base de datos.

Contacto y Soporte

Para dudas técnicas, errores o sugerencias, contactar a cualquiera de los integrantes del Grupo 4, o revisar los canales de soporte indicados en la documentación interna.
