

Catalogo de equipos de finca

Análisis del Sistema (30 puntos)

1.1 Requisitos funcionales del sistema (5 pts)

Escribir los puntos exactos que debe cumplir tu sistema.

Ejemplo: "El usuario puede crear una nueva solicitud de boletos", etc.

- El sistema debe ser capaz de generar un catalogo de 10 equipos aleatorios al momento de iniciarse
- El sistema debe ser capaz de ordenar estos 10 equipos, en base a su consumo eléctrico por uso
- El sistema debe ser capaz de listar los 10 equipos, con todas sus características
- El sistema debe ser capaz de buscar uno de los 10 equipos en base a su ID o su nombre, y mostrar toda la información correspondiente
- El sistema debe ser capaz de soportar la creación de una nueva clase de objeto (Dron), la cual pueda usarse en el catalogo inicial y no presente mayores problemas

1.2 Clases necesarias y su propósito (5 pts)

Clase	Propósito
Equipo tecnológico	Representa a todos los equipos tecnológicos que la cooperativa usa. Implementa el contrato medible, actionable y registrable
Sensor de suelos	Representa los sensores de suelos que la cooperativa usa en sus operaciones. Hereda de la clase de Equipo tecnológico.
Dron de imágenes	Representa los drones de imágenes que la cooperativa usa en sus operaciones. Hereda de la clase de Equipo tecnológico.
Sensor de atmósfera	Representa los sensores de atmósfera que la cooperativa usa en sus operaciones. Hereda de la clase de Equipo tecnológico.

Controlador	Esta clase se encarga de manejar las operaciones del sistema, como la creación de equipos, la búsqueda de equipos y su ordenamiento.
Consola	Esta clase se encarga de mostrar y pedir información al usuario, conforme se requiera.
Main	Es la clase principal, la cual lleva el flujo de procesos del programa.

Interfaz	Propósito
Medible	Es una interfaz que contiene los métodos de medición que cualquier equipo tecnológico puede usar.
Accionable	Es una interfaz que contiene los métodos de acción física que cualquier equipo tecnológico puede usar.
Registrable	Es una interfaz que contiene los métodos de registro de datos que cualquier equipo tecnológico puede usar.

1.3 Atributos de cada clase (10 pts)

Clase: Equipo tecnológico

Atributo	Tipo de dato	Visibilidad	Propósito
Id	Int	Protegida	Es un numero que funciona como identificador del equipo
Nombre	String	Protegida	Es el nombre del equipo tecnológico. Funciona también como identificador.
Marca	String	Protegida	Es la marca que creó el equipo tecnológico.
Consumo de energía por hora	Float	Protegida	Es la cantidad de energía eléctrica que el equipo consume por hora.
Vida útil restante	Int	Protegida	Es la cantidad de meses restantes en la vida útil del equipo.

Clase: Sensor de suelos

Atributo	Tipo de dato	Visibilidad	Propósito
Tipo de terreno	Int	Privada	Es un número que funciona como identificador del equipo

--	--	--	--

Clase: Dron de imágenes

Atributo	Tipo de dato	Visibilidad	Propósito
Capacidad de memoria	Int	Privada	Es la cantidad máxima de fotografías que el dron puede almacenar

Clase: Sensor de atmósfera

Atributo	Tipo de dato	Visibilidad	Propósito
Ambiente	Int	Privada	Es el tipo de ambiente en el que el sensor se encuentra. Se basa en el siguiente código: 1 = Jungla 2 = Montaña 3 = Playa

Clase: Medible

Atributo	Tipo de dato	Visibilidad	Propósito
Ninguno			

Clase: Accionable

Atributo	Tipo de dato	Visibilidad	Propósito
Ninguno			

Clase: Controlador

Atributo	Tipo de dato	Visibilidad	Propósito

Lista de dispositivos			
-----------------------	--	--	--

Clase: Consola

Atributo	Tipo de dato	Visibilidad	Propósito
Teclado	Scanner	privada	Es el método con el cual la consola muestra mensajes al usuario.

Clase: Main

Atributo	Tipo de dato	Visibilidad	Propósito
Consola	Consola	Privada	Es una referencia a un objeto de consola, con el cual el programa se comunica con el usuario
Controlador	Controlador	Privada	Es un controlador encargado de manejar los datos del programa

1.4 Métodos de cada clase (10 pts)

Clase: Equipo tecnológico

Método	Parámetros : Tipo de dato	Salida: Tipo de dato	Visibilidad	Propósito
toString	ninguno	String	Publica	Crea una lista con las características del equipo

Clase: Sensor de suelos

Método	Parámetros : Tipo de dato	Salida: Tipo de dato	Visibilidad	Propósito
ninguno				

Clase: Dron de imágenes

Método	Parámetros : Tipo de dato	Salida: Tipo de dato	Visibilidad	Propósito
ninguno				

Clase: Sensor de atmósfera

Método	Parámetros : Tipo de dato	Salida: Tipo de dato	Visibilidad	Propósito

ninguno				
---------	--	--	--	--

Clase: Medible

Método	Parámetros : Tipo de dato	Salida: Tipo de dato	Visibilidad	Propósito
Medir altitud	ninguno	Int	Publica	Da la cantidad de metros sobre el suelo en donde se encuentra el equipo.
Medir temperatura	ninguno	Int	Publica	Da la cantidad de grados Celsius actuales en el ambiente del sensor.

Clase: Accionable

Método	Parámetros : Tipo de dato	Salida: Tipo de dato	Visibilidad	Propósito
Activar valvulas	ArrayList<String>: Lista de válvulas ArrayList<Boolean>: Lista de estado de válvulas	boolean	Publica	Activa o desactiva las válvulas de la cooperativa, según corresponda
Activar señal de emergencia	Nombre de alarma: String	String	Publica	Activa una señal de emergencia

Clase: Registrable

Método	Parámetros : Tipo de dato	Salida: Tipo de dato	Visibilidad	Propósito
Guardar datos	Base de datos: String	String	Publica	Guarda el calculo de un dato en la base de datos indicada

Clase: Controlador

Método	Parámetros : Tipo de dato	Salida: Tipo de dato	Visibilidad	Propósito
Generar dispositivos	Cantidad de dispositivos: int	void	Publica	Genera la cantidad de dispositivos indicado en el parámetro. Estos dispositivos se guardan en una lista, la cual no se puede modificar
Mostrar dispositivos	Ninguno	ArrayList <String>	Publica	Genera una lista de mensajes, en donde cada elemento representa las características de un dispositivo
Buscar dispositivo por nombre	Nombre de dispositivo: String	String	Publica	Busca un dispositivo del sistema en base a su nombre. Si no se encuentra, se regresa null. De lo contrario, se muestra un resumen de las características del dispositivo.

Buscar dispositivo por ID	ID de dispositivo: int	String	Publica	Busca un dispositivo del sistema en base a su ID. Si no se encuentra, se regresa null. De lo contrario, se muestra un resumen de las características del dispositivo.
Ordenar dispositivos	ninguno	ArrayList <String>	Publica	Genera una copia de la lista de dispositivos del sistema, y la ordena en base a su consumo de energía. Después, regresa una lista con los nombres de los dispositivos, en base a el orden de la nueva lista.

Clase: Consola

Método	Parámetros : Tipo de dato	Salida: Tipo de dato	Visibilidad	Propósito
Mostrar características de tarea	Información de tarea: String	void	Publica	Muestra las características principales de una tarea
Pedir nombre de tarea	Ninguno	String	Publica	Le solicita al usuario el nombre de una tarea
Pedir ID de tarea	Ninguno	Int	Publica	Le solicita al usuario el ID de una tarea

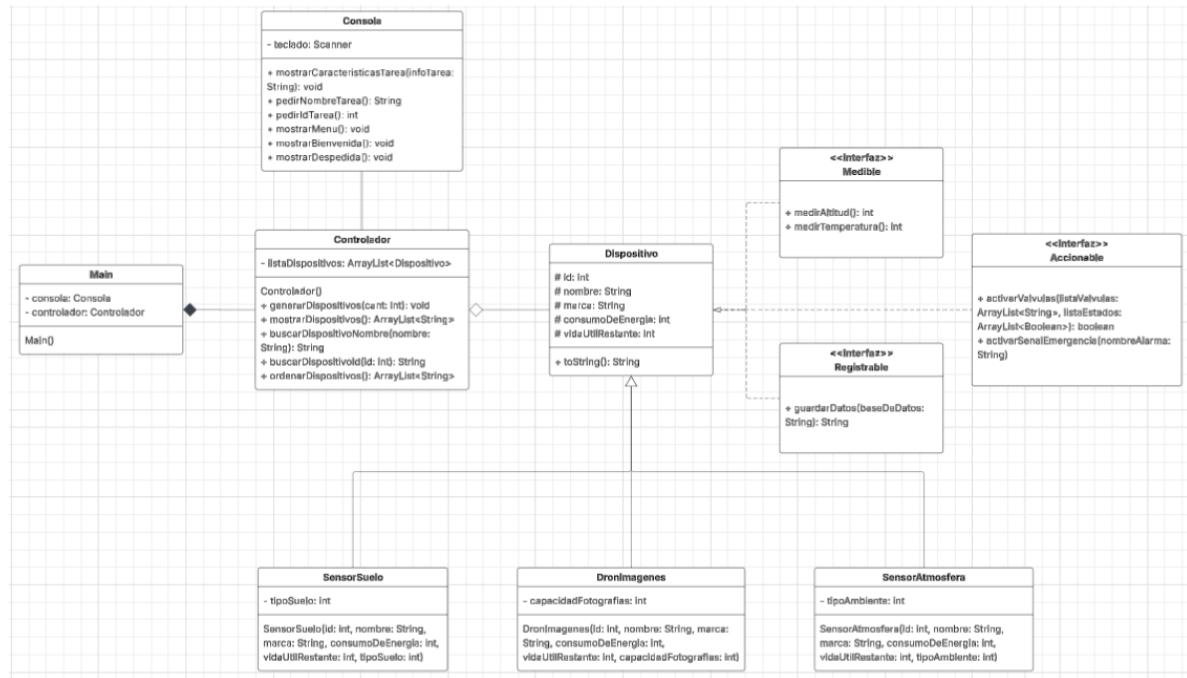
Mostrar menú principal	Ninguno	Void	Publica	Muestra las opciones con las que cuenta el menú del programa
Mostrar mensaje de bienvenida	Ninguno	Void	Publica	Muestra un mensaje de bienvenida al programa al usuario
Mostrar mensaje de despedida	Ninguno	void	Publica	Muestra un mensaje de despedida al usuario

Clase: Main

Método	Parámetros : Tipo de dato	Salida: Tipo de dato	Visibilidad	Propósito
ninguno				

2. Diseño: Diagrama de Clases (30 puntos)

- Asegúrate de mostrar atributos y métodos con visibilidad (+, -).
- Indica relaciones entre clases (asociación, agregación, etc.).
- Incluye el driver program (Main).



3. Programa (40 puntos)

En cada archivo `java`, asegurarse de incluir:

- Las clases necesarias.
- Uso adecuado de objetos.

Menú que debe implementar el driver program:

1. Nuevo comprador
2. Nueva solicitud de boletos
3. ...
4. ...
5. Salir

GitHub: colocar aquí la URL:

<https://github.com/Oscarro11/Equipos-de-finca>

Checklist antes de entregar

Está claro el análisis?

El diagrama tiene los elementos UML correctamente?

Subiste tu código a GitHub con todo lo necesario?