

Ejercicio_6_Polimorfismo_via_Interfaces

Análisis del Sistema (30 puntos)

1.1 Requisitos funcionales del sistema (5 pts)

Escribir los puntos exactos que debe cumplir tu sistema.

- Debe ser capaz de hacer uso de interfaces para el control de ciertas funcionalidades de las clases.
- Debe hacer uso de herencia para ordenamiento de los equipos.
- Debe ser capaz de añadir objetos más objetos a la lista de equipos.

1.2 Clases necesarias y su propósito (5 pts)

Clase	Propósito
Main	Clase de inicializador del programa.
Consola	Manejo de interfaz de interacción con el usuario.
Inventario	Clase principal donde se manejará la lista de objetos y métodos de interacción.
Equipos	Clase padre que recibirá las funcionalidades generales que pueden tener.
Dron	Clase hija con los procesos específicos que puede realizar el dron.
Sensores	Clase hija con los procesos específicos que puede realizar el sensor de suelo.
Estaciones	Clase hija con los procesos específicos que puede realizar la estación meteorológica.
Persona	Clase padre con información de las personas con acceso.
Jefe	Clase hija de persona con los permisos y acciones que puede realizar.
Trabajador	Clase hija de persona con los permisos y acciones que puede realizar.

Interfaz	Propósito
Medible	Contiene los métodos medibles que luego serán modificados con overload y override.
Accionable	Contiene los métodos accionables que luego serán modificados con overload y override.
Registrable	Contiene los métodos registrables que luego

serán modificados con overload y override.

1.3 Atributos de cada clase (10 pts)

Inventario:

Atributo	Tipo de dato	Visibilidad	Propósito
personal	ArrayList<Persona>	Private	Lista de personal disponible.
equipos	ArrayList<Equipos>	Private	Lista de equipos actuales.
permisosActuales	boolean	Private	Comprobación permisos especiales.
consolita	Consola	Private	Manejo interacción con usuario.

Equipos:

Atributo	Tipo de dato	Visibilidad	Propósito
ID	String	Public	Identificadores del equipo
nombre	String	Public	
permisoMedible	boolean	Public	
permisoAccionable	boolean	Public	Controladores de que acciones puede realizar el equipo.
permisoRegistrable	boolean	Public	

Persona:

Atributo	Tipo de dato	Visibilidad	Propósito
Nombre	String	Public	Datos específicos de cada persona dentro del sistema.
Usuario	String	Public	
Ccontraseña	String	Public	
permisos	boolean	Public	

1.4 Métodos de cada clase (10 pts)

Inventario:

Atributo	Método	Parámetros : Tipo de dato	Visibilidad	Propósito
Void	IniciarSesion	0	Public	Comprobar/ cambiar credenciales
Void	CerrarSesion	0	Public	
Void	AgregarEquipo	0	Public	
Equipo	BusquedaPorID	(String ID)	Public	Buscar equipo por variable
Equipo	BusquedaPorNombre	(String nombre)	Public	
Void	RealizarAccion	0	Public	Realizar acción de equipo

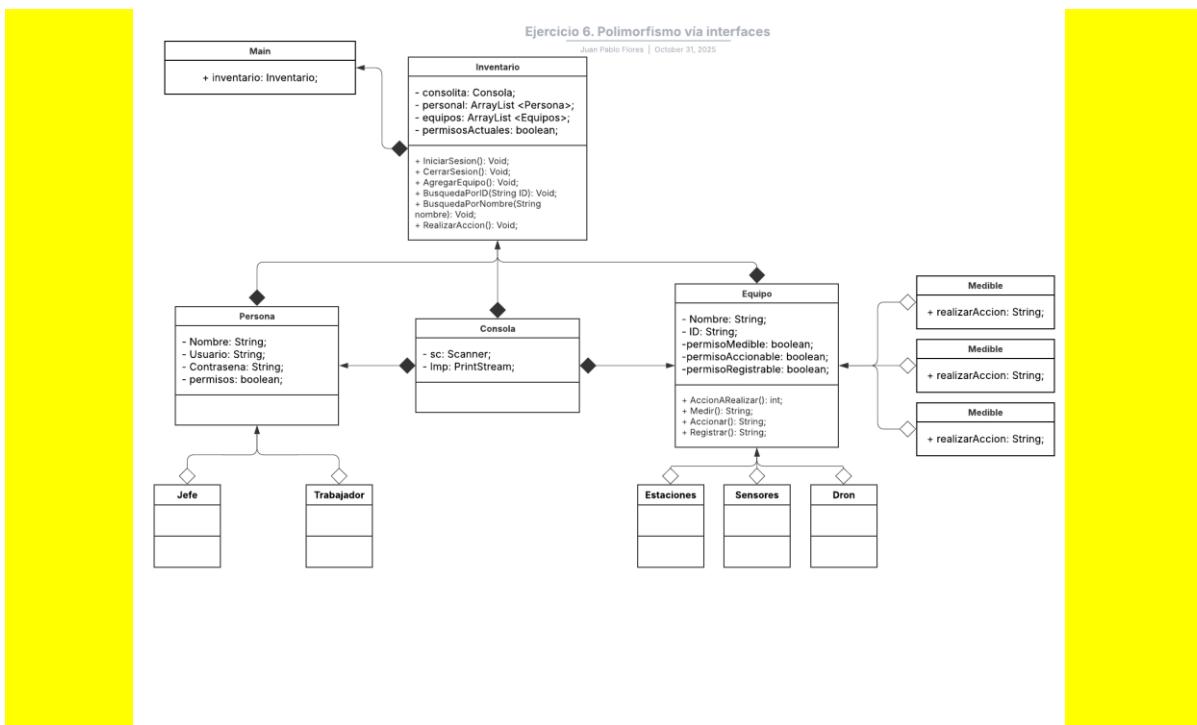
Equipos:

Atributo	Método	Parámetros : Tipo de dato	Visibilidad	Propósito
int	AccionARrealizar	0	Public	Valor pareado para determinar acción a realizar.
String	Medir	0	Public	
String	Accionar	0	Public	
String	Registrar	0	Public	

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).

https://lucid.app/lucidchart/97058b58-0f6f-4c1f-8b6a-ed8a291cd76e/edit?viewport_loc=-167%2C147%2C2601%2C1157%2CHWEp-vi-RSFO&invitationId=inv_d652c6ff-cd64-45bc-8e5a-a1aa2b3e52a8



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/Hola2212/Ejercicio_6_Polimorfismo_via_Interfaces.git

Checklist antes de entregar

- Est醤 claro el an醘isis?
- El diagrama tiene los elementos UML correctamente?
- Subiste tu c醼digo a GitHub con todo lo necesario?