Ejercicio\_6\_Polimorfismo\_via\_Interfaces

# Análisis del Sistema (30 puntos)

## 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.

## 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. **Atributos de cada clase (10 pts)**

Inventario:

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

Equipos:

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

Persona:

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

## Métodos de cada clase (10 pts)

Inventario:

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

Equipos:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo | Método | Parámetros : Tipo de dato | Visibilidad | Propósito |
| int  String  String  String | AccionARealizar  Medir  Accionar  Registrar | ()  ()  ()  () | Public  Public  Public  Public | Valor pareado para determinar acción a realizar. |

# 

1. **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>

A diagram of a company

AI-generated content may be incorrect.

# 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

⬛✓⬛ Está claro el análisis?

⬛✓⬛ El diagrama tiene los elementos UML correctamente?

⬛✓ Subiste tu código a GitHub con todo lo necesario?