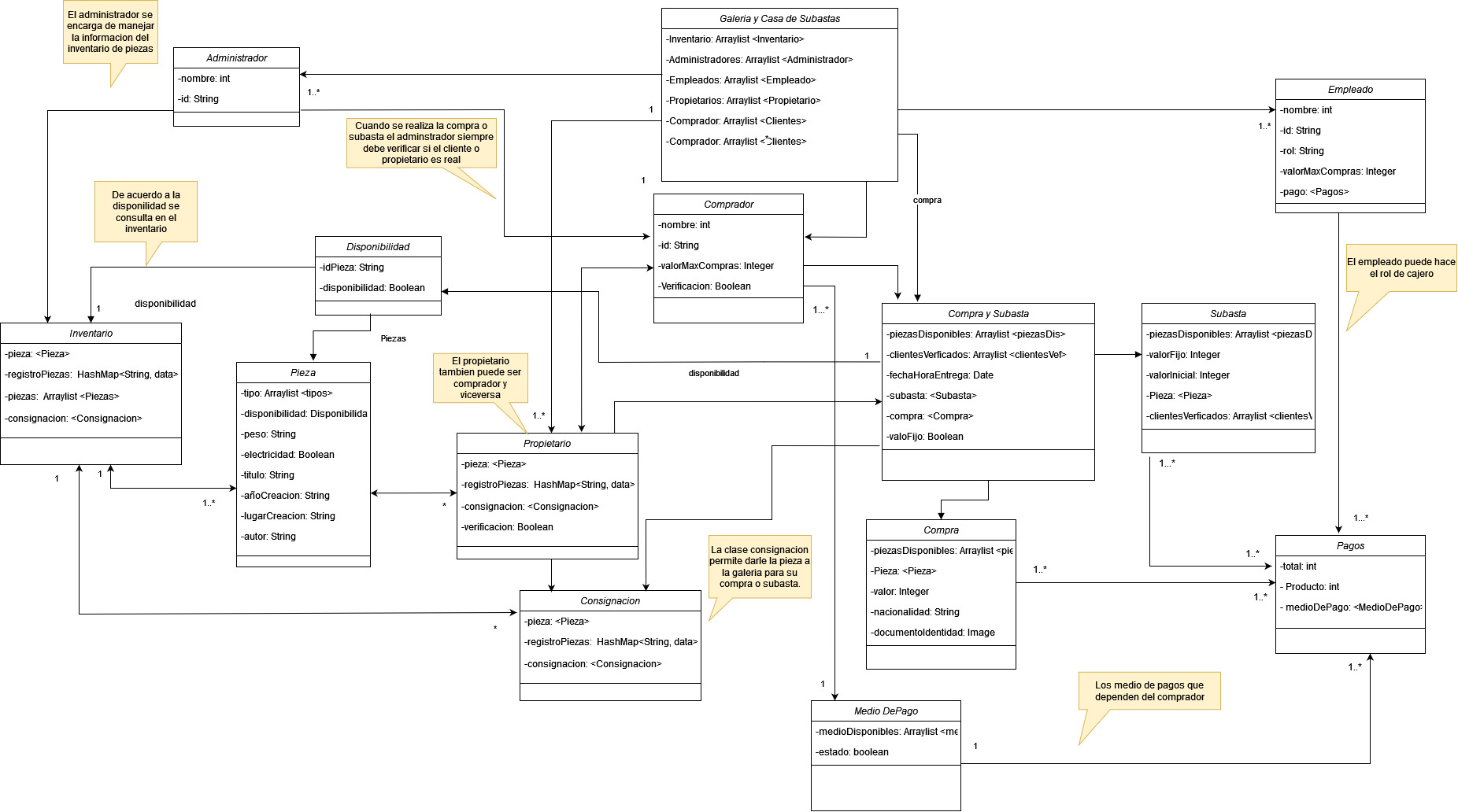
**Proyecto 1 - Análisis**

1. **Modelo de dominio**



**2. Restricciones**

**- Restricciones Funcionales:**

**Autenticación de Usuarios:**

* + Los usuarios deben autenticarse utilizando un nombre de usuario y una contraseña válidos para acceder a las funcionalidades del sistema.

**Roles y Permisos:**

* + Los diferentes roles de usuario (administrador, empleado, comprador, propietario) tendrán acceso a diferentes funcionalidades del sistema, según sus permisos.

**Gestión de Inventarios:**

* + Solo el administrador de la galería puede agregar nuevas piezas al inventario y actualizar su información.
  + Los empleados pueden realizar acciones específicas sobre las piezas del inventario, como actualizar su estado o detalles.

**Compra y Subasta:**

* + Solo los compradores registrados y verificados pueden participar en subastas o realizar ofertas de compra.
  + Las piezas estarán disponibles para compra directa o subasta, y el sistema debe gestionar adecuadamente estas dos modalidades de venta.

**Gestión de Pagos:**

* + Los pagos deben ser registrados adecuadamente en el sistema antes de entregar la obra de arte correspondiente al comprador.
  + Se deben proporcionar diferentes opciones de pago, como tarjeta de crédito, transferencia electrónica y efectivo.
  + Un empleado registrado como cajero también puede registrar el pago de una pieza.

**- Restricciones no Funcionales:**

1. **Seguridad:**
   * Los datos de los usuarios, incluidos los detalles de contacto y los registros de transacciones, deben almacenarse de forma segura para que se accedan de manera estricta.
2. **Usabilidad:**
   * La interfaz de usuario debe ser intuitiva y fácil de usar para todos los tipos de usuarios, incluidos administradores, empleados y compradores.
3. **Rendimiento:**
   * El sistema no debe colapsar por él nivel de información que estaremos manejando.

**3. Descripción programa:**

Según el modelo de dominio podemos ver la clase *Galería y clase de subasta* que funciona como “***controlador****”*, de acuerdo a un modelo MVC, (aunque claramente tendríamos que hacer más clases para un package controlador) esta clase controla la información que se maneja desde la interfaz gráfica, es decir los datos que deben agregar los usuarios de la aplicación, redirige a la lógica de las clases de los usuarios como; Administrador, comprador, empleado y propietario, y también la lógica de la compra y subasta y los requerimientos para cada uno.

El administrador, se encarga de poder manejar todo lo que tiene que ver con el inventario de las piezas y su modificación, es decir si ha sido vendida o su respectiva devolución, esto lo realiza en la parte lógica del inventario para tambien guardarlo en la parte de persistencia de datos, el administrador también tiene el deber de verificar a los usuarios esta información también tendrá persistencia en la parte de compra y subasta

El comprador y el propietario de una pieza pueden compartir sus roles dependiendo de la compra, el comprador se puede volver un propietario y el propietario también puede comprar, esta información también se guarda en la persistencia de la pieza y se tiene en cuenta en la de la *compra y subasta.* Dependiendo del rol ambos pueden hacer distintas funciones, si el usuario ya es propietario puede hacer una *consignación* es decir dar su pieza en una subasta, así como también hacer el proceso de comprar otras, todo esto debería estar en la interfaz de ambos y ciertos elementos deberían estar o no dependiendo de su rol

El ultimo usuario es el que tiene el rol de empleado, este puede manejar y consultar datos de la galería y también del inventario, pero no puede modificarlos. Además, puede tener el papel de cajero en el caso en el que el comprador necesite hacer cualquier pago por compra o subasta