

ANTEPROYECTO

Índice:

I. Planteamiento.....	p.1
II. Diagrama de clases.....	p.2
III. Jerarquías y clases.....	p.3
IV. Interfaz gráfica.....	p.7
V. Futuras ampliaciones.....	p.8

I. Planteamiento

El proyecto que planteo desarrollar consiste en un programa para la gestión de un museo en el que se pueden, además, organizar exposiciones a partir de los fondos del mismo. Se hará uso de un ArrayList para almacenar las distintas obras de arte. A partir de la clase obra de arte se establecerá una jerarquía, de obra de arte heredarán los distintos tipos de obras de arte (pintura, escultura, etc.). Todas las obras de arte estarán identificadas por un código único, que se establecerá de forma automática al crear la obra.

Se implementará la clase Fondos, que será un envoltorio de ArrayList, constituyendo una lista de obras que se podrán añadir, eliminar, etc. No se podrá crear más de una obra con el mismo título, autor y fecha.

Al añadir una obra al museo, ésta siempre se dejará en el almacén, que es una de las constantes de una enumeración de salas, así que al instanciar cualquier obra, el campo sala, se iniciará, por defecto, como almacén. Cuando exponemos una obra puede pertenecer a una de las tres salas de las que dispone el museo, de modo que se puedan englobar en una sala obras que compartan características comunes para que la "visita" a la exposición esté más organizada. Una obra no podrá estar expuesta y en el almacén a la misma vez, el campo sala deberá modificarse cuando se exponga una obra.

Cuando una obra es expuesta varias veces, el daño a la obra aumenta, si el daño hace que el estado de conservación de la obra sea malo, ésta no se podrá exponer. Habrá que restaurarla para que el estado vuelva a ser bueno. Si el estado llega a ser muy malo, no se podrá hacer nada con la obra, ni siquiera restaurar. El estado de conservación será una enumeración.

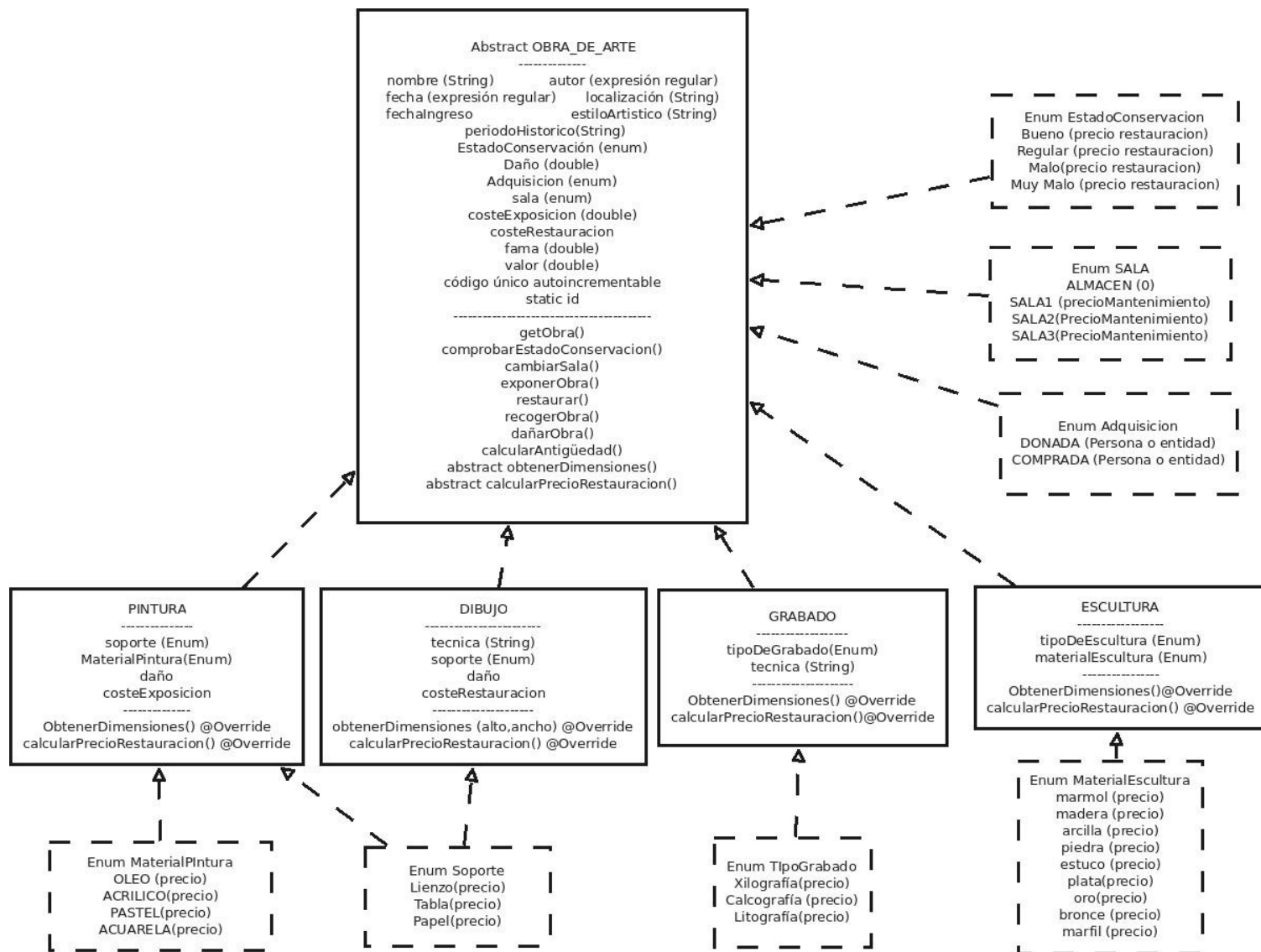
Para gestionar el museo tendremos una clase denominada GestionMuseo compuesta del ArrayList de obras de arte "Fondos", que constituiría la colección de obras de arte que hay en el museo, de un nombre de exposición, de una descripción de la misma y de un presupuesto que se verá aumentado o disminuido mediante los ingresos de las exposiciones y los gastos de las mismas y de las restauraciones.

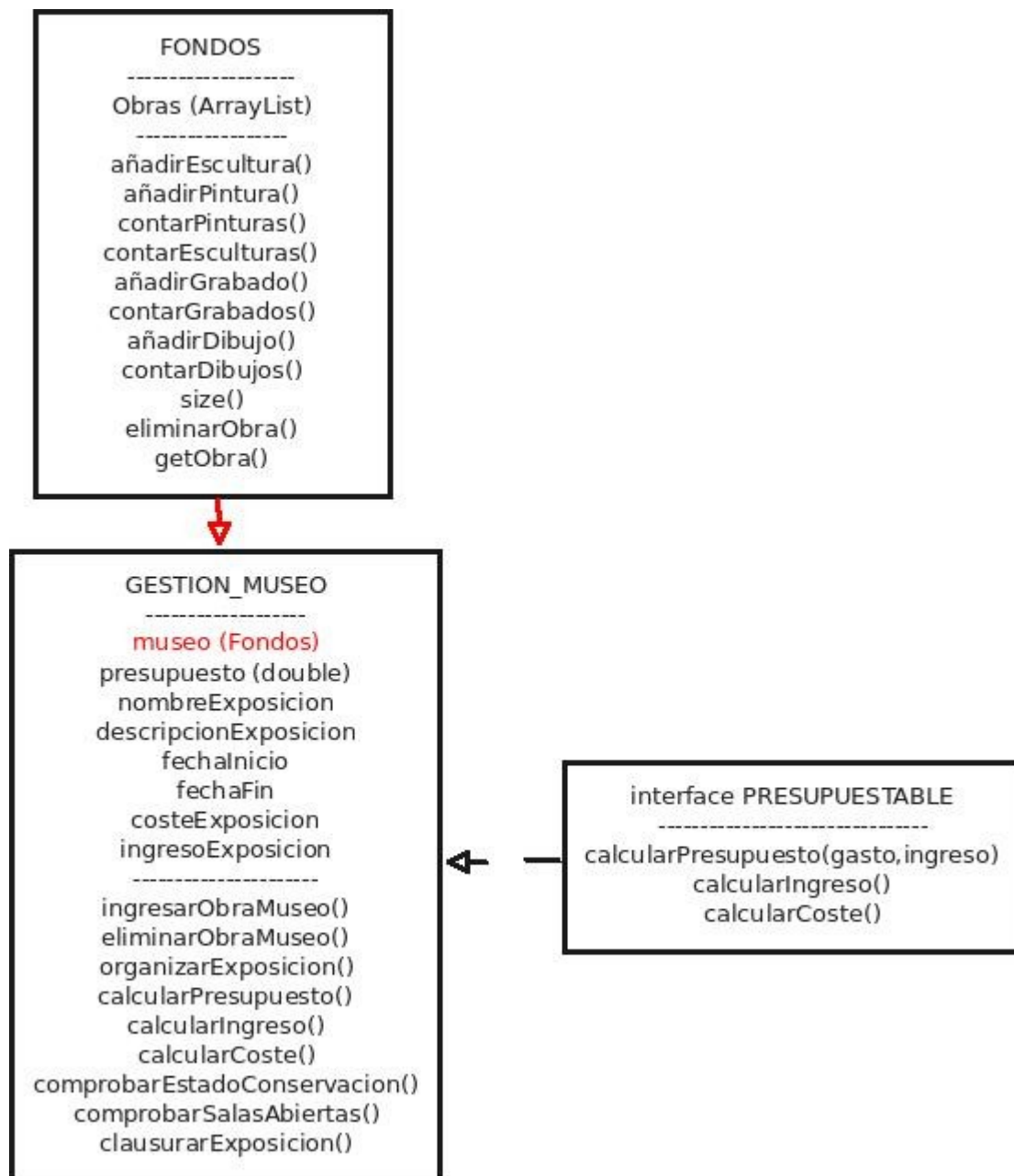
Si una obra de arte se encuentra en cualquier sala que no sea el almacén significará que está en exposición. Al clausurar una exposición, las obras vuelven al almacén y el nombre de exposición y su descripción se ponen a null, en caso de que el nombre de la exposición esté a null, se podrá organizar una nueva exposición. No podrá organizarse otra exposición si ya hay una, ni podrá visitarse una exposición vacía. Sin embargo, podrá consultarse siempre el catálogo de las obras del

museo y por tanto todos los fondos del museo, aunque no formen parte de una exposición.

Se implementarán además dos interfaces: la interfaz presupuestable y la interfaz restaurable. Todos estos aspectos y otros (interfaz gráfica, expresiones regulares, flujo de datos, fechas...) se irán desarrollando y detallando a lo largo de los distintos puntos de los que consta este anteproyecto.

II. Diagrama de clases:





III. Jerarquías y clases:

Abstract **Class ObraDeArte**: implements *Restaurable*, *comparable*, *Serializable*.

Campos:

- Nombre
- autor (expresión regular)
- fecha (expresión regular)
- localización (dónde se encontró)

- fechaIngreso. Para poder calcular el tiempo que lleva la obra en el museo. (Expresión regular).
- EstiloArtistico (expresión regular)
- periodoHistorico (expresión regular)
- código único autoincrementable. (codigo e identificador estático)
- Estado de conservación (enum).
- Daño (contador de daño que hace que la obra tenga uno u otro estado de conservación).
- adquisicion (enum)
- sala (enum)
- costeExposicion (double)
- costeRestauracion (double) suma de todos los pluses de restauración.
- fama (double). Plus que suma al ingreso que se obtenga en la exposición.
- Valor (double). Plus que suma al gasto de la restauración.

Métodos:

- getObra().
- ComprobarEstadoConservación().
 - Daño<=4 Bueno
 - Daño>4&<7 Regular
 - Daño>7 Malo
 - Daño>10 MuyMalo
 Si el estado de conservación llega a ser MuyMalo, ya no se puede restaurar.
- CambiarSala(). Es un setter de sala.
- exponerObra(). Cambiar la sala a sala1, sala2 o sala3.
- RecogerObra(). Cambiar la sala a almacén.
- DañarObra(). Daño++
- restaurar(). Daño=0.
- Abstract calcularPrecioRestauracion().
- CalcularAntiguedad(). Este método calcularía cuánto lleva una obra en el museo.
- Abstract ObtenerDimensiones().

Class Pintura: *extends ObraDeArte:*

Campos:

- soporte (enum)
- MaterialPintura (enum)
- daño= daño+1. Oculto el campo de la clase padre. Las pinturas sufren más de lo normal al exponerlas.
- CosteExposición= costeExposición+5. Cuesta más exponer pinturas.

Métodos:

- ObtenerDimensiones(double alto, double ancho). @Override.
- CalcularPrecioRestauracion(). @Override.

Class Escultura: *extends* ObraDeArte:

Campos:

- tipoDeEscultura (enum)
- MaterialEscultura (enum)

Métodos:

- ObtenerDimensiones (double alto, double ancho, double profundidad). @Override.
- CalcularPrecioRestauracion(). @Override.

Class Grabado: *extends* ObraDeArte:

Campos:

- TipoDeGrabado
- Tecnica

Métodos:

- ObtenerDimensiones(double alto, double ancho). @Override.
- CalcularPrecioRestauracion(). @Override.

Class Dibujo: *extends* ObraDeArte:

Campos:

- Tecnica
- soporte
- daño= daño+0.5. Oculto el campo de la clase padre. El dibujo es más frágil que otras obras.
- costeRestauracion= costeRestauracion-10%. Cuesta menos restaurar un dibujo.

Métodos:

- ObtenerDimensiones(double alto, double ancho). @Override.
- CalcularPrecioRestauracion(). @Override.

Class Fondos

Campos:

- ArrayList de obras de arte.

Métodos:

- AñadirPintura().
- AñadirEscultura().
- AñadirGrabado().
- AñadirDibujo().
- ContarPinturas().
- ContarEsculturas().
- ContarGrabados().

- ContarDibujos().
- Size().
- EliminarObra(). Se eliminará por código.
- GetObra(). Devuelve una obra por código.

Class GestionMuseo: *implements Presupuestable*

Campos:

- museo (Fondos).
- Presupuesto.
- NombreExposicion.
- DescripcionExposicion.
- FechaInicio.
- FechaFin.
- CosteExposicion.
- IngresoExposicion.

Métodos:

- IngresarObraMuseo(). Añadir una obra a los fondos del museo.
- EliminarObraMuseo(). Eliminar una obra de los fondos del museo.
- organizarExposicion().
- calcularPresupuesto(). *Presupuestable*
- ComprobarEstadoConservacion(). Si el estado de conservación de una obra es malo, o muyMalo no se puede exponer. [MalEstadoException]
- comprobarSalasAbiertas(). Devuelve una cantidad que se suma al coste.
- ClausurarExposición(). Recorrer Fondos comprobando si la sala no es almacén y si no lo es... recogerObra().
- CalcularIngreso(). El ingreso se obtendrá multiplicando el ingreso de todas las obras (fama) por los días que dure la misma. Ingreso x días.
- CalcularCoste(). CosteExposicion de todas las obras + precio de las salas abiertas x días.

Interfaz Presupuestable:

- calcularPresupuesto(double coste, double ingreso)
- calcularCoste()
- CalcularIngreso()

Enum EstadoConservación:

- Bueno (precioRestauracion)
- Regular (precioRestauracion)
- Malo (precioRestauracion)
- MuyMalo (precioRestauracion)

Enum Sala:

- almacen (0)
- sala1 (precioMantenimiento)
- sala2 (precioMantenimiento)
- sala3 (precioMantenimiento)

Enum Soporte:

- lienzo (precio)
- tabla (precio)
- papel (precio)

Enum MaterialPintura:

- oleo (precio)
- acrílico (precio)
- pastel (precio)
- acuarela (precio)

Enum MaterialEscultura:

- marmol (precio)
- madera (precio)
- arcilla (precio)
- piedra (precio)
- estuco (precio)
- plata (precio)
- oro (precio)
- bronce (precio)
- marfil (precio)

Eum TipoDeGrabado:

- Xiligrafia (precio)
- Calcografia (precio)
- Litografia (precio)

Enum Adquisicion:

- Donada (personaOEntidad)
- Comprada (personaOEntidad)

Interfaz Gráfica:

Al programa se podrá acceder como visitante o como administrador. Los visitantes sólo podrán utilizar las opciones derivadas de ver el catálogo del museo y de visitar la exposición. Para las demás opciones se pedirá la clave del administrador.

La interfaz gráfica del programa constará de menús desplegables con las siguientes opciones:

1. Gestionar.

1.1. Fondos.

- Añadir obra al catálogo. (Despliega un menú con los tipos de obra que se pueden añadir y al pinchar, aparece un formulario con los campos de la obra a rellenar)
- Eliminar obra del catálogo. (Por código)
- Actualizar datos de una obra. (Por código)
- Consultar datos de una obra. (Por código) Se consultarían todos los datos, incluso cuánto lleva la obra en el museo, si fue donada y por quien o comprada y a quien.

1.3. Restauraciones.

- Mostrar obras a restaurar.
- Restaurar obra.

1.4. Exposición.

- Organizar exposición. Al pinchar en esta opción aparecerían:
 - Datos a rellenar: nombre y descripción.
 - Una pestaña "exponer obras" que despliega un menú con las salas. Al pinchar en una sala se abrirá un checkbox con las obras del museo que estén disponibles para añadir a la exposición con dos botones, uno para añadir y otro para cancelar.
 - Una pestaña "quitar obras" que despliega un checkbox con las obras expuestas con dos botones, uno para aceptar y otro para cancelar.
 - El presupuesto de que se dispone.
 - Un contador con el coste.
 - Un botón para aceptar. Una vez que se acepte, se modifica el presupuesto y aparece una ventanita indicando los ingresos de la exposición y el nuevo presupuesto del que se dispone.
 - Un botón para cancelar.
- Trasladar obra. (Cambiar obra de sala).
- Clausurar Exposición. (Cambiar la sala de las obras a almacén).

2. Visitar.

2.1. Catálogo. En el catálogo sólo aparecerá la información de la obra que se quiera exponer al visitante (por ejemplo, lo que lleve la obra o no en el museo aquí no interesaría).

1.2.1. Todas las obras.

- Por código
- Por autor (Lista autores por orden alfabético y que al pinchar en un autor muestre sus obras).
- Por estilo (Despliega menú con los estilos y que al pinchar en un estilo muestre las obras que tienen ese estilo).

1.2.3. Por tipo de obra. Despliega un menú con los tipos de obras que existen y al pinchar en un tipo aparece el mismo menú anterior:

- Por código
- Por autor (Lista autores por orden alfabético y que al pinchar en un autor muestre sus obras).
- Por estilo (Despliega menú con la enumeración de estilos y que al pinchar en un estilo muestre las obras que tienen ese estilo).

2.2. Exposición: Al pinchar aparecería el nombre y la descripción de la exposición y un menú con salas que tengan obras y dentro de cada sala:

- Ver por orden de exposición.
- Ver por fecha.
- Ver por autor.

V. Futuras ampliaciones:

- Al crear una obra para añadirla al ArrayList se podrá añadir una imagen.
- Una obra se podrá prestar a otros museos para lo que se creará la interfaz prestable con los métodos prestarObra() y recogerObraPrestada().

La clase obra requerá un campo prestada (booleano) e implementá la interfaz prestable. Los métodos de la interfaz se resolverán en la clase obra de la siguiente forma:

-prestarObra(). Cambio de prestada (class obra) a true. *Prestable*

Si está prestada o expuesta (class obra) no puedo prestarla.

[ObraPrestadaException], [ObraExpuestaException]

Si su estado de conservación (class obra) es muyMalo, malo o regular no puedo prestarla. [MalEstadoException]

-recogerObraPrestada(). Lo contrario de prestarObra(). *Prestable*

Si el estado de una obra de arte es regular o malo no se podrá prestar.

En el menú, en gestionar se añadiría otra opción:

1.2. Préstamos.

- Prestar obra.
- Recoger obra prestada.
- Se podrán solicitar obras a otros museos. La clase GestionMuseo dispondrá del método: SolicitarObra(). Según la fama de la exposición (suma de la fama de las obras que se van añadiendo a la exposición) se podrán solicitar ciertas obras de otros museos, una lista de obras ya previamente creadas por museo. (Si la fama de toda la exposición es igual o mayor a la fama de esa obra).

En el menú de Gestionar-exposición-organizar exposición se añadiría un botón "solicitar obras" que despliega un menú con los museos a los que puedo solicitar obras según la fama de la exposición. Al pinchar en un museo, me carga el ArrayList de las obras de ese museo.