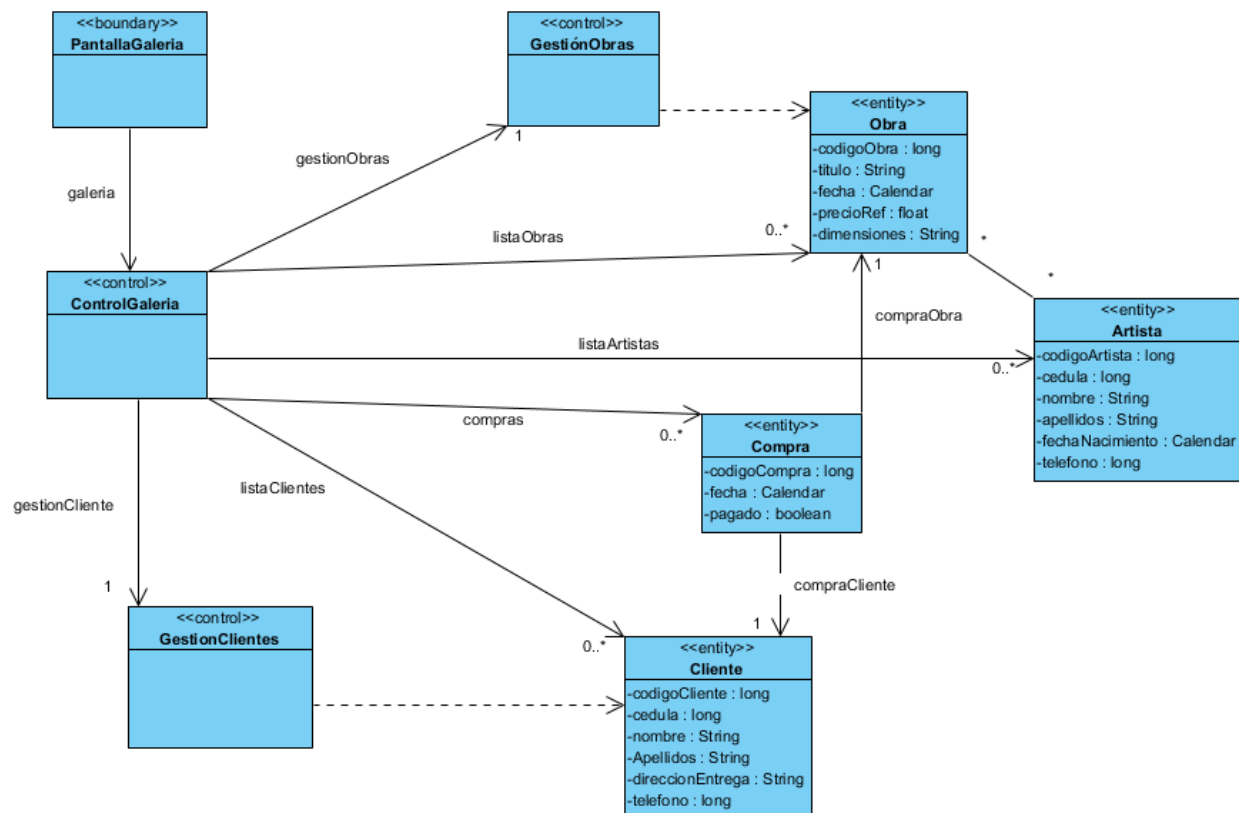




Sistema de Control de una Galería de Arte

- Se requiere hacer un programa orientado a objetos que funcionará en una galería de arte.
- La galería tiene un catálogo de obras artísticas disponibles para la venta. Cuando se compra una obra, esta se le reserva al cliente y se paga en la galería y hay que registrarlo en el sistema.
- El sistema controla la información de las obras disponibles, la información de los clientes que realizan una compra, y la información de la compra en si. El pago se realiza físicamente en la galería pero en el sistema debe quedar registrado.
- En el diagrama de clases UML que se muestra a continuación se observan las clases que intervienen en el problema:



Para esta entrega se deben implementar las siguientes funcionalidades, para ello se deben implementar métodos en las diferentes clases para lograrlo. Usted puede seguir estas funcionalidades para hacer el menú de la aplicación:

1. Ver listado de Obras disponibles
2. Buscar una Obra por título, artista o año
3. Insertar una Obra
4. Modificar una Obra
5. Eliminar una Obra

6. Ver listado de Clientes registrados en el sistema
7. Buscar un Cliente
8. Insertar Cliente
9. Modificar datos de Cliente
10. Eliminar un Cliente
11. Realizar compra de una Obra
12. Eliminar compra de obra
13. Ver listado de Compras existentes
14. Ver listado de Compras para un mes y año específico insertado por el usuario
15. Ver listado de Artistas más vendidos
16. Salir

Para este proyecto se solicita implementar las siguientes funcionalidades:

1. **[5]** Al iniciar el día se debe:
 - a. Crear la colección de Obras llamada listaObras (método en el controlador 'GestionObras' que crea instancias de Obra y llena la lista de obras de la clase ControlGaleria. Esta clase sólo existe para no tener que ingresar datos por pantalla de las obras y facilitar la prueba del programa. No obstante, las funcionalidades para insertar, modificar y eliminar una obra deben programarse y funcionar correctamente.
 - b. La clase GestiónObras también puede llenar la lista de Artistas.
 - c. Crear la colección de clientes llamada listaClientes (método en el controlador 'GestionCliente' que crea instancias de Cliente y llena la lista de clientes de la clase ControlGaleria. Esta clase sólo existe para no tener que ingresar datos por pantalla de clientes y facilitar la prueba del programa. No obstante, las funcionalidades para insertar, modificar y eliminar un cliente deben programarse.
 - d. La clase ControlGaleria debe invocar en su constructor los métodos anteriores.

NOTA: Para los métodos anteriores los objetos se pueden crear con valores literales; en la siguiente entrega esto se cambiará por un acceso a archivos del sistema.

2. **[5]** Ver listado de Obras disponibles
 - a. Se debe mostrar un listado por pantalla que muestra la información básica de una obra que son título, fecha de creación, precio de referencia, una foto (la foto será para la entrega 3) y sus dimensiones. Solo deben salir las Obras que están disponibles para la Compra.
3. **[5]** Buscar una Obra por título, artista o año
 - a. Se debe pedir el criterio por el cual se desea Buscar, insertar el criterio de búsqueda y mostrar las Obras que cumplan con el criterio de búsqueda.
4. **[5]** Insertar una Obra:
 - a. Se debe permitir insertar una nueva obra en la lista de obras de la clase ControlGaleria. Para ello debe apoyarse en los métodos necesarios de la clase Obra.
 - b. Debe listar los artistas existentes (Cedula y NombreyApellidos) y si no está debe permitir insertar un nuevo artista. De un artista se guarda su cédula, nombre y apellidos, fecha de nacimiento y teléfono.
 - c. No se puede crear una obra con el mismo código de una que ya exista
 - d. El código de una obra tiene 7 números. Debe validar que nunca exista una obra con más o menos números.
5. **[5]** Modificar una Obra:
 - a. Se debe solicitar el código de la obra a modificar, si este no existe, se debe mostrar un mensaje y volver al menú principal.
 - b. Si existe, se deben mostrar los datos de la obra, enumerados, y solicitar el dato que se quiere modificar.
 - c. No se puede asignar un código de una obra que ya exista, si esto ocurre se muestra un mensaje y no se modifica la obra.
6. **[5]** Eliminar una Obra:

- a. Solicitar el código de la obra a eliminar, si este no existe, se debe mostrar un mensaje y volver al menú principal.
- b. No se puede permitir eliminar una obra que esté asociado a una compra.
- c. Se debe mostrar un mensaje de confirmación para eliminar la obra.
7. **[5]** Ver listado de Clientes registrados en el sistema:
 - a. Se debe mostrar un listado por pantalla que muestra la información de un cliente. Su número de identificación, nombre completo, cantidad de personas que lo acompañan y teléfono.
8. **[5]** Insertar Cliente:
 - a. Se debe permitir insertar un nuevo cliente en la lista de clientes de la clase ControlGaleria. Para ello debe apoyarse en los métodos necesarios de la clase Cliente.
 - b. No se puede crear un cliente con el mismo código de identificación de uno que ya exista.
9. **[10]** Modificar datos de Cliente:
 - a. Se debe solicitar el código de identificación del cliente a modificar, si este no existe, se debe mostrar un mensaje y volver al menú principal.
 - d. Si existe, se deben mostrar los datos del cliente, enumerados, y solicitar el dato que se quiere modificar.
 - a. No se puede asignar un número de identificación de un cliente que ya exista, si esto ocurre se muestra un mensaje y no se modifica el cliente.
10. **[5]** Eliminar un cliente:
 - a. Solicitar el número de identificación del cliente a eliminar, si este no existe, se debe mostrar un mensaje y volver al menú principal.
 - d. No se puede permitir eliminar un cliente que esté asociado a una compra.
 - a. Se debe mostrar un mensaje de confirmación para eliminar el cliente.
11. **[20]** Realizar compra de una Obra
 - a. Para comprar una obra ya deben existir la obra y el cliente, estos datos se van a solicitar usando los códigos de ambos. Se debe autogenerar un código de compra que no exista previamente.
 - b. Se debe solicitar toda la información de la compra y validar que no exista una compra de ese cliente para esa obra. Si es así mostrar mensaje de error.
12. **[5]** Eliminar compra de obra
 - a. Solicitar el número de compra a eliminar, si este no existe, se debe mostrar un mensaje y volver al menú principal.
 - b. Se debe mostrar un mensaje de confirmación para eliminar la compra.
13. **[5]** Ver listado de Compras existentes:
 - a. Se debe mostrar una lista de todas las Obras que han sido compradas, cliente que la compró, fecha y precio.
14. **[5]** Ver listado de Compras para un mes y año específico insertado por el usuario
 - a. Se debe solicitar mes y año al usuario y mostrar listado de Obras que hayan sido compradas, cliente que la compró, fecha y precio.
15. **[5]** Ver listado de Artistas más vendidos
 - a. Mostrar los artistas más vendidos ordenados de mayor a menor ventas
16. Se debe crear en el main de la clase 'PantallaGaleria' métodos que permitan invocar e imprimir el resultado (valor de retorno) de cada una de las funcionalidades del controlador de 'ControlGaleria', usted debe:
 - a. proveer de manera literal los valores de prueba
 - b. solicitar por pantalla los valores:
 - i. aquí se requiere de un menú interactivo para cada opción.

NOTA: debe probar todos los caminos de negocio.

1. Fechas

• Entrega

La entrega se hace a través del buzón de transferencia digital del sistema de aprendizaje UVirtual (BlackBoard) a más tardar el día **Domingo 27 de Septiembre de 2020 hasta las 23h59 p.m.**

Entregas posteriores a esta fecha **NO** serán tenidas en cuenta.

Si realiza varias entregas en el sistema, sólo se tendrá en cuenta la última entrega.

2. Grupos

La entrega se realizará en grupos de trabajo de 5 personas. Los grupos no podrán cambiar su conformación y desde el comienzo dichos grupos estarán identificados plenamente.

3. Entregables

- Se entrega un archivo zip con el proyecto Maven (archivo pom.xml, carpeta src y archivo readme.txt).
- El nombre del archivo debe ser "nombreGrupo.zip"
- Un envío por grupo. El archivo zip debe contener un "readme.txt" con los integrantes del grupo.

4. Observaciones

- El diagrama de clases y la implementación deben ser concordantes.
- Si no hay código, la nota corresponderá a 0.0
- El proyecto debe funcionar con la versión 11 de Java

5. Restricciones

- La lógica y la presentación deben estar separadas.
 - Se deben leer datos en la presentación y procesarlos en la lógica de negocio
 - Toda la creación y procesamiento de objetos debe realizarse en la lógica pasando los parámetros necesarios desde el GUI (main de la pantallaGaleria)
- Para las colecciones no use arreglos []
- Sólo se calificarán proyectos que tengan la presentación y la lógica completas