Programación 4

EXAMEN FEBRERO 2022

Problema 1 (30 puntos)

Se desea modelar un sistema de gestión de un *parking* (estacionamiento de autos). Dentro del parking pueden estacionar usuarios mensuales o visitantes, todos se identifican por su documento de identidad y se conoce su teléfono. De los usuarios mensuales además se conoce su tarifa mensual fija y qué vehículos posee, registrando marca, modelo y matrícula para identificarlos. A los visitantes se les solicita el email. El parking cuenta con varios pisos que se identifican por un número y poseen una tarifa específica por hora. Cada piso tiene múltiples lugares para estacionar que se identifican con un código. Cada vez que un usuario estaciona se crea un ticket registrando las horas de entrada y de salida, usuario y lugar utilizado. En caso de ser visitante, además se registra el costo total calculado en horas y en caso de ser mensual se registra el vehículo.

Además, se cuenta con la descripción del siguiente Caso de Uso:

Caso de Uso	Alta de Ticket
Actor	Usuario
Descripción	El caso de uso comienza cuando el usuario quiere estacionar, para ello indica su documento. Si el usuario es mensual debe ingresar su vehículo, eligiendo uno de sus vehículos registrados o agregar uno nuevo para el cual debe especificar todos sus datos. El usuario indica su hora de entrada y salida deseada con el cual el sistema realiza el cálculo del costo dependiendo del tipo de usuario. En caso de ser visitante el usuario debe indicar si necesita factura enviada a su email o no. Finalmente, el usuario solicita generar el ticket con el cual podrá ingresar, ocupando el espacio asignando por el sistema.

Se pide:

a) Realizar el Modelo de Dominio de la realidad planteada, incluyendo restricciones en lenguaje natural.
b) Realizar el Diagrama de Secuencia del Sistema para el Caso de Uso “Alta de Ticket”, indicando el uso de memoria del Sistema así como de Datatypes.

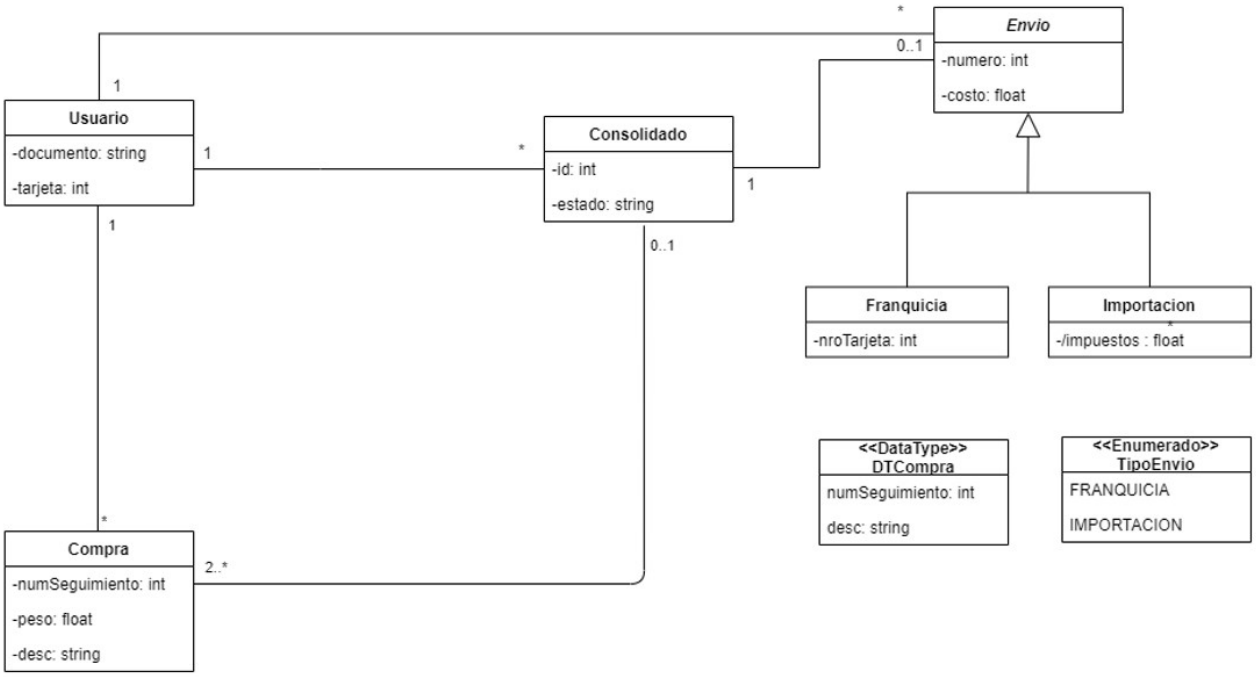
Programación 4

EXAMEN FEBRERO 2022

Problema 2 (35 puntos)

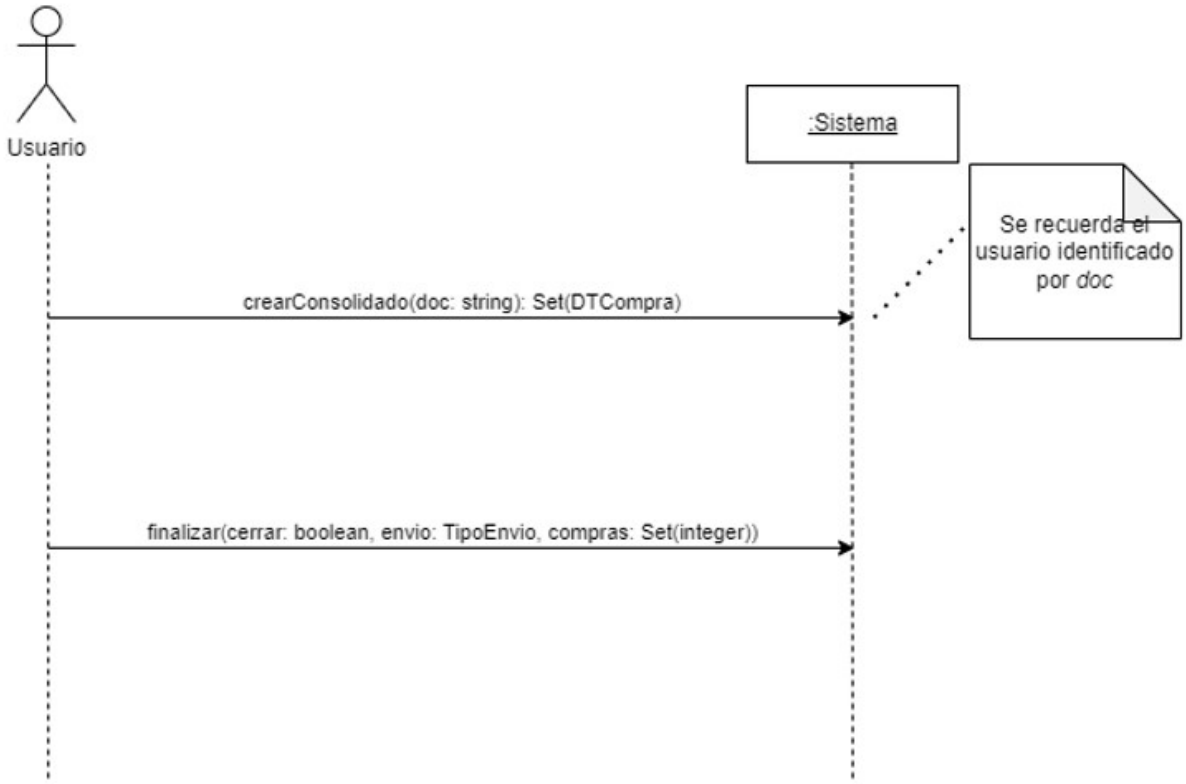
Un *courier* permite a los usuarios registrados hacer compras en Estados Unidos y traerlas a Uruguay. El courier recibe las compras, identificadas por número de seguimiento, en sus oficinas de Miami y las registra en el sistema. Para traer varias compras desde Estados Unidos en un único envío, el sistema permite al usuario crear un consolidado y agregarle los números de seguimiento que desee. Una vez creado el consolidado, el usuario puede elegir cerrarlo para que se genere el envío a Uruguay o dejarlo abierto, lo que le permitirá editarlo a futuro.

Considere el siguiente modelo de dominio para dicho sistema.



Por otra parte, la siguiente figura presenta el Diagrama de Secuencia del Sistema (DSS) para el caso de uso *Crear Consolidado*.

El caso de uso comienza cuando el usuario desea crear un consolidado. Para esto el sistema lista todas las compras que tiene registradas a su nombre en el sistema y que no están asociadas a ningún consolidado. Para cada compra se lista su descripción y número de seguimiento. El usuario selecciona todos los números de seguimiento que quiere incluir en el consolidado. Además, indica si desea cerrar el consolidado o dejarlo abierto y si desea que sea enviado como franquicia o como importación. Si desea cerrar el consolidado, el sistema crea el envío incluyendo el consolidado y calcula su costo. El costo se calcula como el peso del consolidado por el precio por kilo almacenado en el sistema. De lo contrario el consolidado queda abierto y no se crea el envío. Si el usuario indica que sea enviado como franquicia, el envío debe incluir el número de tarjeta del usuario. En caso de indicar que sea enviado como importación se debe calcular el valor de los impuestos, que corresponden al 60% del costo del envío.



Además, se cuenta con los contratos de las dos operaciones del sistema identificadas en el DSS.

`crearConsolidado(doc: string): Set(DTCompra)`	
Descripción:	Recuerda al usuario y retorna los números de seguimiento y las descripciones de las compras asociadas al usuario, que no están incluidas en ningún consolidado.
Parámetros:	• `doc`: Identificador del usuario.
Precondiciones:	• Existe una instancia de `Usuario` en el sistema cuyo documento es igual a `doc`.
Poscondiciones:	• Se recuerda al `Usuario` identificado por `doc`. • Se crea una colección de `DTCompra` que incluye la descripción y el número de seguimiento de cada instancia de `Compra` para la que existe un link con la instancia de `Usuario` identificada por `doc` y que no tiene un link con una instancia de `Consolidado`.

`finalizar(cerrar: boolean, envio: TipoEnvio, compras: Set(integer))`	
Descripción:	Se crea el consolidado y si corresponde se crea el envío.
Parámetros:	• `cerrar`: Indica si se debe cerrar o dejar abierto el `Consolidado`. • `envio`: Indica si el `Envio` será `Franquicia` o `Importacion`. • `compras`: Indica la lista de números de seguimiento de las instancias de `Compra` que formarán parte del `Consolidado`.
Precondiciones:	• Existen instancias de `Compra` identificadas por los números de seguimiento indicados en la colección `compras`. • Existe una instancia de `Usuario` recordada.
Poscondiciones:	• Se crea una instancia de `Consolidado` con un identificador generado por el sistema. • Se crea un link entre la instancia de `Consolidado` y cada una de las instancias de `Compra` identificadas por los números incluidos en `compras`. • Si `cerrar = false` la instancia de `Consolidado` queda con estado `abierto`. No se crea instancia de `Envio`. • Si `cerrar = true` la instancia de `Consolidado` queda con estado `cerrado`, se crea una instancia de `Envio` y se crea un link entre esta y la instancia de `Consolidado`. Se calcula el costo del envío y se guarda en la instancia de `Envio` creada. • Si `envio = FRANQUICIA` se guarda en la instancia de `Envio` el número de tarjeta del `Usuario` recordado. • Si `envio = IMPORTACION` se calculan los impuestos y se guardan en la instancia de `Envio`. • Se crea un link entre la instancia de `Envio` y la instancia de `Usuario`.

Se pide:

a) Realizar los Diagramas de Comunicación correspondientes a las operaciones `crearConsolidado` y `finalizar`, incluyendo visibilidades.
b) Realizar el Diagrama de Clases de Diseño (DCD) resultante.

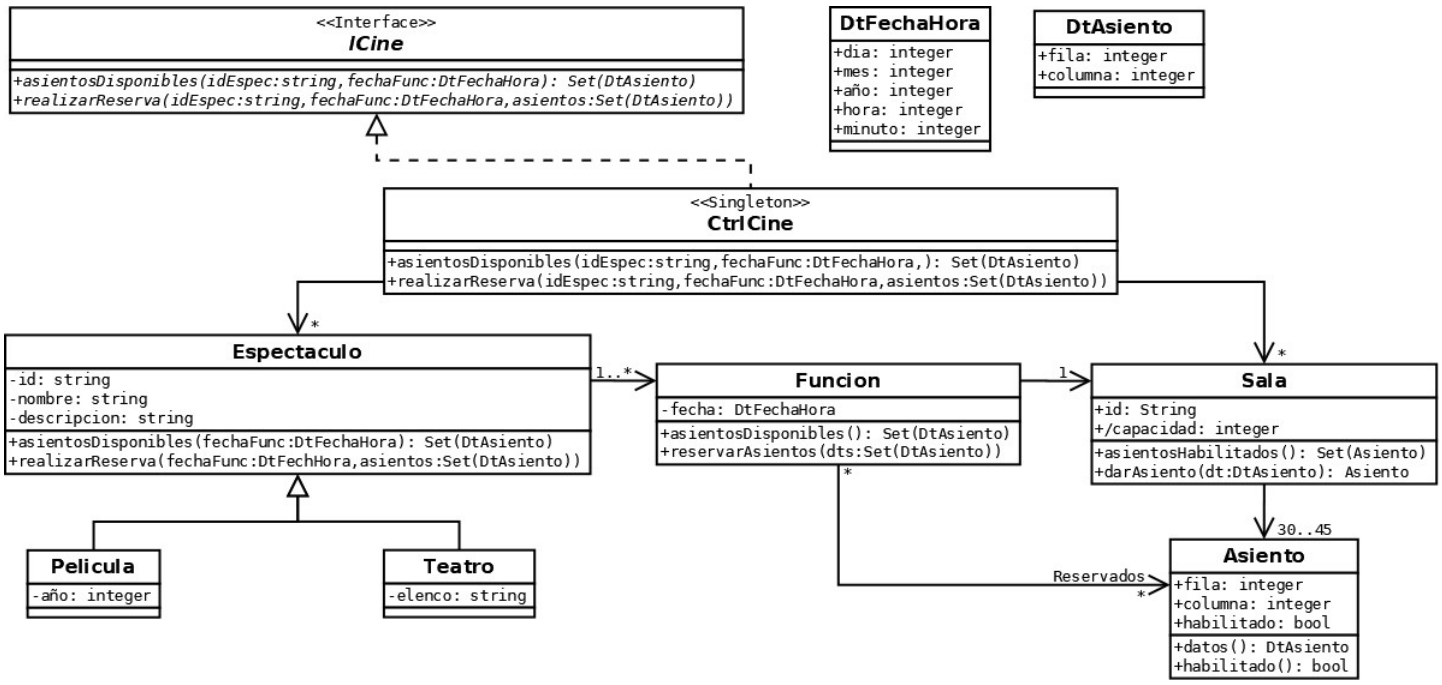
Programación 4

EXAMEN FEBRERO 2022

Problema 3 (35 puntos)

Parte A:

La figura muestra el Diagrama de Clases de Diseño (DCD) parcial de un sistema para la reserva de entradas a espectáculos ofrecidos por un cine. Se ofrecen tanto películas como obras de teatro, y para ambos casos hay varias funciones en diferentes días y horarios. Las funciones de una película se identifican por la fecha y hora. Una sala de cine solo trasmite una función a la vez y sus asientos se identifican por número de fila y columna.



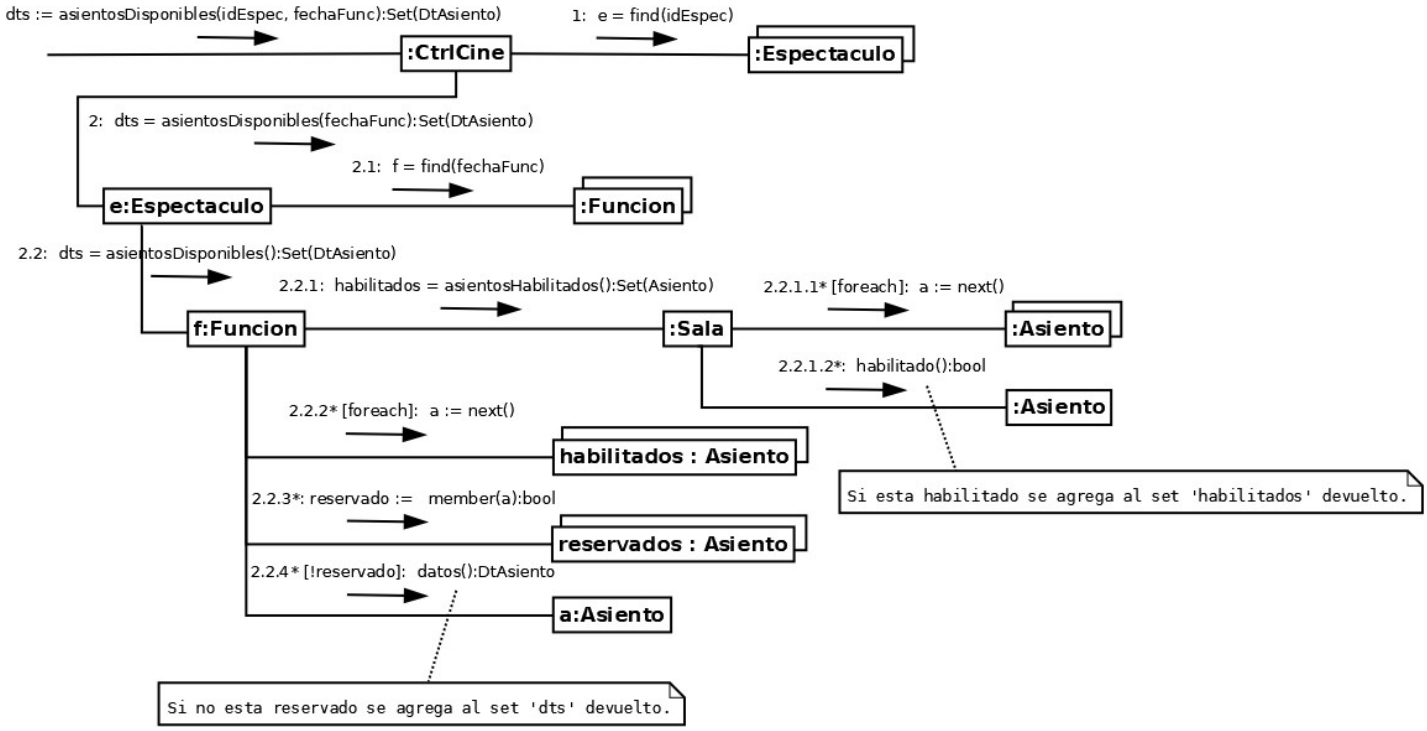
Se pide:

Implementar en C++ los .h de las clases `ICine`, `CtrlCine` y `Teatro`.

(sigue en siguiente página)

Parte B:

A continuación, se muestra el diseño de la operación `asientosDisponibles()` del controlador, la cual permite obtener los asientos disponibles para un espectáculo y función dados. Los asientos disponibles son aquellos que se encuentran habilitados y aún no han sido reservados.



Se pide:

Implementar en C++ las operaciones que aparecen en el diagrama de comunicación.

Observaciones:

- Puede utilizar colecciones genéricas (realizaciones de IDictionary e ICollection) o paramétricas (contenedores STL).
- No incluir directivas al precompilador (#include, etc).