

# Programación 4

## EXAMEN DICIEMBRE 2019

Por favor siga las siguientes indicaciones:

- Escriba con lápiz y de un solo lado de las hojas.
- Escriba su nombre y número de documento en todas las hojas que entregue.
- Numere las hojas e indique el total de hojas en la primera de ellas.
- Recuerde entregar su número de examen junto al examen.
- Está prohibido el uso de computadoras, tabletas o teléfonos durante el parcial.

### **Problema 1 (30 puntos)**

Cada año son más los Uruguayos que por estas fechas, aprovechan las temporadas de descuentos que tienen las principales plataformas de comercio electrónico (e-Commerce) en el mundo, como Amazon, Ali Express, Alibaba, entre otros. Como no todas estas plataformas tienen envío directo a Uruguay, sumado a la complejidad de los trámites de importación, muchos optan por delegar este trabajo a una empresa especializada en envío de cargas, comúnmente denominadas “couriers”. Como este mercado ha crecido año a año, reportando cuantiosas ganancias anuales, un grupo de inversores le ha encomendado a usted la tarea de hacer el análisis de una aplicación para la gestión de envíos de compras internacionales desde USA a Uruguay, llamada **FingeaMiCompra**.

De cada usuario de FingeaMiCompra interesa conocer su email que lo identifica, nombre, fecha de nacimiento y un teléfono de contacto. Además, un usuario puede tener múltiples direcciones (la dirección de su casa, trabajo, etc). De cada dirección interesa conocer departamento, ciudad, la calle, número de puerta, código postal, en caso de ser un apartamento el bloque y la unidad y por último un nombre para identificarla por el usuario (“casa”, “trabajo”, etc). Todos los usuarios cuentan con la misma dirección de correo en Miami/USA a la cual deben enviar las compras que hacen en USA, por ello **NO** interesa modelarla. Una vez que realizan una compra mediante cualquier plataforma de e-Commerce, deben anunciar la misma mediante la aplicación. Al anunciar la compra, interesa registrar la empresa de mensajería que entrega dentro de USA la compra en cuestión, junto con el número de seguimiento de la misma (tracking-id), el cual es generado por la empresa de mensajería. Las empresas de mensajería pueden ser: UPS, USPS, FedEx o DHL. A su vez, para cada compra el usuario debe cargar la factura de la misma (archivo pdf con el detalle de la compra), opcionalmente la dirección a la cual quiere que se le envíe en Uruguay (debe ser alguna de las direcciones del usuario), y por último por cual de los siguientes regímenes esta amparada la compra: Franquicia, Libros y revistas, CDs & DVDs. En caso que no se indique dirección de envío en Uruguay, se asume que el paquete se retirará en las oficinas de FingeaMiCompra. Para evitar que cada compra sea re-entregada a Uruguay directamente y con la finalidad de abaratar costos de envío, FingeaMiCompra tiene la funcionalidad de consolidación de envíos. Esto sirve para juntar varias compras anunciadas a través de la plataforma en un solo envío directo a Uruguay. Igualmente, es posible consolidar un envío con una sola compra. Por último, para mejorar la experiencia del usuario, interesa registrar todos los estados intermedios por los que pasa una compra, desde que el usuario la anuncia en FingeaMiCompra, hasta que es entregada en la dirección de recepción final en Uruguay. Para ello se ha relevado la siguiente tabla de eventos y estados intermedios. Para cada compra interesa conocer su estado actual y los estados anteriores, guardando para cada estado además, la fecha en la que se produce la transición al mismo (para el primer estado, la fecha se corresponde con la fecha en que el usuario anuncia la compra), según la Tabla 1. Adicionalmente, cuando la compra es recibida en la dirección en USA, además de pasarla al estado RecibidoUSA, se le saca una foto para que el usuario pueda ver cómo llegó su compra y se registra su peso.

Estado	Evento
Anunciado	Usuario declara una nueva compra a través de FingeMiCompra
RecibidoUSA	FingeMiCompra recibe la compra en dirección de correo USA
ProntoParaEnvíoURU	El usuario indica que la compra está lista y consolidada para ser enviada a Uruguay
EnTránsito USA-URU	FingeMiCompra envía la compra a Uruguay
RecibidoURU	FingeMiCompra recibe la compra en Uruguay
Entregado	La compra ha llegado a su destino

Tabla 1

Por otra parte, se relevaron además los siguientes caso de uso:

Caso de Uso:	Anunciar Compra
Actor:	Usuario
Descripción:	<p>El caso de uso comienza cuando un usuario desea anunciar una nueva compra en el exterior. Para eso el usuario ingresa la plataforma a través de la cual realizó la compra, el servicio de mensajería que utiliza dentro de USA y el identificador del envío generado por la empresa de mensajería (tracking-id).</p> <p>Luego el sistema solicita al usuario que se adjunte la boleta de compra del paquete y defina el régimen por el cual quiere amparar la compra. Finalmente el sistema registra la compra y le asigna el estado “Anunciado” con la fecha actual.</p>

Caso de Uso:	Consolidar Envío
Actor:	Usuario
Descripción:	<p>El caso de uso comienza cuando un usuario desea consolidar múltiples compras recibidas en su casilla de correo en USA, en un solo envío a Uruguay. Para ello el sistema lista las compras que el usuario ha anunciado y se encuentran en estado “RecibidoUSA”. Luego el usuario selecciona las compras que desea consolidar en un solo envío.</p> <p>Luego el sistema lista las direcciones del usuario y si este lo desea, indica la dirección de recepción en Uruguay. En caso de no indicar una dirección de recepción, el envío quedará en las oficinas de EnvíaMiCompra para retiro del usuario.</p> <p>Finalmente el usuario confirma el consolidado y el sistema registra el envío y cambia el estado de todas las compras en el envío a “ProntoParaEnvíoUru”.</p>

**Se pide a partir de la descripción de la realidad y los casos de uso:**

- Modelo de Dominio de la realidad anterior con restricciones en lenguaje natural. Se deben especificar los tipos de los atributos, así como también los datatypes y enumerados utilizados.
- Diagrama de Secuencia del Sistema (DSS) de los Casos de Uso, incluyendo el uso de datatypes y de manejo de memoria del Sistema, en caso de ser necesario.

**Problema 2 (35 puntos)**

La Figura 1 muestra un Modelo de Dominio para un sistema de búsqueda de pasajes aéreos en internet. Una búsqueda puede ser de dos tipos: (i) viaje ida y vuelta, que involucra dos ciudades, un tramo de ida y otro de vuelta entre ellas, (ii) viaje multitramo, que involucra una cantidad arbitraria de tramos, donde cada uno tiene una ciudad de origen y otra de destino. Cada tramo en el sistema tiene su identificador, la fecha del viaje y está asociado a una única búsqueda; pueden existir tramos con identificadores distintos y tener exactamente las mismas ciudades de origen y destino. Una vez que se ejecuta la búsqueda el sistema le asocia un resultado, el cual contiene para cada tramo un conjunto de etapas (cuando no es posible encontrar vuelos directos para un tramo, el resultado está conformado por varias etapas que pasan por ciudades intermedias). Cada etapa tiene una ciudad de origen y una de destino y además tiene el número de vuelo y la aerolínea (ver ejemplos en Figura 2). Todas las búsquedas, resultados e información asociada, se mantienen almacenadas en el sistema con el objetivo (entre otros) de obtener estadísticas.

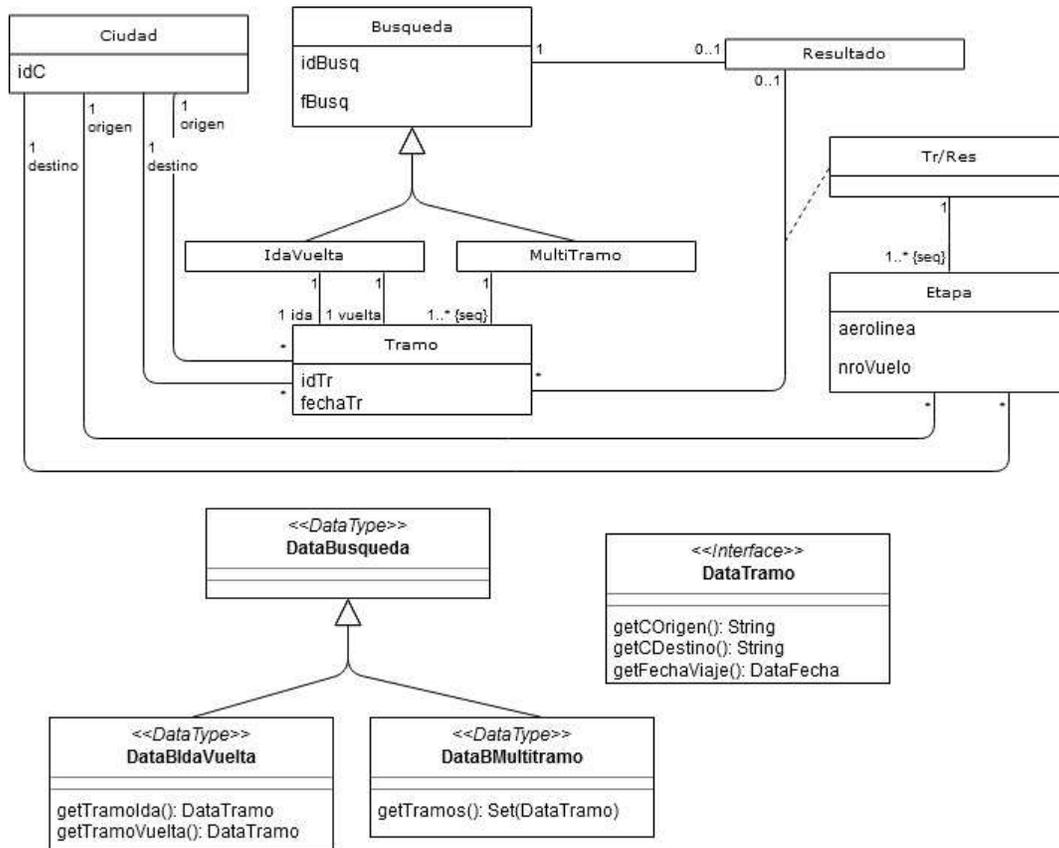


Figura 1

<i>Búsqueda de viaje ida y vuelta</i>			<i>Búsqueda de viaje multitramo</i>		
Ida	Etapa 1	Montevideo	Tramo 1	Etapa 1	Montevideo
		Melo			Mercedes
Vuelta	Etapa 1	Melo	Tramo 2	Etapa 1	Mercedes
		Minas			Artigas
	Etapa 2	Minas	Tramo 3	Etapa 1	Artigas
		Montevideo			Durazno
			Etapa 2	Durazno	
				Montevideo	

Figura 2

Además, se consideran las siguientes operaciones del sistema:

nuevaBusqueda (dtb: DataBusqueda)	
Descripción:	Ingresar una nueva búsqueda para un viaje de tipo ida y vuelta o multitramo
Parámetros:	dtb: Contiene los datos de la búsqueda, dependiendo del tipo
Precondiciones:	Para cada tramo de la búsqueda: - sus ciudades origen y destino son diferentes - existen instancias de Ciudad identificadas por los datos de las ciudades origen y destino del tramo La fecha de viaje de un tramo siempre es mayor a la de su tramo anterior en la secuencia
Postcondiciones:	Se crea una instancia de IdaVuelta o MultiTramo, dependiendo del tipo de búsqueda, con un id generado por el sistema y la fecha actual Para cada tramo de la nueva Busqueda se crea una instancia de Tramo con id generado por el sistema y la fecha indicada por el dato y se asocia a las instancias de Ciudad origen y destino correspondientes
Observaciones:	Las instancias de Resultado, Tr/Res y Etapa son creadas y asociadas a su correspondiente Busqueda en otra operación

vuelosAerolinea (fIni:DTFecha; fFin:DTFecha; aLinea:String):Integer	
Descripción:	Devuelve la cantidad de búsquedas realizadas entre dos fechas, en cuyos resultados haya al menos una etapa realizada por un vuelo de la aerolínea indicada
Parámetros:	fIni, fFin: Rango de fechas para conteo de resultados aLinea: Nombre de la aerolínea buscada
Precondiciones:	fFin es posterior o igual a fIni
Postcondiciones:	El resultado es la cantidad de instancias de Busqueda cuya fecha está entre fIni y fFin y en cuyo Resultado (si tiene) contenga al menos una instancia asociada de Etapa cuyo atributo areolinea coincide con aLinea

### Se pide:

- Realizar los Diagramas de Comunicación correspondientes a las dos operaciones del sistema especificadas, incluyendo visibilidades.
- Realizar el Diagrama de Clases de Diseño resultante.

### **Problema 3 (35 puntos)**

Se desea construir un sistema que maneje el lanzamiento de misiles (intercontinentales, atómicos) tanto para la defensa de otros países como de amenazas del espacio (asteroides, etc). Con dicho propósito fue diseñado el Diagrama de Clases de Diseño de la Figura 3 (los signos de + y - en las clases representan si la operación es pública o privada respectivamente). El comportamiento de las operaciones de cada clase se describe a continuación:

ControladorAereo:

- agregarCohete(coh: Cohete): agrega un nuevo cohete a la colección
- lanzarCohete(id: string): lanza el cohete con id string en caso de existir. Si no existe un cohete con dicho id no realiza ninguna acción.
- destruirCohete(id: string): remueve el cohete con id string de la colección y lo borra en caso de existir. Si no existe cohete con dicho id no realiza ninguna acción.

Cohete:

- encenderMotor(): asuma que posee una implementación de la operación para dicha clase (debe incluir la operación en la especificación de la misma, no debiendo incluir código en la parte de su implementación).

Atomico:

- lanzar(lat: float, lon: float): enciende el motor y emite la alerta de emergencia

- alertaEmergencia(lat: float, lon: float): notifica a todos los observadores sobre la alerta astronómica.

Gobierno:

- notificar(lat: float, lon: float): invoca la alarma astronómica para comunicar a la población.
- alarmaAstronomica(lat: float, lon: float): asuma que posee una implementación de la operación para dicha clase (debe incluir la operación en la especificación de la misma, no debiendo incluir código en la parte de su implementación).

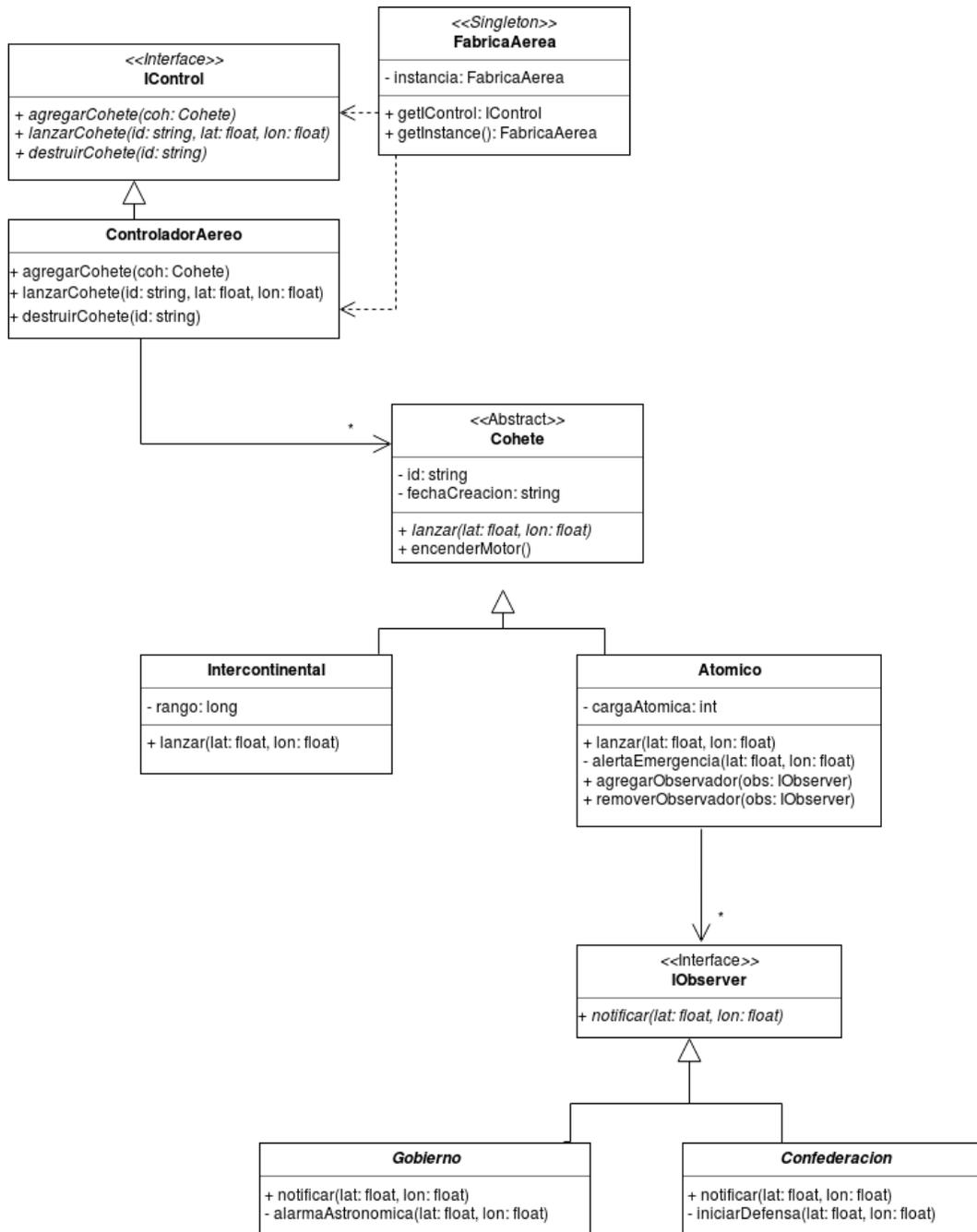


Figura 3

Se pide implementar los siguientes archivos:

<b>.cpp</b>	<b>.h</b>
	IObserver
FabricaAerea	FabricaAerea
Cohete	Cohete
Atomico	Atomico
ControladorAereo	ControladorAereo
Gobierno	Gobierno

**Considerar:**

- Es posible utilizar las clases `set<T>`, `map<K,V>` y `vector<T>` de STL.
- Puede suponer la existencia de la interface `ICollectionable` e implementaciones de `ICollection` (clase `List`) e `IIterator` según sea necesario.
- Las implementaciones **deben** incluir constructores y destructores, liberando en estos últimos toda la memoria que sea necesaria.
- Asumir existencia y no implementar los `get` y `set` de los atributos.
- **No** incluir directivas al precompilador.
- **No** implementar los `datatypes`.