

Trabajo Práctico Integrador Nro 1.

Unidad 4: Paradigma orientado a objetos.

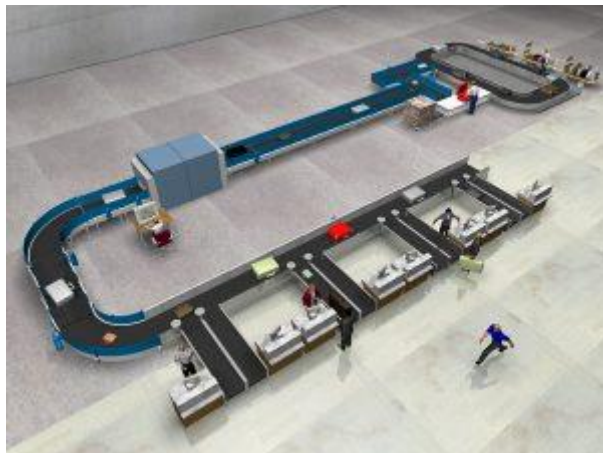
Objetivos:

- Integrar los contenidos prácticos correspondientes a la programación orientada a objetos.
- Reforzar las destrezas necesarias para resolver una situación problemática utilizando la programación orientada a objetos en Smalltalk.
- Modelar una solución de un problema mediante el paradigma orientado a objetos.

Los temas que se evalúan en este trabajo práctico son los que se detallan a continuación: Objetos y mensajes en Smalltalk. Mensajes unarios, binarios y de palabras claves. Bloques. Composición de clases. Abstracción. Encapsulamiento. Herencia. Polimorfismo. Colecciones.

Caso de Estudio:

La empresa TransportMatic S.A. de instalación de cintas transportadoras ganó la licitación para el reemplazo de todo el conjunto de cintas transportadoras de equipajes del Aeropuerto de la Ciudad de Córdoba Ing. Ambrosio Travella. Por ello deberá hacerse cargo del rediseño de la estructura de cintas transportadoras que llevan los equipajes desde los escritorios de checkin en los que los pasajeros despachan sus equipajes hasta la terminal de carga donde se cargan los equipajes en el transporte que los lleva al avión.



Entre las tareas previas a la puesta en marcha de la nueva estructura de cintas transportadoras, se le solicita a TransportMatic la construcción de un modelo de software para simular el funcionamiento del conjunto de cintas transportadoras y han contratado a vuestro equipo de desarrollo de software para la construcción del modelo de objetos que de soporte al software de simulación.

Para desarrollar el modelo solicitado debe comprender los distintos componentes que se piensan instalar en la estructura de Cintas del Aeropuerto.

En primer lugar, tenemos los equipajes de los pasajeros, de los mismos se conoce previamente el número del vuelo del pasajero, el código de pasajero, una bandera indicando si es equipaje prioritario.

Al llegar a la terminal de carga se debe poder observar en los equipajes de la simulación, el tiempo total insumido en atravesar la estructura de cintas, la cantidad total de metros recorridos y una bitácora que incluya: por qué receptora ingresaron, por qué cintas pasaron y a qué despacho llegaron finalmente.

Luego tenemos las cintas transportadoras, las cintas tienen un nombre que las identifica, las mismas pueden ser de 2 tipos diferentes, las cintas receptoras y las cintas centrales, todas tienen o bien la siguiente cinta en la que

depositan los equipajes o bien el despacho al que los transportan. Si una cinta no tiene configurado ninguna de las dos salidas o ambas a la vez está en conflicto y por lo tanto se detiene y se pone en estado en conflicto.

En primer lugar, tenemos las cintas receptoras, las mismas tienen por función pesar los equipajes y asignar el número de identificación del punto de recepción al equipaje y transportarlos hasta están conectados a una cinta central, no agregan tiempo insumido porque siempre son solo de acceso a la cinta principal, existen tantos receptores como escritorios de checkin, y al llegar, los pasajeros se dirigen al primer receptor que se encuentre libre.

Nota: para los efectos de la simulación el peso va a ser suministrado como un valor independiente (colaborador externo) en conjunto con el equipaje a transportar.

En el software de simulación, los receptores van a ser nucleados por la Recepción que va a tener la colección de receptores disponibles y la responsabilidad de elegir uno al azar para el equipaje que llega.

Luego tenemos las cintas centrales, las mismas tienen una longitud en metros y una velocidad de movimiento en centímetros por segundo y en ella se pueden configurar selectores que desvíen los equipajes a distintos despachos especiales. Los Selectores están ordenados de acuerdo con la cantidad de metros a los que están configurados sobre la cinta.

Un equipaje debe pasar por cada uno de los selectores, si no es desviado en ninguno de ellos entonces llega a la siguiente cinta central o al despacho normal. Si alguno de los selectores lo desvía entonces pasará a la cinta que configure el selector como salida.

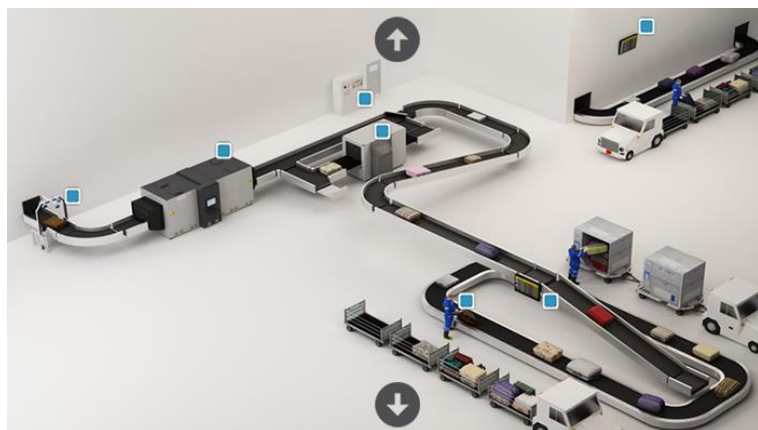
En todos los casos la cinta agrega al equipaje el tiempo que insume recorrer la cinta en caso de ser desviado se debe asignar el tiempo hasta el selector (tomando en cuenta los metros a los que está configurado) y sino el tiempo de recorrer toda la cinta.

Los selectores son sensores capaces de reconocer características en los equipajes y desviarlos de la cinta mediante una herramienta mecánica, tienen configurado la cantidad de metros en la cinta a la que están conectados. Existen

2 tipos de selectores, selector por peso y selector de aleatorio.

Los selectores por peso tienen un peso determinado configurado y seleccionan los equipajes que pesan más que el peso configurado para ser enviados al despacho de equipajes pesados.

Los selectores aleatorios tienen un valor de n configurado y seleccionan de forma automática un equipaje cada n que pasan para ser enviado al despacho de control de equipajes.



Finalmente, los despachos reciben los equipajes y de ellos son retirados para la carga en los vuelos. Tienen una etiqueta que indica si son despachos normales o de control. Tienen una colección con los equipajes recibidos y la capacidad de tomar los equipajes de allí para cargarlos en el avión.

Además, un aeropuerto tiene al menos 1 despacho de cada tipo y por lo tanto los 2 despachos están conectados a la Terminal de carga que es la que retira los equipajes para la carga.

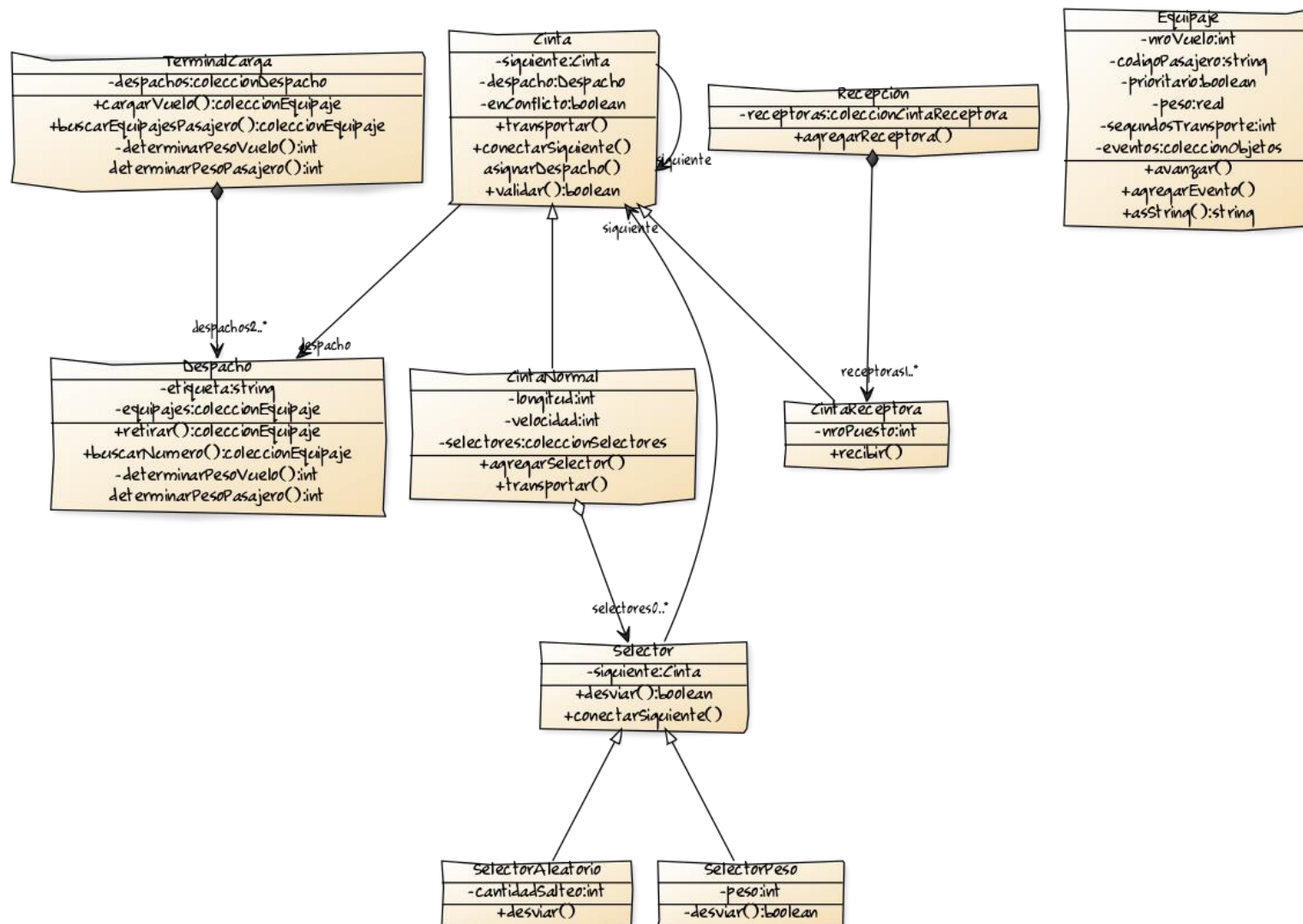
La terminal, además de ser el punto de interacción con todo el equipaje a despachar, tiene la responsabilidad de validar que entre sus despachos cuente con al menos 1 despacho de cada tipo (es decir al menos 1 despacho normal y 1 despacho de control).

Cátedra Paradigmas de Programación Trabajo Práctico Nro. 1: Programación Orientada a Objetos

Ciclo 2019

Diagrama de Clases:

Este diagrama es el propuesto para la resolución de lo solicitado por el enunciado, no es obligatorio trabar estrictamente según el diagrama, pero tener en cuenta que los tests unitarios estarán planteados en base a él.



En base a lo descripto anteriormente usted debe desarrollar las clases necesarias orientado a objetos en Smalltalk para soportar que satisfaga los siguientes requerimientos:

1. Implemente todas las clases del modelo con todos los atributos y métodos que crea necesarios, notar que el diagrama de clases es una guía y no es necesario respetarlo estrictamente (sin embargo, tenga en cuenta que los tests unitarios suministrados, asumen que usted llamó a métodos y clases según lo que figura en el diagrama).
2. Implementar los comportamientos necesarios para permitir configurar una estructura de cintas para simular su funcionamiento, esto implica, poder crear y configurar cintas, tanto receptoras como normales, conectar las cintas receptoras con la cinta normal que corresponda, conectar las cintas normales a los despachos o cintas siguientes, crear selectores y agregarlos a las cintas normales.
Finalmente crear una instancia de Recepción y agregarle las cintas receptoras y una instancia de Terminal agregar los despachos.
3. En la clase recepción implementar los comportamientos necesarios para despachar un equipaje y que el mismo sea transportado hasta llegar a un despacho.
4. En la clase Terminal
 - 4.1. Implementar los comportamientos necesarios para que la terminal permita buscar los equipajes asociados a un pasajero específico indicando el código del pasajero. El comportamiento debe retornar una colección con los equipajes correspondientes y en el caso de no existir equipajes debe retornar nil.
 - 4.2. Implementar un comportamiento determinarPesoVuelo que dado un número de vuelo retorne el peso total de los equipajes para ese vuelo. Tener en cuenta que se deben recuperar los equipajes de todos los despachos.
 - 4.3. Implementar un comportamiento determinarPesoTotalPasajero que dado un código de pasajero retorne la suma de los pesos de todos sus equipajes.
 - 4.4. Implementar los comportamientos necesarios para que la terminal permita simular la carga de un vuelo, es decir un comportamiento cargarVuelo que a partir de un número de vuelo elimine todos los equipajes de ese vuelo de todos los despachos y los retorne como una colección de equipajes ordenados de tal forma que los equipajes con prioridad aparezcan primero que los equipajes normales.

Se sugiere dividir en el trabajo en etapas, de manera que pueda ir resolviendo los requerimientos solicitados a medida que avanzan las clases teórico-prácticas de la materia, tal como se indica a continuación:

- **Etapas 1:** desarrollar la clase Equipaje (sin los eventos por ahora) con método de inicialización, de acceso y asString, además del método, avanzar que recibe centímetros y velocidad dada en centímetros por segundo e incrementa el tiempo insumido).
- **Etapas 2:** Desarrollar las jerarquías de clases de Cintas, Selectores y los despachos, en el equipaje agregar los eventos y desarrollar el comportamiento de transportar el equipaje para que llegue desde una cinta receptora a un despacho.
- **Etapas 3:** Desarrollar las clases Recepción y Terminal con los comportamientos solicitados.

Nota: cada etapa debe ir trabajando en el Playground para comprobar el funcionamiento de lo que se desarrolla. Adicionalmente se proveerán test unitarios para las distintas etapas de forma de poder validar lo desarrollado en cada iteración (reiteramos aquí que para que los test tengan uso real usted debe haber implementado las clases según el diagrama propuesto puesto que los test están preparados de ese modo).

Consignas generales:

- Defina adecuadamente la jerarquía de clases, y asigne en cada clase los atributos y métodos que correspondan según el criterio del grupo.
- Realice la codificación completa en Smalltalk de todas las clases involucradas en el diseño de la solución problemática. Debe haber coherencia total entre el diagrama de clases propuesto y la implementación en Smalltalk. Usted puede reformular el diagrama propuesto, en tal caso deberá presentarlo como parte del trabajo. Siéntase libre de agregar todo el comportamiento extra en las clases que considere necesario.
- Escriba las líneas de código necesarias en la ventana Playground para hacer funcionar el programa y verifique en la ventana Transcript todas las salidas de información generadas.
- Reutilice adecuadamente los comportamientos que sean necesarios.
- Tenga presente la delegación de responsabilidades para cada requerimiento según corresponda.

Criterios de evaluación:

- Identificación de todas las clases involucradas, y asignación correcta de atributos y responsabilidades de cada una.
- Adecuado diseño del diagrama de clases: jerarquía de clases y demás relaciones bien especificadas y representadas, etc.
- Claridad y completitud del diagrama de clases.
- Prolijidad en la codificación en Smalltalk: identificadores, comentarios, envío de mensajes, etc.
- Manejo adecuado de **TODAS** las propiedades esenciales de la programación orientada a objetos en la resolución propuesta.
- Delegación apropiada de responsabilidades en las clases involucradas.
- Reutilización conveniente de los comportamientos implementados.
- Validaciones.
- Correcta utilización de métodos:

- ✓ unarios;
- ✓ binarios;
- ✓ de palabras clave: de una, dos, y más palabras claves.

Se deberán utilizar todos estos tipos de mensajes en la resolución propuesta en forma adecuada.

- Elección y uso apropiado de las colecciones en cada caso.
- Variedad de mensajes utilizados de las diferentes colecciones utilizadas, y de otros objetos.
- Uso de mensajes específicos tanto de colecciones como de otros objetos, para cada caso.
- Correcta identificación y usos de métodos polimórficos.
- Generación y visualización correcta de todas las salidas de información.
- Cumplimiento de todas las responsabilidades solicitadas.
- Realización de las pruebas solicitadas en el Playground, instanciando adecuadamente los objetos, y realizando las colaboraciones necesarias. Recuerde que los resultados del envío de los mensajes a los objetos se deberán visualizar en la ventana Transcript.

Tabla de valoración de los ítems evaluados

Nro. de ítem	Ítems o requerimiento a evaluar	Puntaje	Observaciones	Obtenido
1	Implementación de clases, con métodos comunes y específicos solicitados en las fichas.	20		
2	Definición de métodos polimórficos en las jerarquías Cintas y Selectores.	20		
3	Despachar Equipaje, contemplando que llegue al despacho.	15		
4	Comportamiento de buscar equipajes por pasajero.	15		
5	Comportamientos de determinar Peso por Vuelo y por Pasajero.	10		
6	Comportamiento de Cargar el Avión por Nro de Vuelo.	20		
	Total	100		

Condiciones de entrega:

- Este trabajo práctico integrador **se deberá realizar en forma grupal.**
- Deberán nombrar el **archivo comprimido** con el siguiente formato: TP1_NroLeg1erIntegranteApellido1erIntegrante_NroLeg2doIntegranteApellido2doIntegrante_NroLeg3erIntegranteApellido3erIntegrante. El orden en el cual deberán colocar los legajos y apellidos cada integrante en el nombre del archivo comprimido será de acuerdo al orden alfabético de los apellidos de los integrantes, desde la A hasta la Z.
- Se deberá subir una carpeta comprimida que contenga en su interior el archivo .ST correspondiente a la codificación en Smalltalk; un archivo de texto con el código necesario a colocar en la ventana Playground para hacer funcionar el programa; y el diagrama de clases completo correspondiente a la solución de la situación problemática anteriormente donde se explique para cada clase/método cuáles fueron los desafíos y que solución plantearon a ellos.
- Plazo máximo de entrega de este trabajo: **HASTA el 07/09/2019 a las 23:55 hs.**
- Todas las consultas de este TP1 pueden realizarlas a través del foro del aula virtual.
¡Éxito a todos!