



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

**Metodologías de Programación
APREF
Segundo semestre 2011**

Profesores a cargo: Alicia Díaz y Roxana Giandini

Días, horarios y aulas: de martes 11 a viernes 14 de octubre de 9 a 13-Aula Alfa
Actividad previa al APREF: lunes 3/10-8.30 hs. Lugar a definir.

Cronograma de Actividades del APREF:

	Contenido	Actividad complementaria
Clase 1	Smalltalk: el lenguaje con ejercitación. self, super Clases: <ul style="list-style-type: none">● Magnitude,● Boolean,● BlockClosure,● Collection: clases, iteradores	Ejercicio tipo: 1. ranking- tag cloud 2. Mailbox with Filter
Clase 2	UML: <ul style="list-style-type: none">● Diagramas de casos de uso Ejercitación: Modelado básico de <ul style="list-style-type: none">● Diagramas de secuencia● Diagramas de Clases UP: El Proceso de desarrollo Unificado	Ejercitación: Modelo básico de Comercio Electrónico con UML, aplicando UP
Clase 3	Diseño OO: <ul style="list-style-type: none">● Composición vs. Herencia● Diseño con herencia, polimorfismo, self● Doble dispatching	1. Comercio electrónico: Compra libros. Diseño detallado e implementación en Smalltalk



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

Clase 4	Recursion: <ul style="list-style-type: none">• delegacion recursiva• Propagación• self delegation	1. Red de empresas prestadoras de servicios 2. Leer articulo sobre recursion con Objetos.
----------------	---	--

Actividad Previa:

Lunes 3/10 8.30 hs.

Para agilizar el desarrollo del curso y lograr un mejor aprovechamiento de las clases, los alumnos deberán repasar previamente, los conceptos básicos de la POO, del ambiente de desarrollo Smalltalk y del lenguaje de modelado UML.

En este encuentro podrán hacer consultas sobre estos conceptos básicos.

Para esto, pueden revisar las clases teóricas de la cursada y/o consultar la siguiente bibliografía, que se encuentra disponible en la biblioteca de la Facultad:

The Object-Oriented Thought Process, Matt Weisfeld, Third Edition, Pearson Education, Addison Wesley. ISBN-13: 978-0-672-33016-2

Introduction to Object-Oriented Programming, An (3rd Edition), Thimoty Budd, Addison Wesley; 3 edition (2001), ISBN-10: 0201760312

Joy of Smalltalk. Ivan TOMEK,
<http://plato.acadiu.ca/courses/comp/tomek/jos.htm>

Smalltalk by example. Alex Sharp, McGraw Hill; (1997), ISBN: 0079130364

UML GOTA A GOTA. FOWLER MARTIN, SCOTT KENDALL. Editorial ADDISON-WESLEY IBEROA. Edición 1999, en Rústica ISBN 9684443641



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

EL LENGUAJE UNIFICADO DE MODELADO. BOOCH GRADY, JACOBSON IVAR,
RUMBAUGH JAMES. Editorial ADDISON-WESLEY IBEROA. Edición 2000, en Rústica. ISBN
8478290281

Ejercicios para Resolver:

Ejercicio ranking- tag cloud

Parte A

1. Modele la clase Ranking cuyas instancias son una colección que permite objetos repetidos y los mantiene ordenados (rankeados) de acuerdo a la cantidad de ocurrencias de los mismos. En *unRanking* es como un bag, pero ordenado por ocurrencias.
2. Implemente los métodos para
 - a. crear *unRanking*
 - b. agregar un objeto a *unRanking* (tenga en cuenta que este objeto puede o no existir en el ranking)
 - c. eliminar un objeto de *unRanking*
 - d. recuperar el objeto top en *unRanking*
 - e. recuperar los primeros N objetos en *unRanking*

Parte B

Motivación: En la actualidad, una de las actividades más comunes en la web es agregarle etiquetas a los recursos web. Esto permite en un futuro la recuperación de estos recursos de manera más eficiente. Entre los más conocidos están Flickr donde los usuarios comparten fotografía y YouTube donde los usuarios comparten videos. Cada recurso solo puede ser subido al sitio por un solo usuario.

La actividad de etiquetado consiste en que cada usuario le asocia a un recurso una serie de *tags* (etiquetas). Por ejemplo, el usuario *Juan* viendo en Flickr una foto de la ciudad de la



**UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA**

Hacer:

1. definir los casos de usos
2. Realizar un diagrama de clases que modele el etiquetado de recursos por usuarios
3. Para las siguientes funcionalidades, definir los diagramas de secuencia e implementar en ST haciendo uso de la clase Ranking.
 - a. anotar un recurso con un tag por un usuario, para lo que hay que mantener el *tagCloud* del sistema y el *tagCloud* del usuario
 - b. recomendar tags para un recurso. Recomienda los tag que fueron más usados en el sistema para anotar un recurso dado.
 - c. encontrar los recursos que fueron anotados por un conjunto dado de tags
4. Implementar en Smalltalk a, b y c.

Mailbox with Filter

1. Modele la clase Email cuyas instancias modelan un correo electrónico definido por su emisor (from), receptor (to), asunto (subject) y cuerpo (body). Cada parte del unEmail es un texto. Implementarla en Smalltalk
2. Modele la clase Folder cuyas instancias son una colección de emails. Defina e implemente en Smalltalk los métodos necesarios para crear, agregar y eliminar elementos de un folder. Un email se agrega a un folder a través del mensaje #archive.
3. Extienda el diseño previo para modelar también folders con filtros. Ese filtro es un *bloque* que sirve para decidir si el email se debe archivar o no en el folder. Para archivarlo el email debe satisfacer el filtro. Los filtros definen una condición booleana definida a través de las componentes de un mail. Por ejemplo las figuras de abajo definen dos filtros de dos folders distintos en GMail: "Objetos1" y "CosasUrgentes". El primero implica la condición:

anEmail to contains: 'objetos1@googlegroups.com'

y el segundo la condición:

anEmail subject contains:'urgente' or:[anEmail body contains:'urgente']



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

Estos filtros se especifican en el momento de crear el Folder con filtro.

Implemente en Smalltalk los métodos necesarios para crear un folder con filtro y agregar un email al folder con filtro.

Nota: el mensaje `contains:aString` devuelve verdadero si `aString` esta contenido en el string receptor del mensaje.

by Google **Create a Filter** [Hide filter options](#)

Choose search criteria Specify the criteria you'd like to use for determining what to do with a message as it arrives. Use "Test Search" to see which messages would have been filtered using these criteria. Messages in Spam and Trash will not be searched.

From:

To:

Subject:

Has the words:

Doesn't have:

Has attachment

[Show current filters](#)

by Google **Create a Filter** [Hide filter options](#)

Choose search criteria Specify the criteria you'd like to use for determining what to do with a message as it arrives. Use "Test Search" to see which messages would have been filtered using these criteria. Messages in Spam and Trash will not be searched.

From:

To:

Subject:

Has the words:

Doesn't have:

Has attachment

[Show current filters](#)

1. Modele la clase MailBox cuyas instancias son colecciones de folders. Cuando llega un email al cliente de correo electrónico es archivado en el mailbox de la siguiente forma: todo email es archivado en un folder especial que se llama inbox y luego se lo archiva en todos aquellos folders para los que el email satisface el filtro. Diseñe una solución con objetos que permita resolver los siguientes problemas.



**UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA**

- a. Describa el diagrama de clases que conceptualiza el problema descripto arriba.
 - b. Escriba en Smalltalk el método para crear un mailbox que originalmente solo contiene el folder inbox. Indique cuales son la variables de instancia de la clase Mailbox
 - c. Escriba en Smalltalk un método que agregue un folder de cualquier clase al mailbox.
 - d. Describa el diagrama de secuencia para el método archive:anEmail de la clase MailBox
 - e. Escriba en Smalltalk el método archive:anEmail de la clase MailBox
 - f. Escriba el código necesario en el Workspace para crear un mailbox como el descripto en el ejemplo con los filtros descriptos en el ejemplo.
 - g. Escriba en Smalltalk un método que devuelve el email mas archivado. Si varios cumplen esa condición, devuelve todos.
2. Justifique el diseño e implementación de la clase Folder y Folder con filtros.
 3. Justifique el diseño e implementación de la clase MailBox.

Comercio electrónico: Compra de libros

A. En un sitio de comercio electrónico se venden libros, que pueden ser digitales o impresos. Los libros tienen: Título, Autores, Edición, editorial y precio. En el sitio existen clientes, los que pueden tener categoría normal o VIP. Todos los clientes, cuentan con un crédito para comprar libros. Cuando un cliente compra un libro, se debita el precio de su crédito.

El precio final de un libro digital es el precio de costo más una comisión del 2%, el precio final de un libro impreso tiene una comisión del 1% pero se agregan \$20 de gastos de envío. Un cliente, puede hacer recargas de su crédito, teniendo en cuenta que a los clientes VIP con cada recarga mayor a \$50 se le bonifica el 5% del valor de la misma.

Todos los clientes deben saber que libros compraron. Cuando se registra la compra de un libro para un socio, se debe contemplar que el libro puede cambiar de precio con el tiempo, por ello se debe registrar el precio que se pago del libro en el momento de la compra.



**UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA**

Un cliente puede pasar de ser VIP a Normal de acuerdo a ciertas particularidades de su comportamiento (por ejemplo si en un año supera un límite de compra el cliente es categorizado como vip caso contrario como normal).

Se pide una solución aplicando el proceso **UP**:

1. Escriba en formato extendido los casos de uso correspondientes al problema planteado arriba.
2. Diseñe un modelo conceptual orientado a objetos en base a los conceptos que surjan de los casos de uso
3. escriba los contratos para las operaciones principales que surjan de los casos de uso
4. Escriba los diagramas de secuencia correspondientes a las operaciones:
 - a. Comprar un libro
 - b. Saber si un cliente pasa de categoría
 - c. Recargar crédito
 - d. Cambiar la categoría de un cliente, es decir si es VIP pasa a ser Normal,, y si es Normal pasa a ser VIP.
5. Diseñe el Diagrama de Clases del sistema en base al modelo conceptual, agregándole comportamiento.
6. Implemente en Smalltalk los mensajes de inicialización de cada clase
7. Implemente en Smalltalk los mensajes enunciados en 3.
8. Modifique su solución del ejercicio anterior para considerar ahora que: los clientes con categoría VIP no pagan gastos de envíos.

Red de Empresas

A. Modelar (realice un diagrama de clases e implemente en Smalltalk) una red de empresas prestadoras de servicios. La red está formada por un grupo de empresas. Cada empresa presta uno o mas servicios. Un servicio puede ser prestado completamente por una sola empresa, o puede requerir de sub-servicios prestados por otras empresas que deben estar en la red. Por ejemplo, tenemos una red formada por 6 empresas: Salon de Fiestas A, Floreria B, Confiteria C, Confiteria H y Musicalizacion M. Los servicios prestados por SecuritySA es "proveer seguridad". Los servicios prestados por PanificaciónSH son "proveer



**UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA**

lunch” y “proveer postres”. Los servicios prestados por ParísSA es “proveer postres”. Los servicios prestados por OrnamentosSA es “proveer ornamentación”. Los servicios prestados por EventsSA es “organizar evento”.

En este ejemplo, todos los servicios, excepto “organizar evento”, son prestados por cada empresa, sin requerir subservicios. Para “organizar evento” EventsSA le solicita “proveer lunch” a PanificaciónSH, “proveer postres” a la ParísSA por ejemplo y “proveer seguridad” a SecuritySA

B. Implementar los siguientes mensajes (y todos los demás que se relacionen con ellos):

- agregar una nueva empresa proveedora en la red.
- eliminar un servicio X de una empresa
- Eliminar una empresa E de la red. (para este comportamiento, construya el diagrama de secuencia e implemente)

Importante: tenga en cuenta (tanto al agregar como al eliminar elementos a la red) que la red debe estar siempre consistente, es decir que todo sub-servicio requerido por una empresa de la red debe ser prestado por alguna otra empresa de la red.



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA