



# Orientación a Objetos 1 - 2012

## Práctica 9

### Ejercicio 1

Dado el modelo UML de la Figura 1 implemente en Smalltalk las clases Estudiante y Curso. A continuación se detalla el protocolo de la clase Curso:

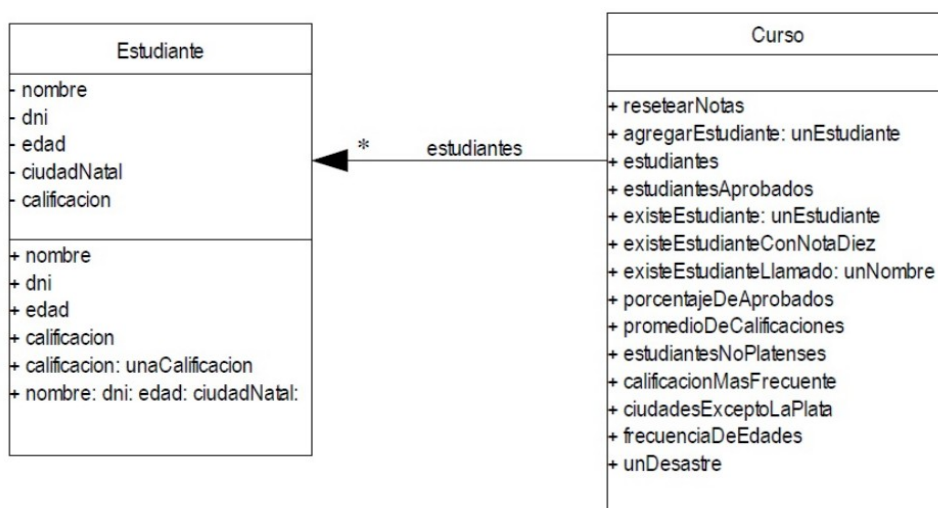


Figura 1: Diagrama de clases

La siguiente lista define los objetivos de los métodos de la clase Curso:

```
#resetearNotas
Pone en cero las calificaciones de todos los estudiantes.

#agregarEstudiante: unEstudiante
"Agrega unEstudiante al curso"

#cantidadDeEstudiantesInscriptos
"Retorna la cantidad de alumnos que se inscribieron al curso"

#estudiantes
"Retorna la colección de estudiantes del curso"

#estudiantesAprobados
"Retorna una colección con todos los estudiantes que aprobaron el curso
(calificación superior a 4)"
```



# Orientación a Objetos 1 - 2012

## Práctica 9

`#existeEstudiante: unEstudiante`

"Indica si unEstudiante se encuentra inscripto en el curso"

`#existeEstudianteConNotaDiez`

"Determina si algún alumno obtuvo la calificación 10"

`#existeEstudianteLlamado: aString`

"Indica si el estudiante llamado aString se encuentra inscripto en el curso"

`#porcentajeDeAprobados`

"Retorna en porcentaje de estudiantes aprobados"

`#promedioDeCalificaciones`

"Calcula el promedio de las calificaciones obtenidas por los alumnos"

`#estudiantesNoPlatenses`

"Retorna una colección con todos los estudiantes que no nacieron en La Plata"

`#calificacionMasFrecuente`

"Retorna la calificación más frecuente del curso"

`#ciudadesExceptoLaPlata`

"Retorna una colección, sin repeticiones, conteniendo los nombres de todas las ciudades (excepto La Plata) donde nacieron los alumnos inscriptos al curso"

`#unDesastre`

"Retorna verdadero si todos los estudiantes desaprobaron el curso"

`#frecuenciaDeEdades`

"Retorna un diccionario que asocia las edades de los estudiantes con la frecuencia en la que aparecen. Esta información luego será utilizada para generar un histograma (ver imagen) cuyo eje x serán las edades y cuyo eje y serán la frecuencia de las ocurrencias"

La Figura 2 muestra un ejemplo del histograma que se desea realizar con la información provista por `#frecuenciaDeEdades`.



# Orientación a Objetos 1 - 2012

## Práctica 9

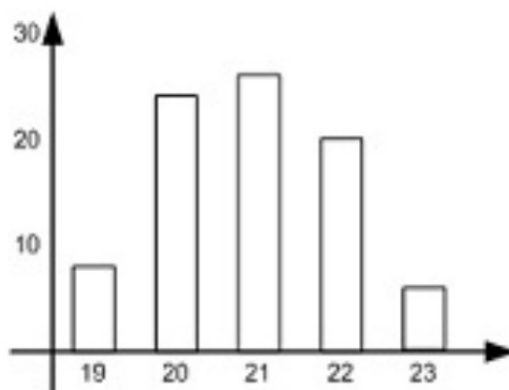


Figura 2: Ejemplo de Frecuencia de Edades

Utilice los test cases provistos en el archivo de material adicional de la práctica para testear su implementación.

### Ejercicio 2

Modifique la implementación de la clase `DateLapse` realizada en la práctica 7 para que en lugar de trabajar con una fecha de inicio (`from`) y una fecha de fin (`to`), utilice una única variable de instancia (`dates`). Dicha variable de instancia hará referencia a una colección con todas las fechas en el rango de un `DateLapse`.

Tareas:

1. Compare las tres implementaciones de `DateLapse` (`from/to`, `from/sizeInDays` y `dates`). ¿Qué ventajas y desventajas puede mencionar de cada una?
2. Verifique su implementación usando los test cases que dispone para `DateLapse` de la práctica 7.

### Ejercicio 3

Sea un web browser que puede abrir tabs. El mismo debe poder abrir hasta 10 tabs. Para abrir un tab es necesario proporcionarle la URL (un `String`). Cada tab puede reportar la cantidad de memoria consumida en KBytes. En cualquier momento el usuario puede perderle a browser cerrar un tab (discuta con el ayudante la mejor manera de referenciar al tab). Cuando el browser recibe el mensaje `close`, debe cerrar todos sus tabs. El browser debe ser capaz de retornar la memoria consumida, que es igual a la suma de la memoria consumida por los tabs + 500KB.

Tareas:

1. Lea atentamente el enunciado y consulte con el ayudante sobre cuestiones no especificadas en el enunciado. Por ej: es posible abrir un browser y que no tenga ningun tab?
2. Realice un diagrama de clases.



# Orientación a Objetos 1 - 2012

## Práctica 9

3. Realice un diagrama de secuencia para la creación de un tab.
4. Realice un diagrama de secuencia para el cierre de un browser con 2 tabs.
5. Implemente en Smalltalk.

### Ejercicio 4

Para continuar con la implementación del ejercicio de la computadora que comenzó en la práctica 8, extienda el modelo con los siguientes mensajes en el protocolo de la clase Computadora:

```
#memoria:unString
"almacena unString como contenido de la memoria"

#archivar
"pasa el contenido de la memoria al disco rígido"

#contenidoMemoria
"imprime en Transcript el contenido de la memoria"

#contenidoDisco
"imprime en Transcript el contenido del disco rígido"
```

Además de la clase `Computadora` modifique todo lo necesario para completar la funcionalidad indicada. Utilice los mensajes que definió sin implementar en la práctica 8 para completar el almacenamiento en memoria y disco rígido.

Tareas:

1. Actualice el diagrama de clases con el diseño completo.
2. Realice un diagrama de secuencia para mostrar la interacción entre objetos que tiene lugar cuando la computadora recibe `#memoria:"Esto va a la memoria"`.
3. Implemente todas las modificaciones.

### Ejercicio 5

Dado el ejercicio 1 de la práctica 6, donde se han modelado e implementado dos tipos de cuentas bancarias: Caja de Ahorro y Cuenta Corriente, se desea extender la aplicación para generar los resúmenes de las cuentas. Para esto, el banco requiere registrar los movimientos históricos realizados en las mismas. Existen dos tipos de movimientos a ser registrados: extracciones y depósitos. De cada movimiento se requiere conocer la fecha en la que fue realizado y además el monto del mismo.

Se espera que el sistema permita generar reportes similares al siguiente:

```
Caja De Ahorro (saldo actual: $2850, disponible: $2850):
Depósito de $3000 el 1/11/2012
```



# Orientación a Objetos 1 - 2012

## Práctica 9

Extracción de \$150 el 2/11/2012

Cuenta Corriente (saldo actual: \$2360, disponible: \$3660):

Depósito de \$2500 el 30/10/2012

Extracción de \$140 el 2/11/2012

Para poder realizarlo, considere la clase `Banco` que debe entender el mensaje `#emitirResumen` a partir del cual se imprime en el Transcript la información de las cuentas como se muestra arriba.

Note que para la Cuenta Corriente, el monto disponible es mayor al saldo debido al límite de descubierto.

### Tareas:

1. Modifique el modelo de clases conseguido en el Práctica 6 para contemplar los nuevos requerimientos.
2. Realice un diagrama de secuencia para el escenario de ejemplo.
3. Implemente en Smalltalk.
4. Responda a las siguientes preguntas y discuta sus respuestas con un ayudante:
  1. Para que cada cuenta imprima en el Transcript los detalles necesarios, ¿Definió todo lo necesario en cada subclase de `CuentaBancaria`, o pudo abstraer parte del comportamiento en la superclase?
  2. ¿En que lugares de su implementación existe polimorfismo?