Algoritmos y programación 3

Trabajo práctico 1:

Algoslide

2014-1c

(trabajo individual)

# 

Fecha de entrega: **semana del 24 de Abril**, en fecha y hora de la práctica en la cual está inscripto el alumno (ver detalles de forma de entrega en este documento)

Nombre:

Padrón:

Email:

Calificación:

Corrector:

[Introducción](#h.7eb8uz5tbqyh)

[Objetivo del trabajo](#h.g9msn02n0zc7)

[Consigna](#h.rfc6cab0r7uo)

[Alcance](#h.bafec6oebuor)

[Especificaciones](#h.vt1xnxcrpu41)

[Informe](#h.hq4adfjsfhzl)

[Supuestos](#h.vj0lxwyygumd)

[Modelo de dominio](#h.ecynjblhgm6q)

[Diagramas de clases](#h.nspcwtfygv3o)

[Detalles de implementación](#h.5ibuq4tbbj5j)

[Excepciones](#h.s6mybycr0i3i)

[Diagramas de secuencia](#h.7inwqh1ish2h)

[Código fuente](#h.fh7xntrg6aos)

[Forma de Entrega](#h.x1ohive8uq5g)

[Checklist de corrección](#h.geazv7165lh0)

[Carpeta](#h.iz6ldlyb0o84)

[Generalidades](#h.1ggp9xky10g5)

[Modelo](#h.8blb0cmjhbd1)

[Diagramas](#h.amyb77fpuckk)

[Código](#h.xtmgw4oov6i)

[Generalidades](#h.xuljpjhwfamz)

# Introducción

## Objetivo del trabajo

Aplicar los conceptos enseñados en la materia a la resolución de un problema, trabajando individualmente utilizando Smalltalk.

## Consigna

Desarrollar el modelo de clases utilizando TDD, de un Software de Presentación de Diapositivas tipo PowerPoint o LibreOffice Impress. Las especificaciones de la aplicación se muestran a continuación. Dado que el diseño se llevará a cabo utilizando TDD como herramienta se pedirá pruebas unitarias y de integración completas.

## Alcance

Se deberá desarrollar el modelo completo de la aplicación, sin incluir la interfaz gráfica.

Deberá entregarse:

* todas las clases con sus métodos , organizados en una o mas categorías según criterio del alumno.
* conjunto de pruebas unitarias que muestren el uso de la biblioteca y su correcto funcionamiento.
* documentación completa del código fuente
* documentación completa del diseño de clases (diagramas UML de clases y secuencia)

Se deberá desarrollar el modelo completo de la aplicación, sin incluir la interfaz gráfica.

Para esto se especificará una API la cual estará descrita por un conjunto de pruebas de integración. Este conjunto de pruebas estará descripto por la clase PresentacionTests.

Como es de esperar de cualquier biblioteca, la misma deberá contar con:

* Conjunto de pruebas unitarias que muestren el uso de la biblioteca y su correcto funcionamiento.
* Documentación completa del código fuente
* Documentación completa del diseño de clases (diagramas UML de clases y secuencia)

# Especificaciones

El software de Presentación deberá soportar las siguientes funcionalidades:

* Agregar diapositivas a la presentación
* Agregar texto a las diapositivas
* Cortar y pegar diapositivas entre presentaciones
* Distintos tipos de transiciones entre diapositivas
* Distintas plantillas que afectan a cómo se muestra una diapositiva

El detalle de estas funcionalidades se encuentra especificado en las pruebas de integración del archivo *TP1-Tests.st*.

# 

# 

# Informe

## Supuestos

[explicar todos los supuestos que no hayan quedado claro en el enunciado]

## Modelo de dominio

[Explicar cómo se encaró el TP]

### Diagramas de clases

[Uno o varios diagramas de clases, mostrando la relación estática entre las clases, pueden agregar todo el texto necesario para aclarar y explicar su diseño, recuerden que la idea de todo el documento es que quede documentado y entendible como está hecho el TP]

### Detalles de implementación

[hay ciertas clases que puede resultar interesante como están implementadas internamente, en esta sección cuenten sobre esas clases]

### Excepciones

[Explicar las excepciones creadas y con qué fin fueron creadas]

### Diagramas de secuencia

[Mostrar las secuencias interesantes que hayan implementado. Pueden agregar texto para explicar si algo no queda claro]

### Código fuente

[Código fuente impreso de las clases del dominio (no pruebas).]

# Forma de Entrega

El TP deberá ser desarrollado en su totalidad dentro dos paquetes:

* TP1.st: donde se ubicarán todas las clases que resuelven el problema.
* TP1-Tests.st: que deberá contener todas las clases de pruebas creadas por el alumno.

Para la entrega deberá hacerse un fileout de estos dos paquetes (sólo estos dos, las pruebas especificadas por la cátedra NO deben entregarse).

El entregable será un archivo en formato .zip, que deberá llamarse tp1-<numero\_de\_padron>.zip y que deberá contener los siguientes archivos:

* TP1.st
* <numero\_de\_padron>-tests.st
* <numero\_de\_padron>-informe\_tp1.odt

Este archivo zip deberá ser subido al sistema de corrección automática cuando sea indicado por la cátedra.

# Checklist de corrección

Esta sección es para uso exclusivo de la cátedra, por favor no modificar.

## Carpeta

### Generalidades

¿Son correctos los supuestos y extensiones?

¿Es prolija la presentación? (hojas del mismo tamaño, numeradas y con tipografía uniforme)

### Modelo

¿Está completo? ¿Contempla la totalidad del problema?

¿Respeta encapsulamiento?

¿Hace un buen uso de excepciones?

¿Utiliza polimorfismo en las situaciones esperadas?

### Diagramas

¿Está completo el diagrama de clases?

¿Está bien utilizada la notación en el diagrama de clases?

¿Está completo el diagrama de secuencia?

¿Es consistente con el diagrama de secuencia con el diagrama de clases?

¿Está bien utilizada la notación del diagrama de secuencia?

## Código

### Generalidades

¿Respeta estándares de codificación?

¿Está correctamente documentado?