

Trabajo Práctico 1 — Smalltalk

[7507/9502] Algoritmos y Programación III
Curso 2
Primer cuatrimestre de 2020

Alumno:	Paredes Ramirez, Luis Jose
Número de padrón:	104851
Email:	lparedesr@fi.uba.ar

Índice

1. Introducción	2
2. Supuestos	2
2.1. Pincel	2
2.2. Rodillo	2
2.3. Alba	2
2.4. Venier	2
3. Diagramas de clase	3
4. Detalles de implementación	4
4.1. Nota de los tests	4
5. Excepciones	5
6. Diagramas de secuencia	5

1. Introducción

El presente informe reúne la documentación del primer trabajo práctico de la materia Algoritmos y Programación III que consiste en desarrollar un sistema que permita encontrar el mejor presupuesto de un pintor de acuerdo a las necesidades de los clientes.

Este trabajo es desarrollado usando los conceptos del paradigma de la orientación a objetos utilizando el sistema Smalltalk.

2. Supuestos

Los siguientes supuestos se tomaron como constantes que siempre se cumplieran en el trabajo práctico.

2.1. Pincel

- El pintor que utiliza Pincel siempre tarda 2 horas en pintar un M^2
- El pintor que utiliza Pincel siempre gastará 4 litros por M^2
- El pintor que utiliza Pincel dará un descuento del 50 % para proyectos mayores a 40 M^2

2.2. Rodillo

- El pintor que utiliza Rodillo siempre tarda 1 hora en pintar un M^2
- El pintor que utiliza Rodillo siempre gastará 5 litros por M^2
- El pintor que utiliza Rodillo no aplica descuento para proyectos mayores a 40 M^2

2.3. Alba

- Pintura Alba siempre requiere 1 mano con pincel o 1 mano con rodillo

2.4. Venier

- Pintura Venier siempre requiere 2 manos con pincel o 1 mano con rodillo

3. Diagramas de clase

Uno o varios diagramas de clases mostrando las relaciones estáticas entre las clases. Puede agregarse todo el texto necesario para aclarar y explicar su diseño. Recuerden que la idea de todo el documento es que quede documentado y entendible cómo está implementada la solución. Todos los diagramas tienen que estar embebidos como imágenes en el informe de manera tal que entren en el ancho de una hoja A4 sin tener que rotarla. No se aceptarán diagramas en archivos sueltos.

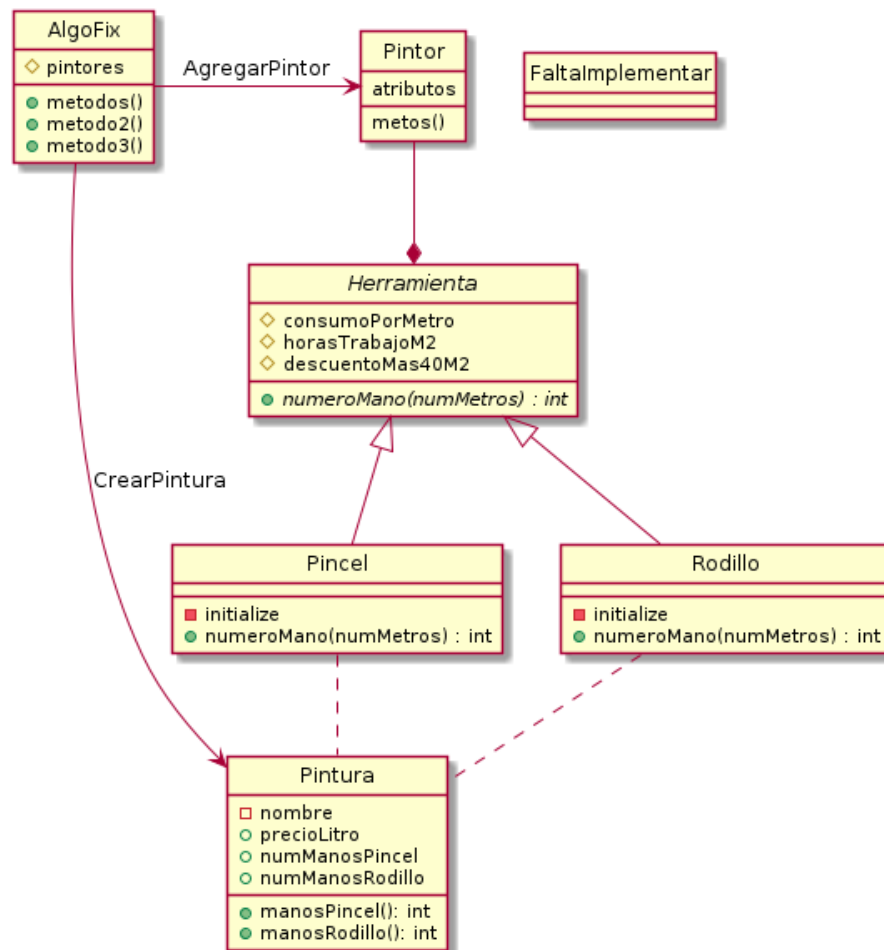


Figura 1: Prueba Diagrama de Clase.

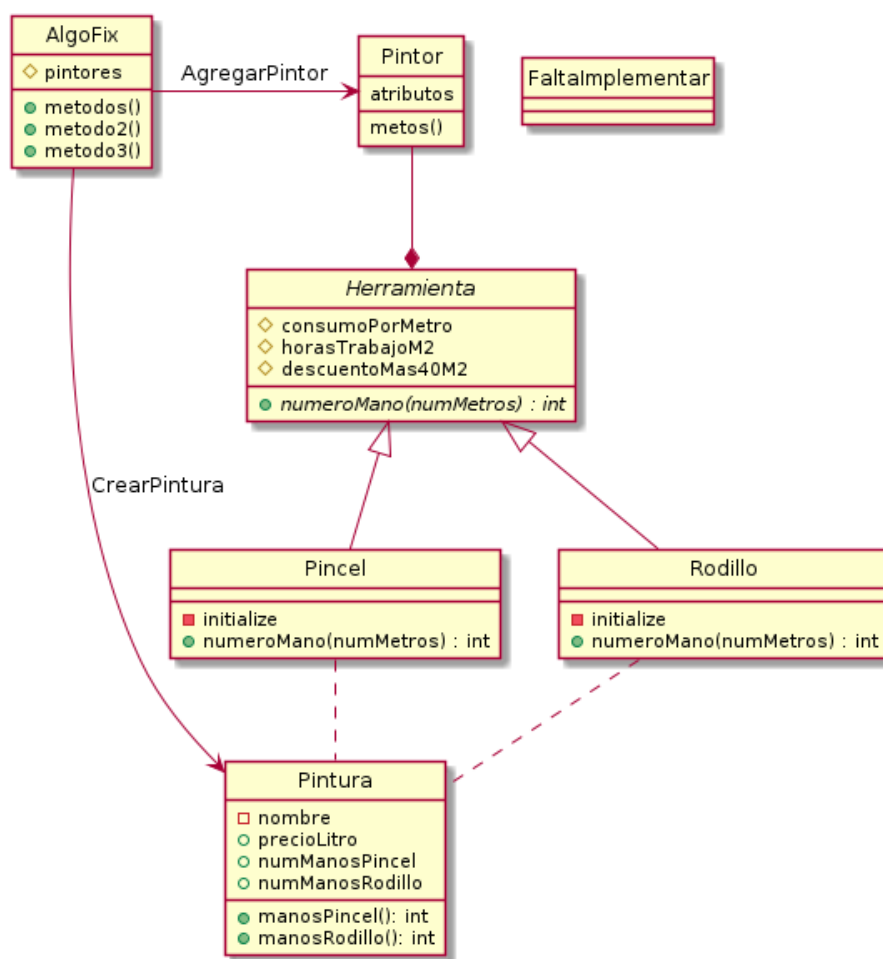


Figura 2: Diagrama de Clase 2.

4. Detalles de implementación

- Por decisión de diseño las excepciones intentarán salir al no mas crear los objetos. Es decir, si en la creacion de un Pincel o Pintura se le pasa algun valor invalido lanzará la excepcion **ValorInvalido** en el momento de su creación. Sin embargo, adicionalmente para darle mayor robustez al programa, en los casos bordes donde por alguna razon llega algun valor invalido a las operaciones esperadas, lanzara la excepcion **ValorInvalido** y asi evitar resultados absurdos.
- Se tomaran los supuestos hechos en la sección 2 como invariantes en todo momento del programa, por tanto las propiedades propias de cada objeto se asignan en su inicializacion en el momento de su creacion.

4.1. Nota de los tests

De las pruebas integradoras se extrajeron cada una de las pruebas unitarias de AlgoFixTestUnitarios de modo de simplificar el panorama y enfocar unicamente la atencion en resolver un problema a la vez. De este modo su proposito es cumplir con las pruebas por separado para que luego pasen las pruebas integradoras.

Adicionalmente se creó una clase de prueba por cada clase que se creó para resolver el trabajo práctico en las cuales se pone a prueba el comportamiento esperado del objeto en tiempo de ejecución. Por tanto las pruebas de AlgoFix están orientadas a que el objeto AlgoFix se comporte como es de esperar en tiempo de ejecución, y es distinto de las otras pruebas.

5. Excepciones

Exception ValorInvalido: Excepción lanzada en caso de pasarle un parámetro que no haga sentido para el comportamiento esperado del objeto.

Se usa al pasarle un **numMetros** inválido al Pintor o a la Herramienta, y también cuando se pasa un **numLitros** inválido a la Pintura.

6. Diagramas de secuencia

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

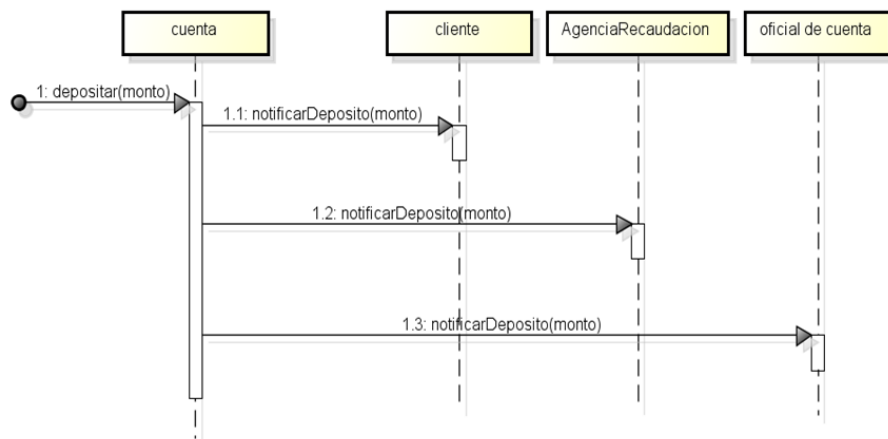
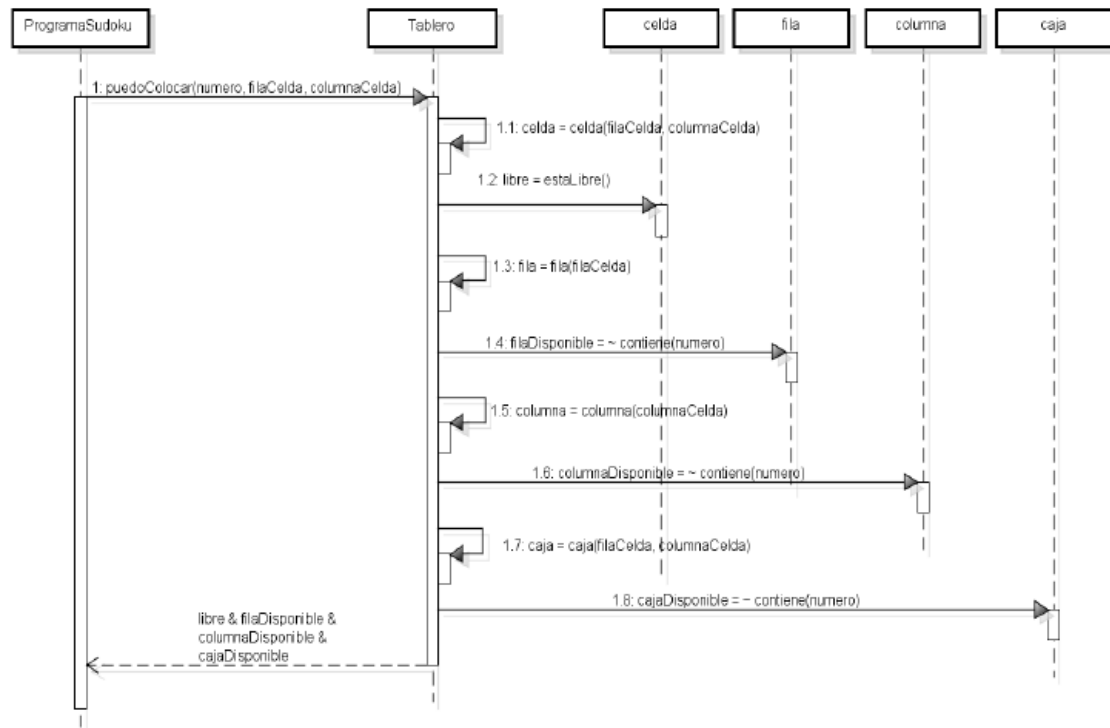


Figura 3: Comportamiento esperado de depositar.

Implementar

Figura 4: Diagrama de Secuencia de `puedoColocar()`.

Implementar