

# Trabajo Práctico 1 — Smalltalk

FIUBA | Algoritmos y Programación III [7507/9502]

Segundo cuatrimestre de 2017

## 1. Objetivo

Este es un trabajo práctico individual que tiene como objetivo aplicar los conceptos del paradigma de la orientación a objetos vistos hasta ahora en el curso mediante la resolución de un problema con Pharo.

## 2. Consigna

Se deberá desarrollar un modelo de clases de un sistema de *comercio electrónico* utilizando la metodología **TDD**. Las especificaciones son un archivo de pruebas SUnit (.st) adjunto a este enunciado y están dadas en forma de pruebas de integración. Se espera que el alumno genere una solución que permita ejecutar exitosamente las pruebas especificadas y que adicionalmente cuente con pruebas unitarias y de integración que cubran la totalidad del modelo.

## 3. Alcance

Se deberá desarrollar el modelo completo de la aplicación, sin incluir la interfaz gráfica. Para esto se especifica una API que está descrita por un conjunto de pruebas de integración. Este conjunto de pruebas está descrito por la clase dada como enunciado.

Para el manejo de errores y situaciones de excepción, se deberán desarrollar y utilizar las excepciones que se crean convenientes, más allá que en la prueba de integración provistas no se manejan excepciones.

Deberá entregarse:

- todas las clases con sus métodos, organizados en una o más categorías según criterio del alumno.
- conjunto de pruebas unitarias que muestren el uso de las clases desarrolladas y su correcto funcionamiento.
- documentación completa del diseño de clases incluyendo diagramas UML de clases y secuencia.

## 4. Informe

El informe deberá tener una carátula que incluya los nombres y apellidos del alumno, su número de padrón y dirección de correo electrónico. Además deberá tener las siguientes secciones:

### 1. Supuestos

Deberá contener explicaciones de cada uno de los supuestos que el alumno haya tenido que adoptar a partir de situaciones que no estén contempladas en la especificación.

## 2. Modelo de dominio

Explicación concisa del diseño general del trabajo.

## 3. Diagramas de clases

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.

## 4. Detalles de implementación

Explicaciones sobre la implementación interna de algunas clases que consideren que puedan llegar a resultar interesantes.

## 5. Excepciones

Explicación de cada una de las excepciones creadas y con qué fin fueron creadas.

## 6. Diagramas de secuencia

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

# 5. Forma de entrega

La solución deberá ser desarrollada en su totalidad dentro de dos paquetes:

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

Para la entrega deberá hacerse un *File Out* de estos dos paquetes (solamente estos dos, las pruebas especificadas por la cátedra **no** deben entregarse).

El entregable será un archivo comprimido en formato `.zip` que deberá llamarse `<curso>-TP1-<numero de padrón>.zip`, en donde `<curso>` es 1 para el curso de la tarde y 2 para el de la noche. Deberá contener **únicamente** los siguientes archivos:

- `TP1.st`
- `TP1-Tests.st`
- `<número de padrón>-tp1s.pdf`

Este archivo comprimido deberá ser subido al campus virtual mediante el botón “Agregar entrega” en la misma tarea donde se descargó la consigna.

**Fecha límite de entrega:** 20 de septiembre (23:55 hs.)

## 6. Criterios de corrección

### ■ Modelo

- ¿Está completo? ¿Contempla la totalidad del problema?
- ¿Respeto 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

- ¿Respeto estándares de codificación?
- ¿Está correctamente documentado?

### ■ Pruebas

- ¿Hay suficiente cantidad de pruebas? ¿Es buena la cobertura?
- ¿Los nombres de las pruebas son adecuados?
- ¿Las pruebas son realmente unitarias?

### ■ Generalidades

- ¿Son correctos los supuestos y extensiones?
- ¿Es prolija la presentación del informe?