



Práctica

Diseño de Aplicaciones Orientadas a
Objetos. Curso 2011-2012

Timoty Read, Juan M. Cigarrán

V 1.0
09/03/2012



Contenido

Enunciado	2
Primera Parte (1.5 puntos)	3
Segunda Parte (1.5 puntos)	4
Tercera Parte (1.5 puntos)	4
Integración (2 puntos)	5
Interfaz gráfica (opcional 3.5 puntos)	5
Normas y orientaciones	5
Material a Entregar	6
Temas organizativos	6
Normas para los Tutores	7
Centros Asociados vs. Prácticas en Asignaturas Obligatorias	8

Enunciado

Suponga que le han encargado diseñar e implementar el sistema informático para controlar los pedidos en todas las franquicias que **TodoPizza** tiene repartidas por toda España (i.e. la Tabla 1 muestra una lista con su ubicación). Cada una de las franquicias se caracteriza por tener a la venta pizzas de diferentes tamaños a las que se les pueden añadir diferentes ingredientes al gusto del cliente. El precio del pedido dependerá del tamaño de la pizza encargada, así como de los ingredientes utilizados. La Tabla 2 muestra los distintos tamaños de pizza con sus precios, así como la lista de ingredientes posibles con sus precios. El precio final de la pizza se podrá calcular sumando al precio de la base el precio de cada uno de los ingredientes seleccionados.

2

Madrid
Barcelona
Guadalajara
Bilbao
Tarragona

Tabla 1. Franquicias de TodoPizza

Tamaño	Precio
Pequeña	5 euros
Mediana	10 euros
Grande	15 euros
XL	20 euros
XXL	25 euros
Ingredientes	Precio
Carne	1 euro
Pimientos	0,25 euros
Bacon	0,75 euros
Aceitunas	0,50 euros
Anchoas	0,75 euros
Piña	0,25 euros
Salami	0,50 euros
Pollo	1 euro
Extra de queso	0,50 euros

Tabla 2. Lista de tamaños e ingredientes para confeccionar una pizza en TodoPizza

ESTADOS	TIEMPO ESTIMADO
Preparando ingredientes	15 seg
En el horno cocinándose	10 seg
En el proceso de corte en porciones	5 seg
Empaquetándola	2 seg

Tabla 3. Estados para la preparación de una pizza y tiempos requeridos para su realización

Después de una serie de reuniones con el cliente se ha llegado a los siguientes requisitos de implementación del sistema:

- 1- El sistema funciona de la siguiente manera. El usuario hace uso de una sencilla consola virtual en SWING que le permitirá seleccionar la tienda en la que realiza el pedido, el tamaño de la pizza, así como el número de ingredientes¹. El cliente podrá repetir un mismo ingrediente varias veces (e.g. tamaño mediano, bacon, bacon, pimientos).
- 2- Una vez confirmado el pedido, la consola deberá mostrar al usuario el precio final de su pedido, así como la configuración final de la pizza y comenzar el proceso de producción de la ésta
- 3- El proceso de producción de la pizza consta de los pasos mostrados en la Tabla 3 junto con el tiempo estimado de realización². La consola virtual deberá mostrar al usuario un pequeño display donde le informe en todo momento de cómo va el proceso de preparado de su pedido y le avise cuando éste esté empaquetado para que pase por la tienda a recogerlo.

3

Se pide realizar un diseño basado en patrones que permita implementar de manera sencilla la especificación proporcionada.

Con el fin de facilitar el diseño final, se propone al alumno abordar la práctica realizando una serie de pasos o partes que le ayudarán a identificar cada uno de los patrones involucrados en el ejercicio para, finalmente, integrar todos ellos en el sistema final.

Primera Parte (1.5 puntos)

En primer lugar abordaremos por separado la creación de las diferentes franquicias de TodoPizza, así como la abstracción del proceso de creación de objetos tipo Pizza. Para ello se propone al alumno diseñar, utilizando alguno de los patrones estudiados, la construcción de objetos de tipo Franquicia que faciliten la creación de las tiendas que TodoPizza tiene por toda España y que encapsulen la creación de un objeto genérico de tipo **Pizza** en las subclases definidas para cada una de las franquicias.

Responda a estas preguntas antes de abordar la implementación:

- 1- ¿Qué familia de patrones le resultaría más adecuada para resolver este problema? ¿por qué?
- 2- ¿Qué patrón en concreto cree que sería el más adecuado en este caso? ¿por qué?
- 3- Plantee el diagrama de clases necesario para abordar la implementación. Comience con un planteamiento genérico y luego particularícelo al problema concreto.
- 4- Implemente el diagrama de clases propuesto
- 5- Implemente una sencilla clase con un método main que ilustre el funcionamiento de esta parte de la práctica. Es decir, que permita crear una franquicia y una pizza genérica a partir de la franquicia.

¹ Para simplificar el problema, en cada pedido, el usuario podrá solicitar una única pizza.

² Los tiempos se han reducido considerablemente para poder ilustrar el funcionamiento en la práctica.

Segunda Parte (1.5 puntos)

En la primera parte nos hemos centrado en el diseño de las clases encargadas de materializar las distintas franquicias, delegando el proceso de creación de pizzas genéricas a éstas. En esta segunda parte nos vamos a centrar en los objetos pizza que podemos crear. Para ello se pide al alumno que consulte la Tabla 2, donde podrá ver que una pizza queda definida por su tamaño y por un conjunto de ingredientes que el cliente puede seleccionar en cualquier orden (i.e. incluso repitiendo alguno de ellos). Nótese como cada uno de estos elementos tiene asociado un precio contribuirá al precio final de la pizza. El precio final se calculará sumando los precios individuales de la base de la pizza y de cada uno de los ingredientes.

Olvídense por un momento de las franquicias y céntrese únicamente en las pizzas. Proponga una aproximación que, a partir de una pizza genérica, genere por composición una pizza específica con todos sus ingredientes. Tenga en cuenta que no está permitido que una pizza tenga diferentes tamaños al mismo tiempo aunque, como hemos comentado, si puede tener ingredientes repetidos.

Responda a estas preguntas antes de abordar la implementación:

- 1- ¿Qué familia de patrones le resultaría más adecuada para resolver este problema? ¿Por qué?
- 2- ¿Qué patrón en concreto utilizaría para generar el tipo de objetos deseado?
- 3- Plantee el diagrama de clases necesario para abordar la implementación. Comience con un planteamiento genérico y luego particularícelo al problema concreto.
- 4- Implemente el diagrama de clases propuesto.
- 5- Implemente una sencilla clase con un método main que ilustre la generación de diversos tipos de pizza y muestre por consola sus precios.

Tercera Parte (1.5 puntos)

En esta tercera parte de la práctica se pretende que el alumno aborde el problema de informar al cliente del estado en el que se encuentra su pedido. La Tabla 3 muestra una relación de los procesos involucrados la preparación de una pizza. Estos engloban desde su preparación inicial con el tipo de base seleccionada y los ingredientes elegidos hasta su empaquetamiento final, pasando por el proceso de cocción y cortado de la pizza. Cada uno de estos pasos tiene asociado un tiempo estimado de modo que el sistema deberá secuenciar cada una de estas acciones e ir informando al usuario cada vez que finalice una de ellas. Puede suponer una pizza genérica o el resultado que ha obtenido en el apartado anterior para abordar este problema.

Responda a estas preguntas antes de abordar la implementación:

- 1- ¿Qué familia de patrones le resultaría más adecuada para resolver este problema? ¿Por qué?
- 2- ¿Qué patrón en concreto utilizaría para monitorizar el proceso completo de preparación de la pizza?
- 3- Plantee el diagrama de clases necesario para abordar la implementación. Comience con un planteamiento genérico y luego particularícelo al problema concreto.
- 4- Implemente el diagrama de clases propuesto.

- 5- Implemente una sencilla clase con un método main que ilustre el proceso, informando del estado de creación de una pizza por consola³.

Integración (2 puntos)

Una vez realizados los tres pasos anteriores, el alumno se encontrará en disposición de abordar la integración de los patrones utilizados e implementados para resolver el problema propuesto. Los pasos que deberá realizar el alumno serán los siguientes:

- 1- Plantee un diagrama de clases que integre los diagramas propuestos en los apartados anteriores. Encapsule al máximo las funcionalidades asociadas a cada clase de objetos definidos.
- 2- Implemente la integración del código generado en los anteriores apartados de forma que mediante un método main se simule el proceso completo de entrada de datos de selección tanto de la franquicia como de la pizza y sus ingredientes y del proceso de producción de la pizza (i.e. informando al usuario al finalizar cada paso de producción).
- 3- Genere los javadocs correspondientes al sistema integrado.

5

Interfaz gráfica (opcional 3.5 puntos)

Diseñe e implemente una interfaz en SWING para la consola virtual propuesta. Para ello deberá realizar las siguientes tareas:

- 1- Dicha consola virtual deberá disponer de controles de entrada que permitan seleccionar la franquicia en la que se realiza el pedido, así como realizar la confección del pedido. Los controles de salida informarán del precio de la pizza y de su configuración final, así como del proceso de preparación de ésta una vez el cliente haya confirmado el pedido. Realice un mockup (i.e. prototipado en papel) que incluirá en el informe de la práctica justificando las decisiones tomadas acerca de los controles utilizados.
- 2- Implemente el prototipo propuesto en una interfaz SWING que deberá, además, controlar posibles errores del usuario al realizar la entrada de datos (i.e. si estos pudieran darse).
- 3- Realice 5 pedidos de prueba y pegue en el informe de la práctica los resultados obtenidos mediante un pantallazo de la consola virtual.

Normas y orientaciones

Para realizar la práctica se seguirá el siguiente método de trabajo:

- En primer lugar deberá leerse detenidamente el enunciado de esta práctica.
- Realice cada uno de los pasos propuestos y no pase al siguiente hasta no haber resuelto los anteriores. Tenga en cuenta que el apartado de integración requiere de la

³ Para simular los periodos de tiempo asociados a cada proceso puede utilizar cualquier clase Java que le proporcione la hora actual en milisegundos e implementar un bucle que le compruebe en cada iteración si se ha superado el intervalo de tiempo correspondiente.

implementación de **TODOS** los patrones propuestos en los pasos primero, segundo y tercero.

- Por cada apartado deberá responder a **TODAS** las preguntas propuestas que deberá documentar en la memoria de la práctica.
- La realización de la interfaz gráfica final es opcional. Esto quiere decir que no es obligatoria su realización para aprobar la práctica. Tenga en cuenta que el hecho de **NO** llevar a cabo este apartado conlleva obtener un **6.5 como nota máxima en la práctica**.
- Todo el código que escriba deberá estar convenientemente documentado (i.e. incluyendo JavaDocs y comentarios).
- Especifique claramente en la documentación cuales son las clases de Test para cada uno de los pasos propuestos.

Material a Entregar

Memoria: La memoria constará de los siguientes apartados:

- Portada con título “Práctica de Diseño de Aplicaciones Orientada a Objetos – Junio 2012” y los datos del alumno: Nombre, Apellidos, dirección de correo electrónico y teléfono.
- Cada sección de la memoria se corresponderá con cada uno de los apartados propuestos en este enunciado.
- Por cada sección se deberán responder a **TODAS** las preguntas propuestas incluyendo de manera clara los diagramas de clases propuestos.
- Los códigos asociados a cada sección deberán incluirse al final de la memoria en una sección de **ANEXOS** donde se identifique claramente a que sección se corresponden cada uno de los códigos.

Código: incluyendo todos los ficheros *.java y *.class de los distintos proyectos realizados en la práctica (**Un proyecto Eclipse por apartado**), así como la memoria en formato electrónico (preferiblemente html o pdf). El soporte estará libre de virus. No se corregirá ni se tendrá en cuenta ninguna práctica que esté infectada por un virus.

Una vez terminada la práctica el alumno tiene que hacer un archivo comprimido (rar o zip) de la memoria y el código y, en el apartado de “**Entrega de trabajos**”, subir una copia a la plataforma aLF según las instrucciones puestas en el curso virtual.

Temas organizativos

1. La realización de la práctica es obligatoria. Sólo se evaluará el examen si la práctica ha sido previamente aprobada.
2. La práctica deberá realizarse utilizando como **IDE Eclipse**.
3. Después de un trabajo inicial de diseño, las prácticas se hacen en los centros asociados supervisadas por el tutor de la asignatura. El trabajo a realizar con la práctica tiene dos

partes: el diseño y la implementación. Cada alumno deberá llevar su diseño el día en que se realicen las prácticas para que el tutor lo supervise y el alumno pueda realizar la implementación.

4. La práctica es individual. Las prácticas cuyo código coincida total o parcialmente con el de otro alumno serán motivo de suspenso para todos los implicados (copiadores y copiados), no pudiéndose examinar ninguno de ellos en el presente curso académico.
5. Cada tutor establecerá unas fechas para la realización de la práctica. El tutor puede organizar las sesiones que considere necesarias para la práctica pero tiene que haber **al menos una sesión presencial obligatoria**. El tutor entrará en el espacio virtual de la asignatura dentro de aLF, antes del 1 de junio, para meter las notas para sus alumnos.
6. No habrá sesión extraordinaria de prácticas ya que la asignatura ya debe estar implantada en todos los centros asociados. En caso de que algún alumno no tuviera tutor, deberá dirigirse a cualquier otro centro asociado donde se imparta la asignatura.
7. El equipo docente tendrá en cuenta prácticas con notas altas para aquellos alumnos cuyo examen esté cercano al aprobado.
8. El alumno debería dirigirse a su tutor para cualquier duda que tenga sobre su práctica y solamente al equipo docente (por correo electrónico) en el caso de que su tutor no pueda resolver su problema. En este caso pedimos al alumno que, además de sus datos personales, nos envíe el nombre del centro asociado en el que está matriculado y el de su tutor.
9. Evidentemente se puede usar los foros para realizar consultas a los compañeros pero nunca para intercambiar código.

Normas para los Tutores

Como se puede apreciar, el papel del tutor es fundamental en todos los aspectos de la práctica tanto en el planteamiento de los problemas, su desarrollo y su integración final. Tratándose de una asignatura obligatoria, cada alumno debería tener acceso a un tutor.

Los tutores deben seguir los siguientes pasos:

- Ayudar a los alumnos al principio del curso con el planteamiento de la práctica.
- Indicar a los alumnos que habrá al menos una sesión obligatoria de seguimiento de la práctica.
- Una vez terminada y entregada la práctica, el tutor tiene que entrar en el espacio virtual de la asignatura dentro de aLF, antes del 1 de junio, para meter las notas para sus estudiantes.
- Comunicar la calificación a sus alumnos.

Centros Asociados vs. Prácticas en Asignaturas Obligatorias

Las prácticas son esenciales en las titulaciones de Informática porque, entre otras cosas, permiten a los alumnos adquirir conocimientos importantes sobre los aspectos más aplicados de ciertas asignaturas, lo cual resulta de gran relevancia e interés a la hora de acceder a un puesto laboral relacionado con la Informática. Para orientar y ayudar a los alumnos, así como para comprobar que realmente un alumno ha realizado su práctica de forma satisfactoria, ésta se debe realizar en un Centro Asociado bajo la supervisión de un tutor, quien decide, en última instancia, la forma en la cual se organiza el desarrollo de la misma en su Centro Asociado (número de sesiones presenciales obligatorias, forma de entrega, etc.)

De vez en cuando sucede que un alumno se pone en contacto con un Equipo Docente del Departamento de Lenguajes y Sistemas Informáticos (L.S.I.) porque se ha matriculado en una asignatura obligatoria en un Centro Asociado que no le proporciona un tutor para supervisar la práctica, aun cuando se le ha permitido matricularse. El alumno busca en el Equipo Docente que se le proporcione una solución a este problema, como por ejemplo, la posibilidad de asistir a unas sesiones extraordinarias de prácticas en la Sede Central de la U.N.E.D. en Madrid o la posibilidad de realizar la práctica por su cuenta en casa, enviándola a continuación al Equipo Docente para su corrección. Sin embargo, los Equipos Docentes de L.S.I. no disponen de recursos para poder llevar a cabo ninguna de estas dos alternativas.

Un Centro Asociado que ha permitido a un alumno matricularse en una asignatura obligatoria de una carrera de Informática debería ayudarle a encontrar una solución al problema de la realización de las prácticas. Si se trata de una asignatura donde no se han matriculado muchos alumnos, quizás el centro no cuente con recursos para proporcionar un tutor específicamente para la asignatura. Si hay otro Centro Asociado cerca que dispone de tutor, quizás el alumno pueda realizar la práctica allí. Pero si no es así, el Centro Asociado debería proporcionar un tutor para supervisar y corregir las prácticas de sus alumnos. Lo más razonable sería que fuera un tutor de otra asignatura de Informática en el mismo Centro el que hiciera la sesión de prácticas para los alumnos de la asignatura en cuestión, y al final de la sesión evaluara los trabajos de los alumnos, según las pautas marcadas por el Equipo Docente, haciendo llegar a éste las calificaciones otorgadas.

Por lo tanto, un alumno que tras haberse matriculado en una asignatura obligatoria en un Centro Asociado, se encuentre con que el centro no tiene tutor para dicha asignatura, debería dirigirse al **Director del Centro Asociado**, para solicitar de él una solución, tal como se ha presentado aquí, es decir, alguien que pueda supervisar y corregir su práctica con plenas garantías. En el caso de que el Director no le proporcione una solución, el alumno debería comunicárselo, por escrito, lo antes posible, al Director del **Departamento de L.S.I., Dr. D. Julio Gonzalo**.