

Instituto Tecnológico de Costa Rica

Programación Orientada a Objetos

Proyecto Programado

Entregable II

Prof. Luis Pablo Soto

Estudiantes:

Fabricio Elizondo Fernández

Hans Fernández Murillo

Jose Andrés Navarro Acuña

Semestre 2, 2016.

Fecha de Entrega 28/11/2016

Introducción

En este documento se incluye una detallada explicación acerca del funcionamiento de la tercera tarea programada correspondiente al curso de programación orientada a objetos. El proyecto consistía en realizar un sistema para una Pyme que se dedica a la preparación de alimentos y elaboración de menús múltiples y variados para ofrecer a sus distintos clientes. Dicho proyecto consistía en dos partes fundamentales: Primero realizar una aplicación servidora para el caso de los administradores o encargados del sistema Pyme y segundo elaborar una aplicación remota que en este caso funcionará para los clientes del sistema. Ambas partes del proyecto presentan diversas funcionalidades, donde los usuarios finales del sistema podrán interactuar de una manera muy interesante y agradable.

Tanto la aplicación que utilizarán los administradores como los clientes cuentan con una interfaz gráfica de usuario, diferentes para cada caso, donde una vez que se inicializa el programa el usuario en curso deberá iniciar sesión con su cuenta respectiva y previamente registrada, para que el programa lo direccione a la pantalla de cliente o administrador dependiendo de su función en el sistema.

Para el caso de los clientes del sistema que desean ingresar para realizar un pedido, la aplicación “cliente” le desplegará en una tabla el menú respectivo que ofrece el sistema en el momento, con los diferentes tipos de alimentos ordenados por su categoría, donde el cliente podrá visualizarlo de acuerdo a un tamaño específico de ración y se desplegará las calorías totales respectivas. Esto es fundamental ya que uno de los objetivos principales del sistema Pyme al ofrecer este tipo de menús variados a sus clientes, es que los mismos puedan realizar combinaciones adecuadas de alimentos, acordes a la información nutricional de los platillos, para satisfacer sus necesidades de la mejor manera. Por consiguiente, la aplicación del sistema Pyme permitirá a los clientes modificar de la mejor manera su pedido, y posteriormente efectuarlo correctamente, de acuerdo a los tres tipos de pedidos ofrecidos. Además de otras funcionalidades que pueden realizar los clientes que se especificarán con más detalle posteriormente en este documento.

Por otro lado, se encuentran los administradores del sistema Pyme, quienes serán los encargados de darle un mantenimiento adecuado al catálogo de productos que ofrece el sistema. La interfaz gráfica de la aplicación de los administradores es bastante similar a la interfaz del cliente, sin embargo, varía notablemente en las funciones que realiza cada usuario. El administrador tendrá diversas responsabilidades para que el sistema funcione de la mejor manera y se pueda ofrecer la información correcta a los clientes. Entre las funciones de un administrador se encuentran: Darle mantenimiento al catálogo, ya sea agregando, eliminando o modificando algún producto en específico, además de poder establecer el producto como visible o no visible a los clientes en general, por otro lado, el administrador podrá conocer en tiempo real todos los pedidos que se generen y consultarlos de diferentes formas de acuerdo a la necesidad del momento, entre otras funciones.

Debido a que la aplicación se desarrolló en un ambiente distribuido entre un cliente y un servidor, este sistema se implementó por medio de sockets los cuales van a permitir la comunicación de varias sesiones en simultáneo a un servidor central que permitirá almacenar en tiempo real la información que se genere ya sea por parte de clientes como de administradores del sistema. La parte del programa que funciona como un servidor central, genera una bitácora general de las diferentes actividades que se realizan en el sistema con su respectiva fecha y hora.

Como parte del proceso de investigación de esta tarea programada, se logró implementar algunas funcionalidades específicas como la carga y lectura de archivos XML, el uso de sockets para el envío de información entre un cliente y un servidor, la incorporación de una interfaz gráfica de usuario, el desarrollo por medio de la arquitectura de capas siguiendo el modelo MVC, el patrón estructural DTO, entre otras.

A continuación, se presenta un índice que incluye los distintos segmentos sobre la documentación presentada:

Índice

* Información del contexto del problema
* Metodología del trabajo
* Cronograma de trabajo
* Diagrama de clases en UML
* Justificación de los patrones utilizados dentro del modelo
* Análisis de resultados
* Enlace al JavaDoc
* Aspectos relevantes y lecciones aprendidas
* Bitácora de trabajo
* Bibliografía y fuentes digitales utilizadas
* Anexos
* Estatus de la entrega
* Información del contexto del Problema

**Especificación del II y III Proyecto**

El presente documento pretende definir los lineamientos de trabajo correspondiente a los proyectos II y III del curso Programación Orientada a Objetos, los cuales comprenden un contexto específico que se detalla en este documento y una serie de actividades que deberán ser realizadas en los equipos de trabajo que se conformen.

**Objetivos para el II Proyecto**

● Elaborar un documento básico de especificación de requerimientos a través de un diagrama de casos de uso, que refleje una propuesta de comportamiento de un posible sistema ante la situación en contexto expuesta.

● Elaborar diagramas de actividad, secuencia y estado que complementen la propuesta de solución descrita en los casos de uso definidos.

● Explorar una herramienta de prototipado de aplicaciones para construir un prototipo asociado a la propuesta documental.

**Objetivos para el III Proyecto**

● Elaborar una propuesta de diseño de software que satisfaga los requerimientos establecidos y confirmados en la primera fase, aplicando el proceso básico completo de la Ingeniería de Software, para crear una propuesta de solución a la luz de los pilares fundamentales de la orientación a objetos: abstracción, encapsulamiento, herencia y polimorfismo.

● Poner en práctica las características de las clases abstractas, métodos abstractos, interfaces y excepciones.

● Incorporar patrones de diseño estudiados en clase (patrón creacional, adapter, strategy)

● Fomentar y desarrollar habilidades de investigación de forma individual por parte del estudiante, en los temas de:

○ Aplicación cliente-servidor y el uso de sockets.

○ Construcción y consumo de XML.

● Fomentar el desarrollo de las habilidades blandas centrándose en el trabajo en equipo.

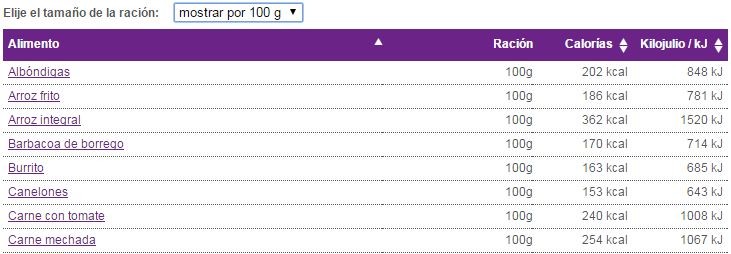
● Desarrollar habilidades de comunicación oral y escrita dentro del equipo de trabajo.

● Construir la aplicación en un ambiente de programación haciendo uso del lenguaje de programación Java.

**El contexto**

Una Pyme que se dedica a la preparación de alimentos ha decidido ofrecer a sus clientes y público en general el servicio de elaboración de menús múltiples y variados, aportando información nutricional de cada plato para permitir combinaciones de alimentos adecuadas a las propias necesidades.

Es por esto que ha definido un menú de alimentos en los cuales, se indica la cantidad de calorías que aporta en una porción como se muestra en la siguiente tabla:





Aunque en el establecimiento se pueden preparar gran cantidad de platillos, el administrador puede habilitar o definir los que estarán disponibles en un momento determinado del tiempo, de igual forma puede dar mantenimiento al catálogo, agregando nuevos platos, eliminando algunos que decida no ofrecerá más al público o modificando alguna información asociada.

Aplicación servidora - administradora

Por esta razón desea tener una aplicación en la que pueda dar gestión tanto al catálogo de productos que se pueden ofrecer en el nuevo establecimiento, así como dar seguimiento y atención a los pedidos de los clientes que visiten el local o que hagan sus pedidos en forma remota.

Con respecto a los productos del catálogo, se puede indicar que cada plato tiene un identificador que contiene una clave alfanúmerica de máximo tres letras seguido de un guión y un número de 3 dígitos.

Así, si el plato es una entrada el código inicia con *ENT*, si es un comida principal el código inicia con *PRN* o si es postre inicia con PTR. Se ofrecen bebidas naturales y gaseosas cuyo código es BEB.

Adicionalmente se conoce el nombre del plato, una descripción y la representación de la medida de la ración recomendada con su respectivo valor en calorías.

Opcionalmente se puede mantener una imagen de referencia del platillo.

La ración puede contener una cierta cantidad de piezas por lo que también se puede manejar la posibilidad de conocer la cantidad de calorías por la cantidad de piezas que se incluyen dentro de la ración.

De esta forma, por ejemplo, un burrito puede tener la siguiente información:

|  |  |  |
| --- | --- | --- |
| **Código** | PRN-008 |  |
| **Nombre** | Burrito |
| **Descripción** | tortilla de trigo rellena de frijoles molidos, trozos de pollo y queso mozarella, cubierta con lechuga, tomate, aguacate y natilla |
| **Tamaño de la porción** | 400 gramos |
| **Piezas por porción** | 2 |
| **Calorías en 1 porción** | 326 kcal |
| **Calorías por pieza** | 163 kcal |
| **Precio** | ¢ 1500 |

Cuando el administrador inicia sus labores, puede visualizar todo el catálogo de productos que puede ofrecer, pero también puede filtrar esta información por tipo de platillo (ENT, PRN, PTR, BEB) o por la posibilidad de estar disponibles al público (visibles, no visibles, ambos).

En todo momento el administrador también puede ver la lista de pedidos que se han generado por parte de los clientes.

En esta primera etapa, se pretende únicamente el obtener una información particular del tipo de pedidos que se reciben. Posteriormente se desea dar atención a los pedidos recibidos y gestionar el pago.

Al administrador le interesa de manera particular conocer:

● La lista de los 10 productos que se solicitan con más regularidad indistintamente su categoría (TOP-TEN).

En este listado se muestra código, nombre, descripción y categoría ordenada descendentemente.

● Los productos que no se han pedido NUNCA. En este caso se muestra código, nombre y categoría ordenado por categoría y por código de producto.

● Relación porcentual entre pedidos en sitio, para recoger y express. En este caso del total de pedidos recibido se obtiene la relación porcentual por tipo de pedido y se muestra una tabla que muestre los valores de dicha distribución con respecto al total general.

Aplicación remota - cliente

Cualquier cliente que tenga acceso a la aplicación de la empresa puede realizar un pedido de comida a través de interactuar con una pantalla que le permite crear una lista de comidas que desea obtener.

Estos pedidos pueden ser en sitio, para pasar a recoger o express.

Cualquiera de los pedidos registra una fecha que es la del día, un número consecutivo global y el nombre de la persona que hace el pedido.

Si el cliente visita el restaurante puede gestionar por auto-solicitud su pedido pues cada mesa cuenta con una tableta desde la que puede realizar su pedido.

Si el pedido es solicitado para pasar a recoger, interesa registrar el nombre de la persona que va a pasar a buscar el pedido (puede ser distinta de quien hace el pedido) y su número de celular. En cuanto al costo hay un incremento en un porcentaje del costo total del pedido por concepto de empaque, cuyo valor es definido por el administrador como un parámetro externo de configuración de la aplicación servidora.

Finalmente, si el pedido es express, se solicita el número de celular de la persona que hace el pedido pero además el detalle de la dirección exacta donde debe ir a dejarse el pedido. En este último caso además hay un rubro por transporte que se cobra adicional que es un monto fijo establecido por el administrador como un parámetro externo de configuración de la aplicación servidora.

La pantalla remota o del cliente se organiza por los grupos de alimentos (ENTRADAS, PLATOS FUERTES, POSTRES, BEBIDAS) y el usuario puede desplegar la lista de los alimentos que desea solicitar, seleccionando cualquiera de los grupos de alimentos.

De cada producto puede solicitar uno o más porciones. Puede eliminar algún alimento si no lo desea, pero no puede quedar uno en su pedido con cero porciones.

Una vez que esté listo y decida enviar a procesar el pedido, se completan los datos necesarios para registrar el pedido en la aplicación dando a conocer al cliente los detalles de costo del mismo. Recuerde que en esta primera etapa no hay mecanismo de cancelación.

Cada vez que el usuario selecciona un producto se incrementa el total de calorías acumuladas en su pedido así como la cantidad de dinero que debe cancelar. De igual forma estos datos disminuyen si se altera la cantidad de porciones o se eliminan productos del pedido.

En esta primera etapa, al enviar el pedido se notifica el costo total de la orden y aún no hay opción de pago automático, por lo que el usuario debe cancelar en sitio o bien cuando le llegue el pedido express con el motorizado encargado de entregar la encomienda.

Consideraciones técnicas

La aplicación debe ser construida haciendo uso del lenguaje de programación Java y con una interfaz gráfica de usuario desarrollada en Swing o Fx según decisión del equipo de trabajo.

La aplicación debe ser desarrollada en una ambiente distribuido cliente-servidor, para lo cual deberá usar sockets para permitir la comunicación de varias sesiones en simultáneo a un servidor central.

El catálogo de alimentos se mantiene almacenado en un archivo con formato XML. La aplicación servidora debe permitir el monitoreo de las conexiones (bitácora de

comportamiento) así como las acciones que se llevan a cabo con el catálogo y las

consultas derivadas de los pedidos. También puede ver el historial de los pedidos recibidos, refrescando cada vez que un nuevo pedido ingresa.

Metodología del Trabajo

Para realizar esta tarea, se utilizó el lenguaje de programación Java, de acuerdo a lo solicitado en las indicaciones de la tarea. Para abordar el problema del modelo de la solución se decidió como grupo, reunirnos en los momentos en que se podía, y debido a las circunstancias, la mayor parte del trabajo se coordinó por medio de mensajes.

Se repartieron tareas (las cuáles se especifican en la bitácora y el cronograma de trabajo) de investigación como en el caso de la carga de los archivos XML para poder convertir objetos a partir de dicha información, el uso de Java Fx para el caso de las interfaces gráficas de usuario implementadas en el proyecto(tanto la del cliente como la del administrador), la utilización de sockets para permitir el manejo de sesiones simultáneas a una estación servidora central, el seguimiento del modelo MVC por capas para establecer las responsabilidades entre cada capa del programa, el patrón estructural DTO para permitir el envío de objetos y clases entre el servidor y el cliente, y el modelado general de las distintas partes del problema presentado.

Cronograma de Trabajo

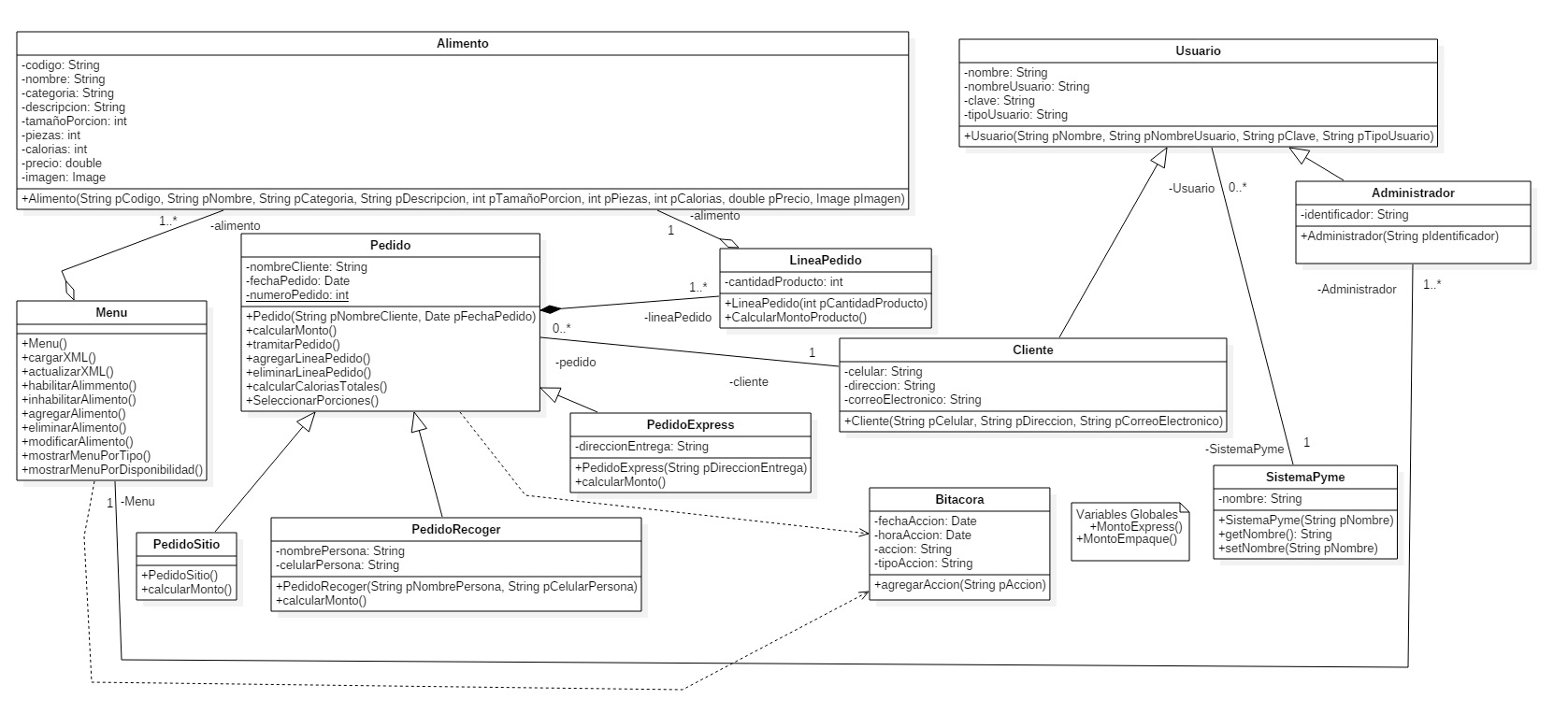
17 de Noviembre: terminar la interfaz gráfica de la aplicación. También investigar la implementación de sockets a una aplicación y guardar y cargar objetos a partir de un archivo XML.

22 de Noviembre: haber completado la parte lógica del servidor o del cliente.

25 de Noviembre: haber completado ambas partes lógicas del programa.

27 de Noviembre: terminar la documentación y corrección o diseño de los faltantes diagramas UML.

Diagrama de clases en UML



Justificación de los patrones utilizados

En esta tarea se utilizó el uso del patrón de diseño MVC, que es el modelo, vista y controlador, que se puede contemplar en la modularidad y en la forma en que están organizados los distintos paquetes con sus respectivas clases dentro del programa. El implementar este diseño era parte de los requerimientos de la especificación del proyecto, y era bastante conveniente su utilización, ya que permite trabajar el programa de una manera más organizada y darles una visualización ordenada a las distintas capas de trabajo incluidas en la tarea. Además, este patrón permite que las responsabilidades de las diferentes partes del programa, sean totalmente separadas, y que, si por alguna razón se tienen que agregar o eliminar partes del programa en alguna capa específica, puedan ser efectuados dichos cambios sin afectar a las otras capas.

Por otro lado, se empleó el patrón estructural DTO: Data Transfer Object. La razón de la incorporación de este patrón de diseño es que permite la transferencia de clases y objetos entre el servidor y el cliente, sin la necesidad, de tener que pasar los parámetros o atributos de uno en uno para la posterior transformación en objetos, lo que se vuelve bastante tedioso y complejo a la hora de programar, además, se complicaría mucho darle mantenimiento o hacerles modificaciones a los parámetros de uno en uno.

Análisis de resultados

En conclusión por análisis del grupo, se puede decir que la aplicación está con todas sus funcionalidades implementadas y trabaja como se dice en la especificación del proyecto. Los sockets y carga de archivos mediante XML funcionan de manera excelente, mencionando que su uso fue aprendido gracias a la realización de este proyecto programado, al igual que el diseño de diferentes tipos de diagramas UML. También se adquirió bastante conocimiento de cómo funcionan en realidad modelos de aplicaciones que se utilizan día a día en dispositivos que la mayoría de las personas utilizan y a cómo implementarlos en una aplicación propia.

Enlace al JavaDoc

Aspectos relevantes y lecciones aprendidas

Entre las lecciones aprendidas durante este trabajo, cabe rescatar como punto fundamental, en el caso de las habilidades blandas, el trabajo en equipo, ya que tanto en el transcurso de nuestra carrera como en la vida laboral es de suma importancia aprender a trabajar en equipo. Y es que, en muchas ocasiones en la vida, se nos presentan retos o situaciones donde por más preparados que se encuentre uno como persona, siempre es necesaria la ayuda de otras personas, además que esto permite reducir el tiempo de trabajo individual, facilita la comunicación y permite mejorar la eficiencia del proceso en todos los aspectos.

Por otro lado, entre los aspectos más técnicos de la tarea, se mejoró mucho en el manejo del lenguaje de programación de Java por medio de la práctica constante y el uso de diversas librerías, y sobre el paradigma de la programación orientada a objetos, lo cual es beneficioso tanto para este curso como para el resto de la carrera.

A manera de concluir, otra de las lecciones aprendidas es la mejora en el proceso de investigación, ya que todos los miembros del grupo, como parte de las tareas asignadas, en uno u otro momento se tuvo que investigar diversas fuentes tanto digitales como consultas a personas externas que manejaban un dominio del tema.

Bitácora de trabajo

13 de Noviembre: se termina el desarrollo de la interfaz gráfica.

20 de Noviembre: se termina el desarrollo de la parte lógica de cliente y administrador. Además se lleva la parte de servidor bastante desarrollada.

23 de Noviembre: se junta la interfaz gráfica con la parte lógica de la aplicación y se hacen varias pruebas. La parte de desarrollo de la aplicación culmina.

27 de Noviembre: se termina de desarrollar la documentación y diagramas relacionados a la aplicación. Se termina el desarrollo del proyecto.

Bibliografía y fuentes digitales utilizadas

Librería para parsear el XML

<https://www.tutorialspoint.com/java_xml/java_jdom_parse_document.htm>

Diálogos en JavaFX

<http://code.makery.ch/blog/javafx-dialogs-official/>

Excepciones en Java

<https://docs.oracle.com/javase/7/docs/api/java/lang/Exception.html>

<https://www.tutorialspoint.com/java/java_exceptions.htm>

Serialización en Java

<http://www.javapractices.com/topic/TopicAction.do?Id=45>

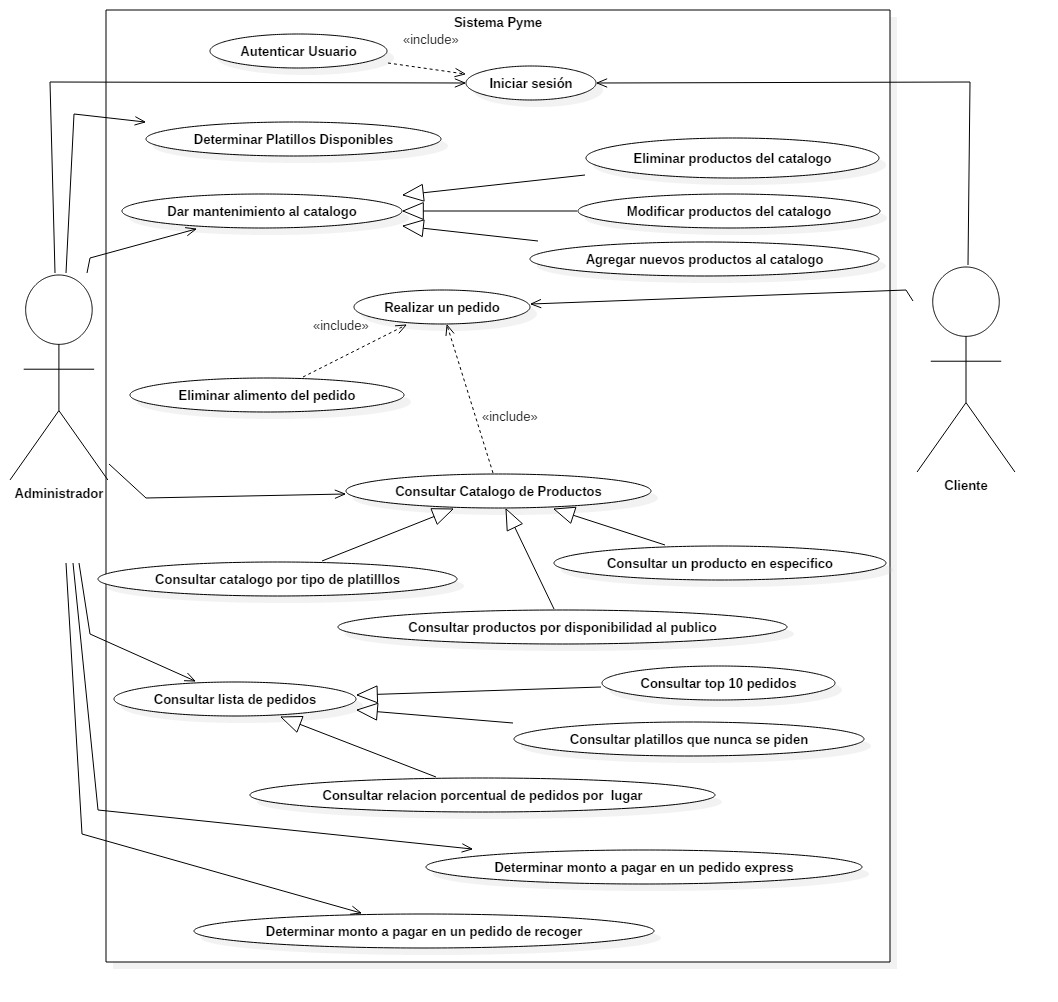
<http://www.journaldev.com/2452/serialization-in-java>

Tutorial para el manejo del patrón MVC

<http://desarrollo-java.readthedocs.io/es/latest/tutorial5.md.html>

Anexos

* Diagrama General de Casos de Usos



* Casos de Uso Expandidos

**Casos de uso expandidos**

|  |
| --- |
| **Nombre del caso de uso**  Realizar un pedido |
| **Descripción**  El caso de uso inicia cuando el cliente ingresa al sistema para poder realizar un pedido. El cliente en curso podrá seleccionar los alimentos que desee para su pedido e irlos agregando a una lista donde se incluirán dichos alimentos. El cliente deberá indicar la cantidad de porciones por alimento que desee. Una vez finalizada la selección, el sistema re direccionará al cliente a una ventana donde deberá seleccionar la opción de acuerdo al tipo de pedido que desea realizar. El sistema brindara las herramientas necesarias para que el cliente ingrese los datos requeridos de acuerdo al tipo de pedido seleccionado. El sistema valida los datos y guarda la información requerida. El caso de uso concluye cuando el cliente logro realizar el pedido satisfactoriamente y el sistema lo tramito correctamente, guardando la información necesaria. |
| **Actores**  Cliente |
| **Precondiciones**   * El cliente debe haber ingresado correctamente al sistema. |
| **Flujo normal**   1. El sistema muestra el catálogo de productos al cliente. 2. El cliente selecciona los alimentos que desea para su pedido. 3. El cliente selecciona las porciones respectivas que desea para cada alimento. 4. El sistema muestra las calorías totales de los pedidos seleccionados. 5. El sistema solicita al cliente seleccionar el tipo de pedido que desee. 6. El cliente selecciona el tipo de pedido. 7. El cliente ingresa los datos. 8. El sistema valida los datos. 9. Los datos son válidos. 10. El sistema muestra el monto total a pagar. 11. El sistema solicita una confirmación del pedido. 12. El cliente acepta la confirmación. 13. El pedido se tramita. 14. Los datos son guardados correctamente. |
| **Flujo alternativo**  1.1 El catálogo de productos no puede ser mostrado ya que todavía el sistema no ha cargado o no pudo cargar satisfactoriamente el archivo con los productos. Se indica un mensaje que dice “El catálogo de productos no puede ser mostrado en este momento, por favor verifique con la persona encargada que el archivo con los productos haya sido cargado satisfactoriamente”. Termina el caso de uso.  9.1 Los datos ingresados presentan errores de sintaxis por lo tanto son incorrectos. Se muestra un mensaje que dice “Los datos ingresados son incorrectos, por favor vuelva a ingresar los datos”. Se espera la nueva acción del cliente que desea ingresar.  12.1 El cliente no acepta la confirmación por alguna equivocación en el proceso. El sistema imprime en la pantalla lo siguiente “No se aceptó la confirmación, por lo tanto, no se prosigue con el pedido”. El sistema re direcciona al paso 1 del flujo normal. |
| **Postcondiciones**   * El cliente logra realizar el pedido satisfactoriamente * Se imprime en pantalla un mensaje que dice “¡Se ha efectuado el pedido exitosamente!”. * Los datos del pedido son guardados satisfactoriamente. |

|  |
| --- |
| **Nombre del caso de uso**  Autenticar Usuario (Nota: debido a lo solicitado en el documento, este caso de uso expandido se realiza específicamente sobre la autenticación de un administrador, aunque en el diagrama general de casos de uso, autenticar usuario permite validar tanto a administradores como a clientes). |
| **Descripción**  El caso de uso inicia cuando el administrador en curso desea ingresar al sistema y el mismo tiene que validarle los datos para verificar que el usuario se trate de un administrador registrado previamente. El caso de uso concluye cuando los datos son validados y el administrador logra ingresar satisfactoriamente al sistema. |
| **Actores**  Administrador |
| **Precondiciones**   * El administrador debe estar previamente registrado en el sistema. * El administrador debe haber ingresado al caso de uso “Iniciar sesión”. |
| **Flujo normal**   1. Los datos que ingresó el administrador son validados por el sistema. 2. Los datos son correctos. 3. El sistema permite el ingreso. |
| **Flujo alternativo**  2.1 Los datos ingresados presentan errores de sintaxis por lo tanto son incorrectos. Se muestra un mensaje que dice “Los datos ingresados son incorrectos, por favor vuelva a ingresar los datos”. El sistema se devuelve al caso de uso iniciar sesión y se espera la nueva acción del administrador que desea ingresar.  2.2 Los datos ingresados no corresponden a los datos de un administrador válido. Se indica “Los datos ingresados no corresponden a ningún administrador del sistema”. El sistema termina este caso de uso y redirige al caso de uso de iniciar sesión con la información correspondiente. |
| **Postcondiciones**   * El administrador logra ingresar al sistema. * Se envía un mensaje de bienvenida. * Se muestran las opciones del sistema al administrador. |

|  |
| --- |
| **Nombre del caso de uso**  Consultar catálogo de productos |
| **Descripción**  El caso de uso inicia cuando el administrador desea consultar el catálogo de productos que ofrece el sistema para algún propósito en particular. El administrador en curso deberá seleccionar la opción del sistema para consultar el catálogo de los productos. El sistema desplegará dicho catálogo, mostrando los productos en general con sus datos respectivos por piezas. El caso de uso concluye cuando el sistema muestra el catálogo al administrador. |
| **Actores**  Administrador |
| **Precondiciones**   * El administrador debe haber ingresado al sistema. * El administrador debe haber seleccionado la opción del sistema para ingresar al caso de uso. |
| **Flujo normal**   1. El sistema muestra el catálogo de productos al usuario. |
| **Flujo alternativo**  1.1 El catálogo de productos no puede ser mostrado ya que todavía el sistema no ha cargado o no pudo cargar satisfactoriamente el archivo con los productos. Se indica un mensaje que dice “El catálogo de productos no puede ser mostrado en este momento, por favor verifique con la persona encargada que el archivo con los productos haya sido cargado satisfactoriamente”. Termina el caso de uso. |
| **Postcondiciones**   * El usuario logra observar el catálogo de los distintos productos alimenticios que ofrece el sistema. |

|  |
| --- |
| **Nombre del caso de uso**  Eliminar alimento del pedido |
| **Descripción**  El caso de uso inicia cuando el cliente del sistema desea eliminar algún producto o alimento, del pedido que está realizando en un momento determinado. El sistema brinda las herramientas necesarias para que el cliente seleccione los productos que desea eliminar del pedido. El cliente selecciona dichos productos. El sistema solicita una confirmación del proceso para validar que todo este correcto. El caso de uso concluye cuando el cliente confirmó que la selección de los productos era correcta y el sistema eliminó dichos productos del pedido en curso. |
| **Actores**  Cliente |
| **Precondiciones**   * El cliente debe haber ingresado correctamente al sistema. * El cliente debe haber ingresado al caso de uso “Realizar un pedido”. |
| **Flujo normal**   1. El cliente selecciona los productos. 2. El sistema solicita confirmación. 3. El cliente acepta la confirmación. 4. El sistema elimina los productos del pedido. |
| **Flujo alternativo**  3.1 El cliente no acepta la confirmación por alguna equivocación en el proceso. El sistema imprime en la pantalla lo siguiente “No se aceptó la confirmación, por lo tanto, no se prosigue con la eliminación”. El sistema re direcciona al paso 1 del flujo normal. |
| **Postcondiciones**   * El cliente eliminó los productos deseados del pedido. |

|  |
| --- |
| **Nombre del caso de uso**  Determinar monto a pagar en un pedido express |
| **Descripción**  El caso de uso inicia cuando el administrador del sistema desea determinar cuál va a ser el monto a pagar por los clientes cuando realicen un pedido express. El sistema brinda las herramientas necesarias para que el administrador implemente o ingrese cual va a ser el monto seleccionado. El administrador ingresa el monto y posteriormente el sistema solicitara una confirmación del proceso. El sistema realiza una validación de datos de entrada. El caso de uso concluye cuando el administrador acepta la confirmación, el monto fue agregado y se actualiza la información en el sistema. |
| **Actores**  Administrador |
| **Precondiciones**   * El administrador debe haber ingresado correctamente al sistema. * El administrador debe haber seleccionado la opción del sistema para ingresar al caso de uso. |
| **Flujo normal**   1. El sistema le brinda las herramientas al administrador para que ingrese el monto. 2. El administrador ingresa el monto. 3. Se valida el monto ingresado por el administrador. 4. El monto es válido. 5. El sistema solicita confirmación del proceso. 6. El administrador acepta la confirmación. 7. Se guarda la información ingresada en el sistema. |
| **Flujo alternativo**  4.1 Los datos ingresados presentan errores de sintaxis por lo tanto son incorrectos. Se muestra un mensaje que dice “Los datos ingresados son incorrectos, por favor vuelva a ingresar los datos”. Se espera la nueva acción del administrador.  6.1 El administrador no acepta la confirmación por alguna equivocación en el proceso. El sistema imprime en la pantalla lo siguiente “No se aceptó la confirmación de los cambios por lo tanto no se efectúa la actualización del estado de los productos”. El sistema re direcciona al paso 1 del flujo normal. |
| **Postcondiciones**   * El administrador determino el monto a pagar en un pedido express.   - Se muestra un mensaje en la pantalla que dice “¡El monto a pagar en un pedido express fue agregado satisfactoriamente!”. |

Estatus de la entrega

Se considera como grupo que el estatus de entrega de esta tarea es excelente.