
	<b>Principios y Fundamentos de la Ingeniería del Software</b>	
	Examen convocatoria de septiembre 2013	2º Grado en Ing. Informática 3/9/2012

**Problema Diseño (60% nota).** GesBarTolo, Gestión de Bares y Cafeterías

**Duración: 2 horas**



Se desea modelar el software que facilite el funcionamiento de la cadena de Bares 'Tolo'.

Cada Bar se encuentra formado por una barra, donde se dispensan bebidas y cafés, la cocina, donde se dispensa la bollería, sandwiches y demás aperitivos y un gran salón dividido en mesas, numeradas.

Los usuarios del sistema serán, por un lado, el **camarero** (hay varias unidades de este *recurso*), encargado de atender las mesas y de servir desde la barra, y por otro lado, el **cocinero**, responsable de los platos y aperitivos.

El **escenario habitual** de uso del sistema es el siguiente:

El camarero observa que una mesa vacía pasa a estar ocupada, por lo que se dirige a ella. Con una PDA ejecuta la funcionalidad de *nuevo servicio*, donde indica el número de la mesa y pasa a tomar nota de todos los productos que solicitan, tanto de las bebidas como de las comidas. Una funcionalidad gráfica de *selección de mesa* debe facilitar la identificación de la mesa.

Una vez acaba este tratamiento, el sistema discrimina los productos, y desde equipos sitos en la barra muestran continuamente las *bebidas por servir* y en un equipo en la cocina, las *comidas por servir*.

Algún camarero libre preparará las bebidas solicitadas, las ubicará en la zona reservada de la barra y marcará en el equipo el pedido como *listo*. Del mismo modo, los cocineros también tienen la posibilidad de marcar su pedido como *listo*, cuando haya terminado de preparar todas las peticiones de una mesa.

A partir de ese momento, cualquier camarero, a través de la *funcionalidad bebidas y comidas listas* (que se visualiza continuamente en la barra en una pantalla táctil) podrá comprobar que artículos están listos y a que mesa han de servirse, lo cual hará a la mayor brevedad. Una vez realizado este trabajo, marca el pedido como *servido* con la funcionalidad correspondiente.

Los clientes pueden solicitar nuevos productos, para lo que los camareros deben disponer de la opción de *añadir producto* a un servicio, lo que implica que primeramente seleccionen la mesa a fin de acumular en una sola cuenta todos los servicios solicitados por un cliente. En este momento se ayudarán de la funcionalidad previamente referida de *selección de mesa*.

Finalmente, cuando el cliente lo solicita, se *cobrará servicio*, tras seleccionar nuevamente la mesa. Se le imprime un ticket con el total de las consumiciones, que se entrega y cobra.

El sistema no controlará qué camarero o cocinero realiza o atiende cuál pedido, sólo la gestión del proceso de servicios y mesas. Tampoco se identificará al cliente, se trata de manera genérica.

Se le pide que analice, diseñe e implemente los siguientes entregables:

1. **Diagrama de Casos de Uso** (2 puntos)
2. **Especificación textual del CU “Nuevo Servicio”** usando la plantilla adjunta (1 punto)
3. **Diagrama de Clases** (2 puntos)
4. **Diagrama de Secuencia para el CU “Nuevo Servicio”** (1 punto)

### Problema Estimación (20% nota). Añadiendo un nuevo cliente

Las dos siguientes pantallas se usan para añadir nuevos clientes a una aplicación. El cliente no se considera añadido hasta que ambos, información de la dirección e información adicional, se completan.

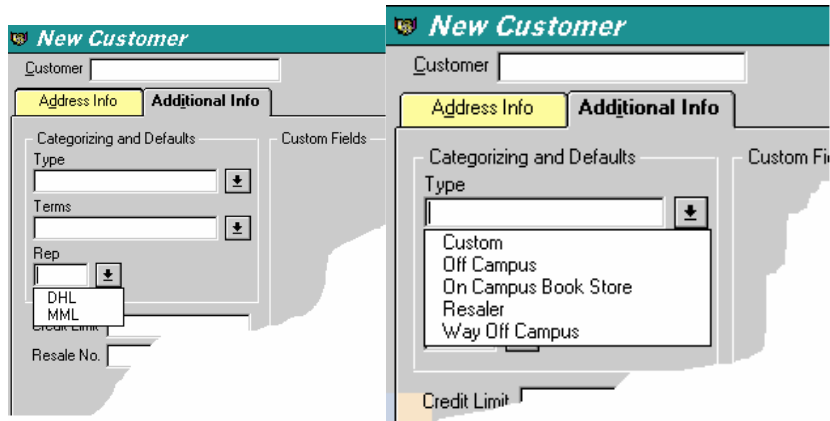
Los botones *OK* y *Next* (Siguiente) guardan ambos información al fichero.

The screenshot shows the 'New Customer' dialog box with the 'Address Info' tab selected. The 'Customer' field is at the top. Below it, the 'Address Info' tab contains fields for 'Company Name', 'Contact', 'Mr./Ms./...', 'Phone', 'First Name', 'M.I.', 'FAK', 'Last Name', 'Alt. Ph.', and 'Alt Contact'. At the bottom, there are two large text areas for 'Bill To' and 'Ship To', with a '>> Copy >>' button between them. On the right side, there are three buttons: 'OK' (with a green checkmark), 'Cancel' (with a red X), and 'Next' (with green arrows).

Hay cuatro listas desplegables, en la pestaña “*Additional info*—Información adicional” (*Type*-Tipo de cliente, *Terms*-tipo de Contrato, *Rep*-Iniciales del Representante y *Tax item*-Partida impositiva). Las tres primeras se leen desde ficheros mantenidos por la aplicación. *Tax item* es una tabla codificada manualmente. Debe ignorar el botón “Define fields”.

The screenshot shows the 'New Customer' dialog box with the 'Additional Info' tab selected. The 'Customer' field is at the top. Below it, the 'Additional Info' tab contains a 'Categorizing and Defaults' section with dropdown menus for 'Type', 'Terms', and 'Rep'. There is a 'Credit Limit' field. Below that, the 'Sales Tax Information' section has a 'Tax Item' dropdown (showing 'Out of State'), a checked 'Customer is taxable' checkbox, and a 'Resale Number' field. To the right of these is a 'Custom Fields' section with a 'Define Fields' button. At the bottom right, there is an 'Opening Balance' section with a value of '0.00' and a date 'as of 05/19/97'. On the right side, there are three buttons: 'OK' (with a green checkmark), 'Cancel' (with a red X), and 'Next' (with green arrows).

Las listas desplegables se muestran a continuación.



Para este fragmento de la aplicación, responda a las siguientes preguntas:

1. ¿Cuántas Entradas Externas hay? Diga cuáles son

2. En términos de punto-función, ¿qué son *Type* y *Terms*?

3. En términos de punto-función, ¿cómo se trata a *Type* y *Terms* la segunda vez que aparecen, después de haber sido seleccionados en la primera?

Plantilla para la especificación textual del CU “Nuevo Servicio”

UC-01		
Descripción		
Precondición		
Secuencia Normal	Paso	Acción
Postcondición		
Excepciones	Paso	Acción

## Test (20%).

**Conteste V (verdadero) o F (falso) a las siguientes preguntas.** (Cada respuesta correcta vale +0.1 puntos, cada respuesta incorrecta vale -0.033 puntos)

Uno de los factores de calidad de McCall es la <b>portabilidad</b> del software que hace referencia al grado en el que un programa (o partes de un programa) pueden volverse a utilizar en otras aplicaciones.	
<b>Corrección</b> se define como el grado en el que un programa satisface sus especificaciones y cumple con los objetivos del desarrollador.	
La <b>facilidad de uso</b> es un factor de calidad que se define como el esfuerzo que se requiere para aprender, operar, preparar las entradas e interpretar las salidas de un programa.	
La ley de <b>Meskimen</b> viene a decirnos que si dedicas suficiente tiempo a diseñar bien tu software, ahorrarás tiempo en la fase de desarrollo.	
Crear software <b>“suficientemente bueno”</b> , es decir distribuirlo con errores detectados pero reconociendo que algunas funciones y características se mejorarán en las siguientes versiones, es una práctica habitual, sobretodo en pequeñas compañías de desarrollo o en aquellas dedicadas al software incrustado en tiempo real o al software de telecomunicaciones, que impide la pérdida de la ventana de mercado y sitúa su costo en un punto medio y razonable para la empresa.	
Según el informe CHAOS, las principales causas de fracaso de los proyectos están relacionadas con la escasa formación técnica de los desarrolladores.	
Con una <b>estrategia reactiva</b> para tratar el riesgo en un proceso de desarrollo software, se identifican los riesgos, se valora la probabilidad de que ocurran, se estima su impacto (daño que ocasionaría), se establece un plan de contingencia por si ocurre, y se clasifican dichos riesgos por probabilidad e impacto.	
Con respecto a las <b>técnicas empíricas</b> de estimación, podemos afirmar que se basan en datos históricos de proyectos, que aplican técnicas de regresión lineal y que una vez calibrados son efectivos y rápidos.	
El modelo de proceso <b>incremental</b> es aplicable, entre otras situaciones, cuando los requerimientos iniciales están suficientemente bien definidos.	
Una superclase contiene todas la propiedades (atributos, operaciones y asociaciones) comunes, que son heredados por las subclases. Por lo tanto la <b>generalización</b> consiste en factorizar las propiedades comunes de un conjunto de clases en una clase más general.	
Los requisitos no <b>funcionales</b> identifican restricciones impuestas por el entorno, la tecnología, etc.	
Entre las propiedades deseables en un requisito se encuentra la independencia de la implementación.	
El conjunto formado por los Requisitos del Producto, los Requisitos de Organizaciones y los Requisitos Externos, constituyen los denominados Requisitos de Ámbito.	
El Desarrollo Conjunto de Aplicaciones ( <b>JAD</b> ), la Tormenta de Ideas ( <b>Brainstorming</b> ) o los <b>Casos de Uso</b> , son ejemplos de técnicas para elicitar requisitos.	
Los <b>Casos de Uso</b> no describen el funcionamiento interno del sistema o su diseño, sólo las responsabilidades que tiene.	
Un Caso de Uso A <b>excluye</b> a otro Caso de Uso B cuando A <b>completa la funcionalidad</b> de B.	
Una forma de organizar los Casos de Uso es recurrir a los <b>paquetes</b> de UML.	
Según las necesidades los Casos de Uso pueden especificarse con distinto nivel de detalle.	
En una <b>agregación</b> (◇), los componentes no pueden ser compartidos por varios objetos compuestos, la cardinalidad máxima a nivel de objeto compuesto es obligatoriamente 1 y la supresión del objeto compuesto comporta la supresión de los componentes	
<b>UML</b> es un lenguaje de modelado visual de propósito general que se utiliza para especificar, visualizar, construir y documentar artefactos de un sistema software.	