

HORA 15

Adaptación del UML en un proceso de desarrollo

Ahora que ha comprendido los diagramas del UML y su estructura, ya casi es hora de ponerlo a funcionar. El UML es una herramienta maravillosa, pero no la utilizará de manera aislada. El UML tiene la intención de impulsar el desarrollo de software, por ello, es necesario aprender los procesos y metodologías de desarrollo como un vehículo para comprender el uso del UML en un contexto.

En esta hora se tratarán los siguientes temas:

- Por qué es importante un proceso de desarrollo
- Por qué no son adecuadas las antiguas metodologías para los sistemas de hoy
- El Proceso de desarrollo GRAPPLE
- Cómo incorporar al UML en el proceso

Su empresa requiere un nuevo sistema de cómputo. Los nuevos componentes de hardware y software darán por resultado una ventaja competitiva, misma que usted necesitará. Debe empezar el desarrollo, ¡y pronto!

Es usted quien ha tomado la decisión de generar al nuevo sistema, ha establecido un equipo de desarrollo completo, con un gerente de proyectos, modeladores, analistas, programadores e ingenieros de sistemas. Ya todos se trueñan los dedos, ansiosos por empezar.

Usted es, en otras palabras, un cliente. ¿Cuáles productos de trabajo esperará obtener del equipo de desarrollo? ¿Cómo querrá que el gerente de proyectos le reporte? Al final, claro, querrá un sistema en funcionamiento; pero antes de ello, necesitará ver indicios de que el equipo ha comprendido el problema que usted intenta resolver y su visión para hacerlo. Necesitará ver el avance de su solución, y necesitará saber cuál ha sido su avance en diferentes puntos.

Tales inquietudes son comunes con cualquier cliente y en cualquier proyecto de desarrollo que requiera una considerable cantidad de tiempo, dinero y recursos humanos.

Metodologías: antiguas y recientes

A usted no le gustaría que el equipo de desarrollo comenzara a codificar sin más. Después de todo, ¿qué sería lo que codificarían? El equipo de desarrollo tiene que proceder de manera más estructurada y metódica. La estructura y naturaleza de los pasos en un esfuerzo de desarrollo es lo que yo entiendo como *metodología*.

Antes de comenzar a programar, los desarrolladores tienen que comprender con claridad el problema. Esto requiere que alguien analice sus requerimientos. Una vez hecho ese análisis, ¿se podrá iniciar la codificación? No. Alguien tiene que convertir tal análisis a diseño. De esta manera, los codificadores comenzarán a producir el código a partir del diseño, después de probar y distribuir el código se convertirá en un sistema.

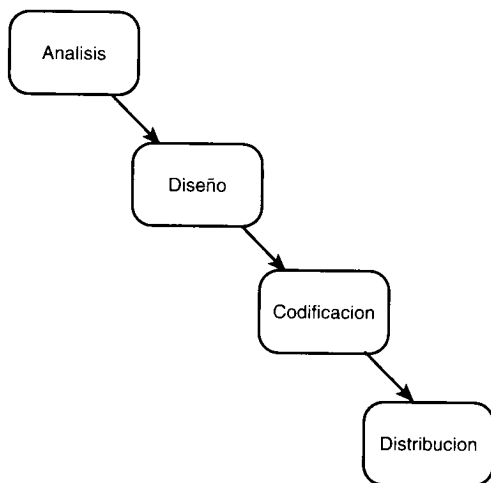
El método antiguo

Esta idea demasiado simplificada del proceso de desarrollo podrá darle una idea de que las etapas deberán sucederse en lapsos claramente definidos, una después de otra. De hecho, las metodologías de desarrollo iniciales se estructuraban de esa manera. La figura 15.1 le muestra una forma de pensar cuya influencia trascendió por varios años. Éste es el método “en cascada”, y establece que el análisis, diseño, codificación y distribución van uno después de otro como las actividades en un diagrama de actividades: solamente cuando se haya completado uno se podrá iniciar el otro.

Esta forma de hacer las cosas tiene algunos puntos inquietantes. Por un lado, tiende a la realización de tareas individuales. Si un analista no tiene contacto con un diseñador, y éste a su vez no tiene contacto con un desarrollador, existe la posibilidad de que los tres miembros rara vez trabajen juntos para compartir puntos de vista importantes.

FIGURA 15.1

El método “en cascada” del desarrollo del software.



Otro problema con este método es que reduce el impacto de la comprensión obtenida en el proyecto. (No se equivoque, la comprensión evoluciona durante la vida de un proyecto —aun después de que un análisis se haya volcado en un diseño—). Si el proceso no puede retroceder y volver a ver los primeros estados, es posible que las ideas desarrolladas no sean utilizadas. Intentar introducir con calzador nuevos puntos de vista durante el desarrollo es, cuando menos, bastante difícil. La revisión de un análisis y diseño —y la ulterior incorporación de una idea desarrollada— establece una mayor oportunidad de éxito.

El método reciente

En contraste con el método de cascada, la moderna ingeniería de programas tiende a la colaboración entre las fases del desarrollo. Los analistas y los diseñadores, por dar un ejemplo, hacen revisiones para desarrollar un sólido fundamento para los desarrolladores. Éstos, a su vez, interactúan con los analistas y los diseñadores para compartirles sus puntos de vista, modificar los diseños y fortalecer su código.

La ventaja es que conforme crece la comprensión, el equipo incorpora nuevas ideas y genera un sistema más confiable. La desventaja (en caso de haberla) es que algunas personas no son muy participativas y pueden mantenerse al margen. En ocasiones, los gerentes de proyectos quisieran decirle a sus clientes: “Ya se finalizó el análisis y ahora continuaremos con el diseño. Luego de unos dos o tres días de diseño, empezaremos a codificar.”

Tal mentalidad es peligrosa. Establecer barreras artificiales entre las fases podría dar por resultado un sistema que no haga exactamente lo que los clientes desean.

El método antiguo fomenta otro problema: es común el caso de que los partidarios al método “en cascada” reparten el tiempo del proyecto en la codificación. El verdadero efecto de esto es que se quita un tiempo valioso al análisis y diseño.

Lo que debe hacer un proceso de desarrollo

En los primeros años de la programación de computadoras, una persona podía analizar un problema, otorgar una solución y escribir un programa. En los primeros años de la construcción de viviendas (cuando la tierra era plana), una persona podía construir una casa, también.

En la actualidad la historia ha cambiado. Para desarrollar las diferentes naturalezas de sistemas complejos que demandan los negocios de hoy, es más necesario el uso de un equipo. ¿Por qué? El conocimiento se ha especializado tanto que una persona no puede conocer todas las facetas de un negocio, comprender un problema, diseñar una solución, traducirla a un programa, distribuir el programa en el hardware y asegurarse que los componentes del hardware funcionen de forma conjunta.

El equipo tiene que formarse de analistas para comunicarse con el cliente y comprender el problema, diseñadores para generar una solución, programadores para codificarla e ingenieros de sistemas para distribuirla. Un proceso de desarrollo tiene que tomar en cuenta todos los procesos anteriores, utilizarlos adecuadamente y asignar la cantidad de tiempo necesaria para cada fase. El proceso también debe dar por resultado diversos productos del trabajo que den indicios de progreso y conformar una estela de responsabilidad.

Finalmente, el proceso deberá asegurar que sus fases no sean discontinuas. En lugar de ello, debe obtenerse información entre las fases para fomentar la creatividad y aumentar la facilidad de innovar. La base sería: es más sencillo hacer una modificación a un proyecto y luego hacerla en la casa, en lugar de modificar la casa mientras construye la estructura.

Al llegar a un proceso, existe la tentación de generar una serie de fases que podrían traer una gran cantidad de papeleo. Algunas metodologías que están disponibles de forma comercial lo hacen, con lo que hacen que los gerentes de proyectos llenen interminables formularios. El papeleo se complica por sí mismo.

Una razón de esto es la idea errónea de que la metodología de la “unitalla” es posible; cada empresa es única. Una empresa tiene su propia cultura, normatividad, historia y personal. La metodología del desarrollo que pueda aplicarse a un consorcio internacional posiblemente fallará en una pequeña empresa y viceversa. Al intentar meter con calzador una metodología en una empresa, se tendrá la mala impresión de que un papeleo extremo podrá ayudar.

Así que aquí está el reto. Un equipo de desarrollo deberá:

- Asegurar que el equipo de desarrollo cuenta con una firme comprensión del problema que se intenta resolver
- Dar pie a un equipo que conste de una colección de responsabilidades
- Fomentar la comunicación entre los miembros del equipo que ostentan tales responsabilidades
- Dar pie a la intercomunicación entre las fases del proceso de desarrollo
- Desarrollar productos de trabajo que comuniquen el progreso al cliente, y eliminar el papeleo superfluo

¡Ah! Por cierto, sería una buena idea si el proceso origina un producto terminado en un lapso corto.



Se habrá dado cuenta que utilizo las palabras *proceso* y *metodología* de forma indistinta. Aunque es posible encontrar algunas diferencias entre ellas, prefiero no discutir los detalles. En mi experiencia, la palabra *metodología* se ha tergiversado de forma paulatina. Creo que si se mezcla la palabra *proceso* en el tema, se puede eliminar tal tergiversación.

GRAPPLE

Para enfrentar este reto de varias facetas, le presento GRAPPLE (Guías para la Ingeniería de Aplicaciones Rápidas)). Las ideas dentro de GRAPPLE no son originales. Son una condensación de las ideas de varias otras personas. Los Tres Amigos han creado el Proceso Racional Unificado, y antes de ello, cada Amigo tenía su propio proceso; las ideas en tales procesos son similares a GRAPPLE. El libro de Steve McConnell, *Rapid Development* (Microsoft Press, 1996), contiene varias de las mejores prácticas que se aplican al... bueno... desarrollo rápido de aplicaciones.

La primera palabra en las siglas de GRAPPLE, *Directivas*, es importante: ésta no es una férrea metodología. En vez de ello, es un conjunto de ideas adaptables y flexibles. Imagínelas como un patrón simplificado de un proceso de desarrollo. Lo presento como un vehículo para mostrar al UML dentro de un contexto. Con algunos ajustes, GRAPPLE puede aplicarse en diversas organizaciones (aunque, tal vez, no a todas). Da la oportunidad a un gerente de proyectos, con creatividad, de agregar sus propias ideas respecto a lo que funcionará en una organización en particular, y puede sustraer los pasos incluidos que no funcionen.



Antes de adentrarnos en el tema de GRAPPLE, he aquí una pregunta que tal vez se formule: “¿Por qué me está hablando de esto un libro que, se supone, trata del UML?”

La respuesta es que si no le digo a usted algo respecto al proceso de desarrollo y le doy un contexto para utilizar al UML, todo lo que habré hecho es mostrarle cómo dibujar diagramas. Lo importante de esto es mostrarle cómo y cuándo utilizar cada uno de ellos.

En la parte II, “Estudio de un caso”, verá un caso de prueba que aplicará a GRAPPLE en el UML.

RAD³: la estructura de GRAPPLE

GRAPPLE consta de cinco *segmentos*. He utilizado “segmentos” en lugar de “fases” para eliminar la idea de que una “fase” debe completarse antes de iniciar la otra. (Resistí la tentación de llamarlos “piezas”. “Cinco piezas fáciles” era demasiado hermoso.) Cada segmento, en turno, consta de diversas *acciones*. Cada acción trae consigo un *producto del trabajo*, y cada acción es responsabilidad de un *jugador*.

En muchos casos, el gerente de proyectos puede combinar los productos de trabajo en un informe que presente al cliente. Los productos de trabajo, de hecho, tienen el mismo propósito que un avance en papel, sin sumergirse en el papeleo.



Para adaptar a GRAPPLE, un gerente de proyectos podría agregar acciones a cada segmento. Hay otra posibilidad, que es profundizar a un nivel inferior, y subdividir a cada acción en subacciones. Aun hay otra posibilidad de reordenar las acciones dentro de cada segmento. Las necesidades de organización establecerán el camino a seguir.

GRAPPLE se encausa a los sistemas orientados a objetos. Por ello, las acciones dentro de cada segmento se orientan a crear productos de trabajo de una naturaleza orientada a objetos.

Los segmentos son:

1. Recopilación de necesidades
2. Análisis
3. Diseño
4. Desarrollo
5. Distribución

Esto nos otorga un acrónimo RADDD o RAD³. Luego del tercer segmento, el gerente de proyectos combina los productos de trabajo en un documento de diseño para dárselo al cliente y los desarrolladores. Cuando se han completado todos los segmentos RAD³, todos los productos de trabajo se combinan para conformar un documento que define al sistema.

Antes de iniciar tales segmentos, debe asumir que el cliente ha generado un caso de negocios para el nuevo sistema. También debe asumir que los miembros del equipo de desarrollo, particularmente los analistas, han leído tanta documentación relevante como sea posible.

Examinemos cada segmento con mayor atención, sin perder de vista las partes del UML que se ajusten a cada uno.

Recopilación de necesidades

Si intenta asignar una importancia relativa a cada segmento, éste es un buen candidato para ser el *número uno*. Si no comprende lo que desea el cliente, nunca podrá generar el sistema adecuado. Todos los análisis de casos de uso en el mundo no le ayudarán si no comprende las bases del dominio del cliente y el problema que quiera que usted resuelva.

Descubra los procesos de negocios

Es una buena idea empezar el proceso de desarrollo mediante la comprensión de los procesos de negocios del cliente, en especial aquellos que tratará de mejorar con el sistema propuesto. Para comprenderlo, un analista entrevistará al cliente o a una persona con el conocimiento necesario que sea designada por el cliente, a quien le preguntará los pasos relevantes del proceso uno por uno.

Una consecuencia importante será que el analista obtendrá un vocabulario de trabajo en un subconjunto de la terminología del cliente. El analista usará este vocabulario cuando entreviste al cliente en la siguiente acción.

El producto del trabajo es un diagrama de actividades o conjunto de ellos que captan los pasos y puntos decisivos en el proceso (o procesos) del negocio.

Realice un análisis del dominio

Esta acción es como el ejemplo de la plática con el entrenador de baloncesto. Puede realizarse durante la misma sesión en la acción anterior. El objetivo es comprender de la mejor manera posible el dominio del cliente. Observe que esta acción y la anterior tratan de conceptos, no del sistema que va a generar. El analista tiene que acomodarse en el mundo del cliente, pues, a fin de cuentas, es el interlocutor entre el cliente y el equipo de desarrollo.

El analista entrevista al cliente con la finalidad de comprender las principales entidades en el dominio del cliente. Durante la plática entre el cliente y el analista, otro miembro del equipo tomará las notas (de forma óptima) en un equipo de cómputo portátil equipado con un procesador de textos, y un modelador de objetos generará un diagrama de clases de alto nivel. Si puede contar con más de un miembro del equipo que tome notas, por su bien hágalo.

El modelador de objetos prestará atención a los sustantivos y empezará a convertir a cada uno en una clase. Finalmente, algunos de esos sustantivos se convertirán en atributos. El modelador también prestará atención a los verbos, que se convertirán en operaciones de las clases. En este punto, una herramienta de modelado computarizada sería muy útil.

El producto del trabajo es un diagrama de clases de alto nivel y un conjunto de minutas.

¿GRABAR O NO GRABAR?

¿Debería grabar en cinta de sonido tales entrevistas o tan sólo basarse en las minutas? Ésta es una pregunta común. Cuando se graba una entrevista, se tiende a no prestar tanta atención, o a sólo tomar algunas notas muy obligatorias. (Después de todo, siempre podrá recurrir a la cinta). Si decide grabar, le sugiero que ignore a la grabadora y tome todas las notas como si la grabadora no existiera.

La grabación en cinta de sonido puede ser muy útil cuando se encuentra en proceso de entrenamiento de un nuevo modelador de objetos. Un modelador experimentado podrá comparar los diagramas del nuevo mediante la grabación del debate y verificar su minuciosidad.

Identificación de los sistemas cooperativos

El poeta del s. XVII John Donne, escribió: “Nadie es una isla, completo en sí mismo”. Si él hubiera escrito en la época actual, tal texto debió decir “Ninguna *persona* es una masa de tierra rodeada de agua, completa en sí misma”. También pudo haber escrito “Ningún *sistema* es una isla...”, y así por el estilo.

En cualquier caso Donne estaría en lo correcto. Normalmente, los sistemas de negocios actuales no emergen de la nada, tienen que colaborar con otros. En las primeras instancias del proceso, el equipo de desarrollo verá exactamente de qué sistemas dependerá el nuevo sistema, y cuáles dependerán de él. Un diseñador de sistemas se encargará de esto, y producirá un diagrama de distribución como su producto del trabajo. El diagrama muestra a los sistemas como nodos, con líneas de comunicación entre ellos, componentes residentes y dependencias entre componentes.

Descubra las necesidades del sistema

Descubrir las necesidades es muy importante, ya que en esta acción, el equipo realiza su primera sesión de JAD (Desarrollo conjunto de aplicaciones). Habrá otras más en el curso del GRAPPLE.

Una sesión JAD reúne a quienes toman las decisiones en la empresa del cliente, a los usuarios potenciales y a los miembros del equipo de desarrollo. Debe haber alguien que modere la sesión; el trabajo del moderador es obtener una respuesta de quienes toman las decisiones y de los usuarios acerca de lo que esperan que haga el sistema. Al menos deberá haber dos miembros del equipo que tomen notas, y el modelador de objetos deberá refinar el diagrama de clases que se obtuvo previamente.

El producto del trabajo es un diagrama de paquetes. Cada paquete representa a un área de alto nivel de la funcionalidad del sistema (por ejemplo: “ayuda con el servicio a clientes”). Cada paquete agrupa un conjunto de casos de uso (por ejemplo: “obtener el historial del cliente” o “tratar con el cliente”).

La complejidad del sistema será lo que determine la duración de la sesión. Casi nunca será menor a medio día laboral, y podría durar hasta toda una semana laboral. La empresa del cliente debe hacer el compromiso de invertir el tiempo que sea necesario.

¿Para qué acceder a una sesión JAD para desarrollar los requerimientos del sistema? ¿Por qué no sólo entrevistar a cada individuo? Como podrá recordar, dije que la última parte del reto de un proceso de desarrollo es generar un sistema en un corto lapso. Las entrevistas individuales pueden tardar semanas, mucho más si existen conflictos en los itinerarios de las personas. La espera de entrevistas individuales puede ocupar mucho tiempo y, con él, puede irse por tierra la supuesta ventaja competitiva de completar rápidamente el sistema. Las entrevistas individuales posiblemente contendrían puntos de vista conflictivos, y se perdería tiempo en intentar resolverlos. Agruparlos a todos crea una expectativa general, en la que los participantes podrían hacer una simbiosis de sus puntos de vista en beneficio de todos.

Presentar los resultados al cliente

Cuando el equipo finaliza todas las acciones de Necesidades, el administrador de proyectos presentará los resultados al cliente. Algunas empresas podrían requerir la aprobación del cliente en este punto, para que pueda proceder el desarrollo. Otras podrían necesitar una estimación de los costos de acuerdo con los resultados. De esta manera, el producto del trabajo podría variar de acuerdo con la empresa.

Análisis

En este segmento, el equipo profundiza en los resultados del segmento Necesidades y aumentará su comprensión del problema. De hecho, partes de este segmento empezarán durante el segmento de Necesidades, conforme el modelador de objetos empieza a depurar el diagrama de clases durante la sesión JAD de Necesidades.

Comprensión del uso del sistema

Esta acción es un análisis de casos de uso de alto nivel. En una sesión JAD con usuarios potenciales, el equipo de desarrollo trabaja con los usuarios para descubrir a los actores que iniciarán cada caso de uso, y los actores que serán beneficiados. (Recuerde que un actor puede ser un sistema o una persona.) Un moderador interviene en la sesión, y dos miembros del equipo toman notas. Luego de algunos proyectos, el moderador de esta sesión podría convertirse en el analista de casos de uso.

El equipo también intentará desarrollar nuevos casos de uso y casos de uso abstractos. El producto del trabajo será un conjunto de diagramas de casos de uso que muestren a los actores y las dependencias estereotipadas (“extender” e “incluir”) entre los casos de uso.

Hacer realidad los casos de uso

En esta acción, el equipo de desarrollo continúa su trabajo con los usuarios. El objetivo es analizar la secuencia de pasos en cada caso de uso. Esta sesión JAD puede ser la continuación de la sesión previa. Cuidado: ésta es, por lo general, la sesión JAD más compleja para los usuarios. Tal vez no estén acostumbrados a dividir una operación en los pasos que la conforman y, a su vez, tampoco puedan enumerarlos. El producto del trabajo es una descripción textual de los pasos en cada caso de uso.

Depurar los diagramas de clases

Durante las sesiones JAD, el modelador de objeto escuchará todos los debates y continuará con su depuración del diagrama de clases. En este punto, el modelador de objetos deberá rellenar los nombres de las asociaciones, clases abstractas, multiplicidades, generalizaciones y agregaciones. El producto del trabajo es un diagrama de clases depurado.

Analizar cambios de estado en los objetos

El modelador de objetos depurará el modelo mediante la presentación de cambios de estado conforme sea necesario. El producto del trabajo es un diagrama de estados.

Definir la comunicación entre objetos

Ahora que el equipo cuenta con un conjunto de diagramas de casos de uso y un diagrama depurado de clases, se definirá la forma en que los objetos se comunican. El modelador de objetos desarrollará un conjunto de diagramas de secuencias y de colaboraciones para delinear la comunicación. Deberán incluirse los cambios de estado. Estos diagramas conforman el producto del trabajo de esta acción.

Analizar la integración con diagramas de colaboraciones

Al tiempo de realizar los pasos anteriores, el diseñador del sistema descubre los detalles específicos de la integración con los sistemas cooperativos. ¿Qué tipo de comunicación está envuelto? ¿Cuál es la arquitectura de red? Si el sistema tendrá que utilizar bases de datos, un analista de bases de datos determinará la arquitectura (física o lógica) de ellas. Los productos del trabajo son diagramas de distribución detallados y (de ser necesario) modelos de datos.

Diseño

En este segmento, el equipo trabajará con los resultados del segmento de Análisis para diseñar la solución. En el diseño y en el análisis se harán las revisiones pertinentes hasta que el diseño se haya completado. De hecho, algunas de las metodologías combinan al Análisis y al Diseño en una sola fase.

Desarrollo y depuración de los diagramas de objetos

Los programadores tomarán el diagrama de clases y generarán cualesquier diagramas de objetos que sea necesario. Darán vida a los diagramas de objetos mediante el análisis de cada operación y el desarrollo de un diagrama de actividades correspondiente. Los diagramas de actividades fungirán como la base de gran parte del código en el segmento de desarrollo. Los productos del trabajo serán los diagramas de objetos y los de actividades.

Desarrollo de diagramas de componentes

Los programadores serán quienes jueguen un importante papel en esta acción. La tarea será visualizar los componentes que resultarán del siguiente segmento y mostrar las dependencias entre ellos. Los diagramas de componentes serán el producto del trabajo.

Planeación para la distribución

Cuando se haya completado el diagrama de componentes, el diseñador del sistema empezará a planear la distribución e integración con sistemas cooperativos. Creará un diagrama de distribución que muestre el lugar donde se encontrarán los componentes. El producto del trabajo será un diagrama que sea parte del de distribución generado con anterioridad.

Diseño y prototipos de la interfaz del usuario

Esto trae consigo otra sesión JAD con los usuarios. Aunque esto es parte del Diseño, esta sesión puede ser la continuación de anteriores sesiones JAD con los usuarios —un indicio de la interacción entre el Análisis y el Diseño—.

La interfaz del usuario debería permitir la consumación de todos los casos de uso. Para ello, un analista de GUI deberá trabajar con los usuarios para desarrollar prototipos, en papel, de las pantallas que corresponderán a grupos de casos de uso. Los usuarios pegarán papeletas removibles que representen los componentes de la pantalla (botones, casillas de verificación, listas desplegables, menús y cosas así). Cuando los usuarios queden satisfechos de la posición de los componentes, los desarrolladores generarán prototipos de las pantallas para que sean aprobados por los usuarios. Los productos del trabajo serán capturas de pantalla de los prototipos resultantes.

Pruebas de diseño

Los casos de uso permiten el diseño de pruebas del software. El objetivo es evaluar si el software hace lo que se supone que debería (esto es, que hace lo que se especifica en los casos de uso). Preferentemente, un desarrollador o especialista de pruebas externo al equipo de desarrollo deberá utilizar los diagramas de casos de uso para crear secuencias de comandos en herramientas automatizadas de pruebas. Tales secuencias de comandos conformarán el producto del trabajo.

Iniciar la documentación

Nunca es demasiado pronto para empezar a documentar el sistema para los usuarios finales y gerentes de sistemas. Los especialistas en la documentación trabajarán en conjunto con los diseñadores para empezar a generar un panfleto de la documentación y llegar a una estructura de alto nivel para cada documento. Tal estructura es el producto del trabajo.

Desarrollo

De este segmento se encargan los programadores. Con suficiente análisis y diseño, este segmento debería realizarse con rapidez y sin problemas.

Generación del código

Con los diagramas de clases, de objetos, de actividades y de componentes a la mano, los programadores generarán el código del sistema. Tal código es el producto del trabajo de esta acción.

Verificación del código

Los especialistas en pruebas (no los desarrolladores) ejecutarán secuencias de comandos de prueba para evaluar si el código hace lo que se pretende. Los resultados de las pruebas son los productos del trabajo. Esta acción alimenta a la anterior y viceversa, hasta que el código pase todos los niveles de prueba.

Generación de interfaces del usuario, conexión con el código y prueba

Esta acción crea las interfaces de usuario ya aprobadas. El especialista en GUI las genera y conecta con el código. Las pruebas ulteriores aseguran que las interfaces funcionen adecuadamente. El sistema en funcionamiento junto con las interfaces de usuario, son el producto del trabajo.

Consumación de la documentación

Durante el segmento de desarrollo, los especialistas en documentación trabajan en paralelo con los desarrolladores para asegurar la entrega oportuna de toda la documentación, la cual es el producto del trabajo de esta acción.

Distribución

Cuando un sistema se ha finalizado, se distribuye en el hardware adecuado y se integra con los sistemas cooperativos. No obstante, la primera acción en este segmento puede iniciar antes de que el segmento de Desarrollo comience.

Planeación para copias de seguridad y recuperación

El diseñador del sistema creará un plan que incluya los pasos a seguir en caso de que el sistema falle. El plan, producto del trabajo de esta acción, establece lo que se deberá hacer para crear una copia de seguridad del sistema y para recuperarse del error.

Instalación del sistema terminado en el hardware adecuado

El diseñador del sistema, con toda la ayuda necesaria de los programadores, distribuye el sistema terminado en los equipos de cómputo adecuados. El producto del trabajo es el sistema completamente distribuido.

Verificación del sistema instalado

Finalmente, el equipo de desarrollo verifica el sistema instalado. ¿Se ejecuta como se esperaba? ¿El plan de copias de seguridad y recuperación funciona? Los resultados de estas pruebas determinarán si se necesita hacer una depuración ulterior. Tales resultados conforman el producto del trabajo de esta acción.

Celebración

Sin mayor explicación, el equipo de desarrollo podrá inventar los productos del trabajo de esta acción.

Resumen de GRAPPLE

Si revisa los segmentos y acciones en GRAPPLE, verá que los movimientos van de lo general a lo específico: de lo rústico a lo refinado. Empieza con una asimilación conceptual del dominio, trasciende a la funcionalidad de alto nivel, profundiza en los casos de uso, depura los modelos y diseña, desarrolla y distribuye el sistema.

También notó que hay más acciones en los segmentos de Análisis y Diseño que en el de Desarrollo. Esto es por diseño, valga la redundancia. La idea es utilizar tanto tiempo como sea necesario en el análisis y diseño, para que la codificación se realice sin problemas. Podría parecer una herejía, pero en el mundo ideal la codificación es sólo una pequeña parte del desarrollo de sistemas. Entre más analice, más cerca estará del ideal.

GRAPPLE, como lo dije, es un patrón simplificado de un proceso de desarrollo. No me centré en los detalles en ciertos puntos importantes, como los niveles de prueba. También pasé por alto algunas cuestiones básicas: ¿Dónde y cómo el equipo alberga los productos del trabajo en ejecución? ¿Cómo trata el equipo el importante punto de la administración de la configuración?

No aludí a estos puntos porque se salen del tema del UML. Una respuesta corta para estos puntos importantes es cobijarse en la tecnología. Los productos del trabajo (finalizados o en ejecución) pueden encontrarse en un área de almacenamiento que se encuentre en la red de la empresa. Una opción es contar con una jerarquía de directorios a la que puedan acceder los miembros del equipo. Una opción más segura es instalar un paquete de almacenamiento central que lleve un control del cumplimiento e inicio de los productos del trabajo, y que sólo permita que una persona a la vez verifique una copia modificable de un elemento. Éste es el fundamento de una solución para la administración de la configuración. La tecnología de almacenamiento avanza con regularidad, y hay varias opciones.

La hora siguiente dará inicio a la parte II, el estudio de un caso que aplica tanto al UML como a GRAPPLE.

Resumen

Una metodología de desarrollo estructura los segmentos y actividades en un proyecto de desarrollo de sistemas. Sin una metodología habría un caos y los desarrolladores no comprenderían el problema que se supone deberían resolver, así como los sistemas no cumplirían con las necesidades de los usuarios. Las metodologías de antaño forzaban a una secuencia “en cascada” de análisis, diseño, codificación y distribución.

Este tipo de metodología secuencial podía fragmentar el desarrollo, de modo que un equipo de desarrollo podría no aprovechar la mejor asimilación que se obtiene durante la vida de un proyecto. Por lo general, también distribuye la mayor parte del tiempo en la codificación, y esto resta una enorme cantidad de tiempo al análisis y diseño.

Esta hora presentó al GRAPPLE (Directivas para el Rápido Diseño de Aplicaciones), un patrón para el proceso de desarrollo. GRAPPLE consta de cinco segmentos: Recopilación de necesidades, Análisis, Diseño, Desarrollo y Distribución. Cada segmento consta de diversas acciones, y cada una de ellas da por resultado un producto del trabajo. Los diagramas UML constituyen productos del trabajo para varias de las acciones.

La parte II aplica a GRAPPLE y el UML al estudio de un caso.

Preguntas y respuestas

- P** ¿En algún momento se podría aplicar el método “en cascada”?
- R** Si el ámbito del sistema propuesto es muy pequeño (claro que es algo subjetivo), podría aplicarlo sin problemas. No obstante, en el moderno desarrollo de sistemas orientados a objetos, una metodología que propenda a la interacción entre los segmentos de desarrollo podrá otorgar un mejor resultado.
- P** En la respuesta anterior, mencionó el desarrollo de sistemas orientados a objetos. Suponga que el sistema propuesto no está orientado a objetos. ¿Aún así se aplica?
- R** Incluso en los sistemas que no estén orientados a objetos (como en muchos proyectos basados en grandes computadoras centralizadas) se aplican las ideas que ha visto en esta hora. Las sesiones JAD, un análisis y diseño de avanzada y la interacción entre los segmentos de desarrollo inclusive serán muy útiles. Podría adaptar a GRAPPLE (por ejemplo: mediante la eliminación del modelado de clases), pero esa es la idea: es un conjunto de directivas flexibles en lugar de ser una metodología que tenga que seguirse a pie juntillas.

Taller

Ahora que ya comprendió las metodologías, verifique, con las siguientes preguntas, qué tanto ha asimilado. El apéndice A, “Respuestas a los cuestionarios”, le dará las respuestas.

Cuestionario

1. ¿Cuáles son algunas de las inquietudes de un cliente?
2. ¿Qué se debe comprender como *metodología* de desarrollo?
3. ¿Cuál es el método “en cascada”? ¿Cuáles son sus debilidades?
4. ¿Cuáles son los segmentos de GRAPPLE?
5. ¿Qué es una sesión JAD?



PARTE II

Estudio de un caso

Hora

- 16 Presentación del caso por estudiar
- 17 Elaboración de un análisis de dominio
- 18 Recopilación de las necesidades del sistema
- 19 Desarrollo de los casos de uso
- 20 Orientación a las interacciones y cambios de estado
- 21 Diseño del aspecto, sensación y distribución
- 22 Noción de los patrones de diseño



HORA 16

Presentación del caso por estudiar

Ahora que ya cuenta con cierta experiencia del UML y se le ha presentado el patrón de una metodología de desarrollo, verá cómo aplicar el UML en un proceso de desarrollo. A partir de aquí daremos inicio a la parte II, el estudio de un caso que aplica al UML bajo el contexto del proceso GRAPPLE.

En esta hora se tratarán los siguientes temas:

- El panorama del caso por estudiar
- Cómo descubrir y modelar los procesos del negocio
- Sugerencias al hacer entrevistas

La empresa multinacional (y ficticia) LaHudra, Nar y Goniff, S.A., ha hecho una encuesta sobre el mundo de los restaurantes y ha llegado a sorprendentes conclusiones: a la gente le gusta comer fuera, pero no disfrutan algunos momentos de esa experiencia.

“Bueno”, dijo LaHudra, “pude haber predicho los resultados de nuestra encuesta. Cuando salgo a comer, me disgusta cuando el mesero toma mi orden y se desaparece por una hora. Al ir uno a un restaurante con clase se espera un trato diferente”.

“Cierto”, respondió Nar, “En ocasiones cambio de parecer luego de hacer mi orden y quiero decirle al mesero que aguarde. O tengo una pregunta... o algo... y no puedo encontrarlo.”

Goniff asiente: “Estoy de acuerdo. Pero, con todo, el comer fuera es divertido; me agrada cuando alguien me sirve y además, me gusta la idea de que el personal de cocina me prepare la comida. Los resultados de nuestra encuesta muestran que la mayoría de las personas también opinan de esa forma.”

“¿No habría alguna manera en que podamos mantener la experiencia básica y mejorarla de alguna forma?”

“¿Cómo?”, preguntó Nar.

“¡Yo sé cómo!”, dijo LaHudra. “Con tecnología.”

Y así fue que decidieron que uno de sus equipos de desarrollo de software corporativo construyera el restaurante del futuro.

Aplicación de GRAPPLE al problema

Los miembros del equipo de desarrollo se ciñen al esquema de GRAPPLE. Saben que la mayor parte del tiempo en el proyecto deberán orientarlo al análisis y diseño. De esa forma, la codificación se generará con rapidez y eficiencia, con lo que aumentará la posibilidad de una instalación y distribución sin problemas.

El proyecto debe iniciar con la recopilación de necesidades, y con la asimilación del dominio del restaurante. Como podrá recordar de la hora anterior, la recopilación de necesidades consta de las siguientes acciones:

- Descubrir los procesos del negocio
- Realizar un análisis del dominio
- Identificar los sistemas cooperativos
- Descubrir los requerimientos del sistema
- Presentar los resultados al cliente

En esta hora trataremos la primera acción.

Descubrir los procesos del negocio

LaHudra, Nar y Goniff lo hacen todo a lo grande. Están listos para entrar en el mundo de los restaurantes y han conformado una División de Restaurantes LNG. Han contratado a varios restauranteros, meseros (camareros), chefs y personal de mantenimiento experimentados.

Todo lo que esperan es el apoyo tecnológico para el restaurante del futuro. Luego, establecerán su primer restaurante, íntegro, con todo y la tecnología para mejorar el placer de comer en un restaurante.

Los miembros del equipo de desarrollo están de suerte. Iniciarán con un papel en blanco. Todo lo que deberán hacer será comprender los procesos del negocio y el dominio para continuar en esa línea.

El análisis del proceso del negocio empieza con la entrevista de un analista a un restaurantero. Durante la entrevista, alguien tomará nota en una computadora portátil. Al mismo tiempo, un modelador plasmará en una pizarra blanca un diagrama de actividades que el analista, quien está tomando nota y el restaurantero podrán ver.

En las siguientes subsecciones, seguiremos la entrevista en cada proceso del negocio en un restaurante. La meta es producir los diagramas de actividades que modelen los procesos.

Servir a un cliente

“Gracias por atendernos”, dice el analista.

“Es un placer”, dice el restaurantero. “¿Qué es exactamente lo que desean saber?”

“Empecemos con una sencilla transacción de negocios. ¿Qué sucede cuando un cliente entra al restaurante?”

“Bueno, si el cliente tiene un abrigo o chaqueta, le ayudamos a quitárselo, lo almacenamos en un guardarropa y le damos un boleto para solicitarlo posteriormente. Eso mismo hacemos con un sombrero. Luego, nosotros...”

“Un momento. Suponga que hay una línea de espera. ¿Primero se forma, o da su nombre al capitán, o...?”

“No. Intentamos que se sienta tan cómodo como sea posible al llegar. Luego nos preocupamos por la línea de espera, en caso de haber alguna”

“De hecho, si hay una lista de espera, le preguntamos al cliente si hizo alguna reservación. Si la hizo, intentamos honrarlos de forma oportuna y darles asiento tan pronto como sea posible. Si no hay una reservación, deja su nombre y puede ir a nuestro bar para tomar algo antes de comer. Claro que no es obligatorio que lo hagan. Pueden tan sólo sentarse en un área de espera claramente indicada.”

“Interesante. Aún no se han sentado a comer y ya se han logrado algunos puntos decisivos.”

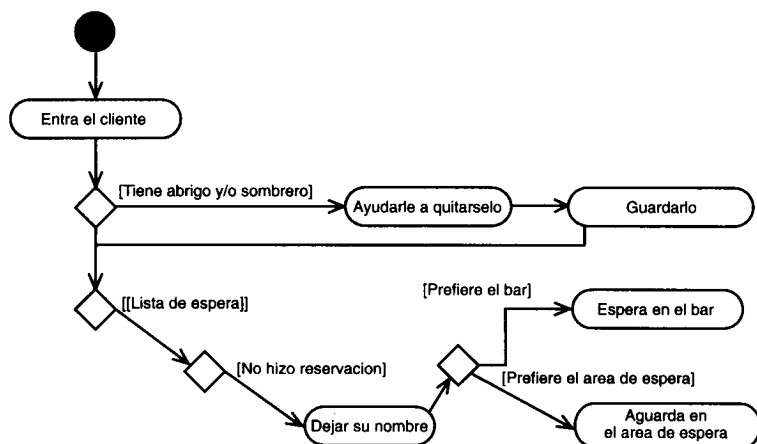
Hagamos una pausa para determinar a dónde llegamos. El diagrama del proceso del negocio ahora luce como en la figura 16.1.

De vuelta a la entrevista.

El trabajo del analista es continuar por el proceso del negocio.

FIGURA 16.1

Las fases iniciales del diagrama de actividades del proceso del negocio del restaurante, “Servir a un cliente”.



“Bien. Cuando le llegue el turno al cliente, o que haya llegado un cliente que hizo reservación, será hora de sentarlo, ¿no?”

“Sí, pero ahora que lo pienso no es tan sencillo. La mesa deberá estar lista; para ello, deberá ser limpiada. Un mozo de piso debe cambiar el mantel usado por el cliente anterior, y acomodar la mesa. Cuando está lista, el capitán de meseros lleva al cliente a su mesa y luego llama a un mesero.” “¿lo llama?”



Observe lo que hace el analista. El restaurantero ha utilizado un nuevo término (nuevo dentro del contexto de la entrevista), y el analista desea que se lo defina.

El saber cuándo y cómo hacer esto es parte del arte de entrevistar, donde la experiencia será el mejor maestro.

“Sí. No es muy complicado dado que los meseros tienen sus áreas asignadas de servicio y, por lo general, saben cuándo está lista una mesa. Normalmente circundan su área y están atentos a las expresiones del capitán.”

“¿Luego qué ocurre?”

“Bueno, el mesero llega a la mesa, entrega un menú a cada comensal y les pregunta si desean alguna bebida mientras se deciden. Luego llamará a un ‘asistente’, quien colocará una charola con pan y mantequilla, y llenará un vaso con agua para cada persona en la mesa. Si alguien ha pedido una bebida, el mesero la traerá.”

“Un momento. Dijo ‘el mesero’. ¿La persona que sirve siempre es un hombre?”

“No. Hablo de ‘el mesero’ por hábito. Lo siento.”

“Bien. ¿Qué hay si utilizamos el término neutral ‘sirviente’? También vi que el cliente tiene un par de oportunidades para pedir una bebida.”

“Sirviente no es una palabra que usemos en el ámbito restaurantero. Sugiero que sigamos con ‘mesero’ y que se aplique indistintamente a hombres y mujeres. En cuanto a lo de las bebidas, es cierto. Si un cliente está a la espera de una mesa y pasó al bar, puede llevarse su bebida a la mesa si no se la ha terminado cuando se le haya asignado una mesa. A propósito, siempre nos reservamos el derecho de negar el servicio a quien ha consumido demasiado alcohol.”



El entrevistador no es sólo un oyente pasivo luego de hacer una pregunta. En este caso, el analista ha conjugado un tema en común a partir de algunas respuestas previas, y ha hecho una pregunta basada en algo que ha salido a colación en algunas ocasiones (la oportunidad de pedir algo de tomar). La respuesta contiene un extracto de la lógica de negocios, una regla que sigue el negocio en una situación en particular. En este caso, la lógica de negocios se aplica a negar el servicio a un cliente alcoholizado.

“Es bueno saberlo. Regresemos a la mesa donde los comensales deciden qué van a consumir.”

“Sí. Siempre tenemos algunas sugerencias del día que no están en el menú, y el mesero se las propone a los clientes.”

“¿Sabe qué es lo que he visto que pasa? La gente le pide al mesero las recomendaciones, y los meseros por lo general parecen ser sinceros: le dicen si un platillo es mejor que otro. ¿Es algo que usted alienta a hacer?”

“Así es. Ciertamente nuestros meseros comen en el restaurante y tienen sus propias opiniones de lo que les gusta y lo que no. Si a ellos realmente no les gusta un platillo en particular, queremos que lo digan al chef antes que a los clientes, aunque no tengo inconveniente que expresen una preferencia. Claro que no queremos que los meseros le digan a los comensales que la comida es terrible, pero el expresar una preferencia por un platillo no está mal.”

“Bien. Vamos a resumir. El cliente o comensal deja sus abrigo, entra al bar, aguarda una mesa, se sienta ante ella, posiblemente ordena una bebida, se le sirve pan y agua y mira el menú.”



Se sugiere detenerse y resumir de vez en cuando. Le ayudará a verificar qué tanto ha asimilado y le da la oportunidad de utilizar la terminología del dominio, además de que reconforta al entrevistado saber que usted le ha puesto atención.

“Así es. El mesero regresa con una bebida y los clientes la beben mientras ven el menú. El mesero les da de cinco a diez minutos para hacer su elección y, luego, regresa. El mesero regresa antes sí, claro, los comensales hacen su elección antes.”

“¿Cómo saben que deben regresar antes?”

“Bien, los clientes llaman la atención del mesero. Por lo general está cerca del área de la mesa, a menos que haya tenido que ir a la cocina a traer un pedido o necesitara hablar con los chefs por alguna razón.”

“¿Área?”

“Sí, a cada mesero se le asigna un área que consta de varias mesas. Hay una sección asignada como área de fumadores, y el resto es para quienes no fuman.”

“¿Cómo determina quién atiende un área?”

“Alternamos a los meseros en las diferentes áreas.”

“Bueno, volvamos al proceso de servir. Los comensales hacen su elección, el mesero toma la orden en su comanda y luego...”

“Y luego notifica al chef. Esto lo hace mediante la transcripción de la elección en un formulario, llamado comanda, que le da al chef.”

“¿Qué hay en la comanda?”

“La mesa, la elección y, muy importante, la hora.”

“¿Por qué es tan importante?”

“Debido a que generalmente la cocina está (espero) muy ocupada, y el chef tiene que dar prioridad a las comandas en el orden en que hayan llegado.”

“¿Eso es complicado?”

“En realidad, se complica más de acuerdo con su naturaleza.”

“¿Cómo está eso?”

“La mayoría de las comidas incluyen un entremés antes del plato principal. A la mayoría de la gente le gusta comer el plato fuerte caliente, así que el chef prepara el entremés, muchos de ellos ya están preparados, como algunas ensaladas, y el mesero se los lleva al cliente. El reto está en llevar el plato fuerte, caliente, a todos los comensales en la mesa al mismo tiempo. Digo ‘reto’ porque las personas de la mesa por lo general se terminan el entremés en diferentes momentos. Todo el asunto debe estar coordinado.”

“Hum. Esto suena a proceso separado. Tomaremos esto en una reunión por separado desde el punto de vista del chef.”



El analista ha tomado una importante decisión: sacar del contexto una secuencia que posiblemente sea parte de un proceso separado. El reconocer en qué momento hacerlo se obtiene mediante la experiencia.

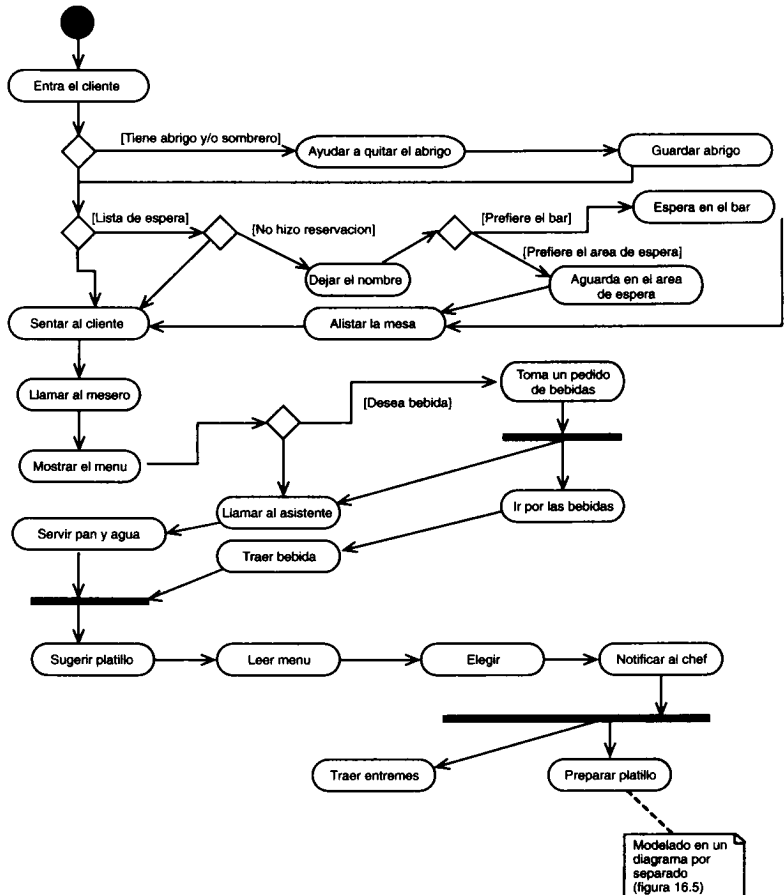
Un buen principio es que, si el entrevistado utiliza palabras como “complejo” o “complicado”, o responde que “sí” cuando le pregunta si algo es complicado, tal vez estará ante un conjunto de pasos que necesitarán su propio modelo. Deje que el entrevistado hable un poco antes de tomar una decisión como ésta.

“De acuerdo, es una buena idea.”

“Estamos en el punto donde el chef preparará el plato fuerte. A propósito, ¿qué le parece este diagrama?” (Vea la figura 16.2.)

FIGURA 16.2

Las fases intermedias del diagrama de actividades para el proceso del negocio “Servir a un cliente”.



“Creo que ha comprendido. De cualquier forma, el chef preparará el plato fuerte y el mesero lo recogerá cuando se de cuenta de que los comensales han terminado con el entremés; posteriormente, el mesero llevará el plato fuerte a la mesa. Los comensales ingerirán sus alimentos, y el mesero regresará al menos una vez para verificar si todo está bien.”

“Suponga que a un cliente no le ha satisfecho algo de la comida. ¿Qué ocurre?”

“Bueno, hacemos nuestro mejor esfuerzo para que le satisfaga, aun cuando nos cueste algo de dinero. Es mejor perder algo de dinero que a un cliente.”

“Buen concepto.”

“Gracias. Cuando los comensales terminan con sus alimentos, el mesero regresa y pregunta si desean un postre. En tal caso, el mesero dará el menú de postres y tomará las órdenes. En caso de que no deseen postre, preguntará si desean un café. Si es el caso, traerá café y tazas y les servirá. Si no desean nada, traerá la cuenta. Luego de unos instantes, regresará y obtendrá el pago en efectivo o en tarjeta de crédito. Luego traerá el cambio o los vouchers, los clientes dejarán una propina, recogerán sus abrigos y se irán.”

“¿Es todo?”

“No necesariamente. El mesero llamará al mozo de piso para limpiar la mesa, la preparará y estará lista para los siguientes comensales.”

“Dado que ello no tiene que ver con el cliente, voy a considerarlo dentro de un proceso por separado, aunque sea breve. Quiero hacerle un par de preguntas. La primera ¿Cómo sabe el mesero que la gente ha terminado?”

“Él permanece en su área, y echa una mirada a cada mesa. La experiencia le dicta cuánto le toma a los comensales ingerir sus alimentos, de forma que se puede anticipar a ello cuando esté cerca de una mesa. ¿Otra pregunta?”

“Sí. Dijo que el mesero podría estar en la cocina para hablar con el chef por alguna razón. ¿Cuál sería tal razón?”

“En ocasiones un cliente necesita saber cuánto tardará la preparación de un alimento. En tales casos, el cliente llama al mesero, quien le pregunta al chef. Una vez que se informa, regresa y le responde al cliente.”



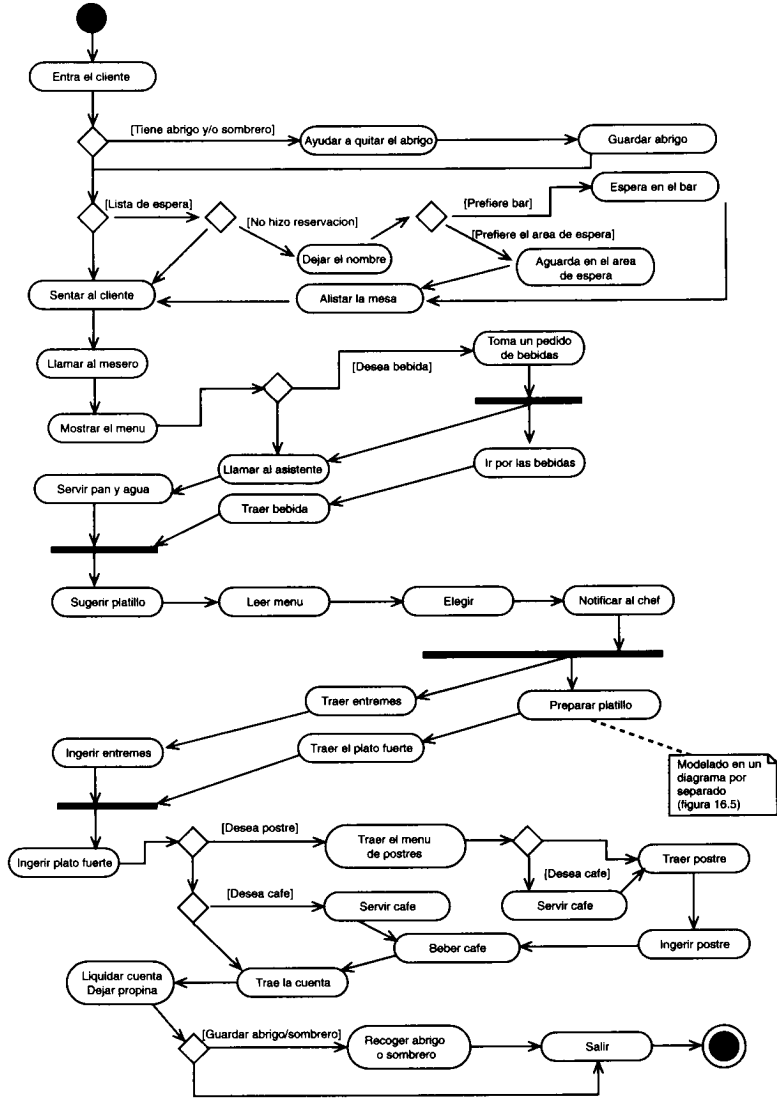
Los entrevistadores se aseguran de agregar cualquier pregunta restante al final.

“¿Sabe? Nunca me había dado cuenta de todo lo que ocurre para servir a un cliente en un restaurante.”

“Es curioso que lo diga. Hasta que me pidió que le describiera los pasos, yo tampoco lo había analizado mucho. Pienso que su diagrama esquematiza todo lo que he dicho, y es una imagen que aclara mis propias ideas.” (Vea la figura 16.3.)

FIGURA 16.3

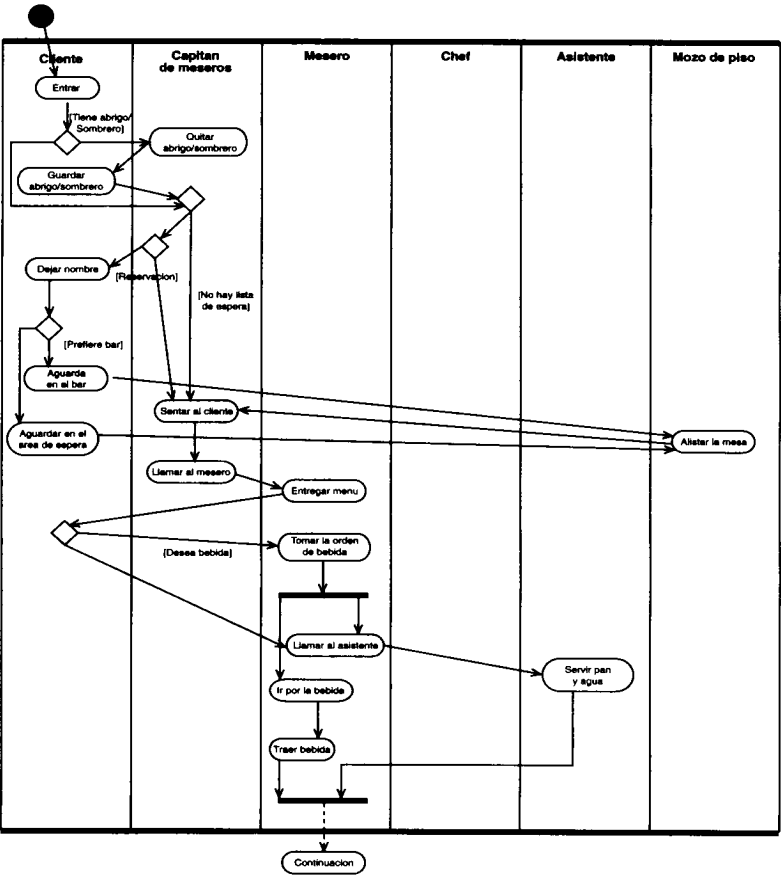
El diagrama de actividades completo para el proceso del negocio del restaurante, “Servir a un cliente”.

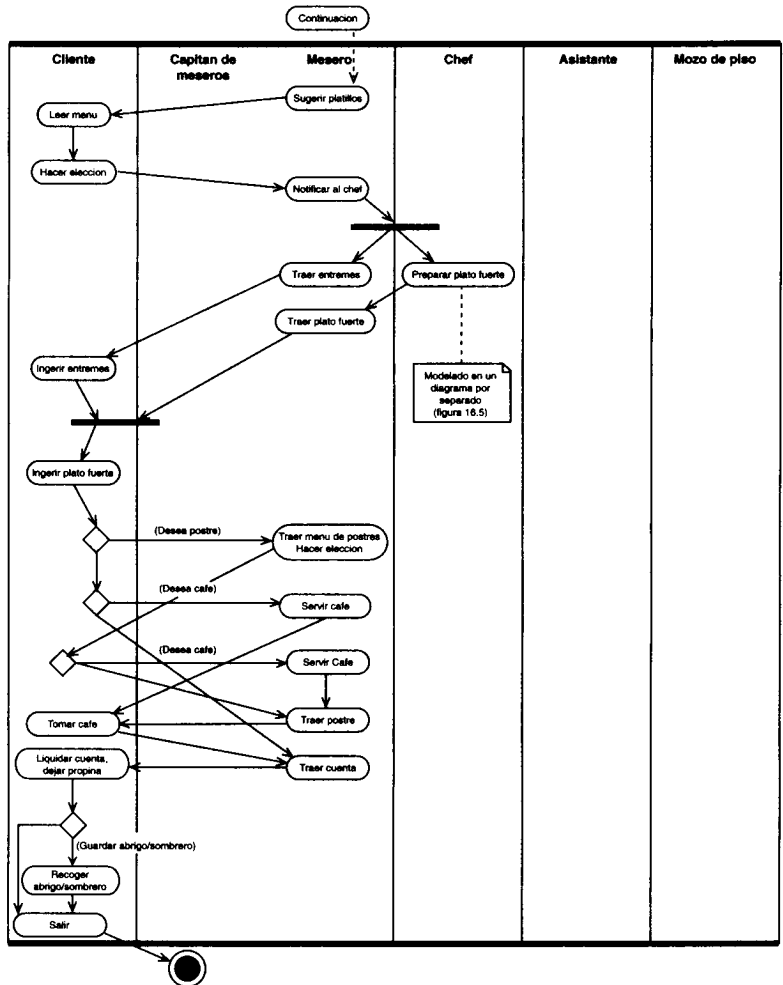


Como lo vio en la hora 11, “Diagramas de actividades”, puede convertir un diagrama de éstos en uno con marcos de responsabilidad. Cuando modele un proceso del negocio, será algo muy útil dado que el diagrama con marcos de responsabilidad le muestra la forma en que cada uno de los responsables figuran en el proceso. La figura 16.4 es un diagrama con marcos de responsabilidad para el proceso del negocio “Servir a un cliente”.

FIGURA 16.4

Un diagrama con marcos de responsabilidad para “Servir a un cliente”.





Preparación de platillos

¿Recuerda que separé los procesos del negocio que aparecieron en la entrevista? Volvamos a reunir al analista con el restaurantero y exploremos el proceso de la “Preparación de platillos”.

“En la plática anterior”, decía el analista, “mencionó que muchos platillos incluyen un entremés antes del plato fuerte, y que la mayoría de las personas preferían el plato fuerte caliente. Indicó el reto de traer el plato fuerte a todos al mismo tiempo y tenerlo caliente; y mencionó la importancia de la coordinación. ¿Podría ahondar?”

“Claro.” dice el restaurantero. “La gente en una mesa casi siempre finaliza sus entremeses, ensaladas o sopas en momentos distintos. Tenemos que coordinarnos para traerles a todos los platos fuertes. La coordinación se realiza entre el mesero y el chef. El chef recibe la comanda del mesero y empieza a preparar los entremeses y el plato fuerte. Cuando se han preparado los entremeses, el mesero va a la cocina, los toma y los lleva a la mesa.”

“¿Y el mesero sabe que ya se han preparado los entremeses porque...?”

“Porque va a la cocina de vez en cuando. Es aquí donde se inicia la coordinación: el chef, luego de dar el entremés al mesero, espera a que éste le avise cuando la mayoría de los comensales ya casi ha terminado con sus entremeses para poderle dar el toque final a cada plato fuerte. El mesero permanece en su área designada, y no pierde de vista a la mesa. En el momento adecuado, el mesero va a la cocina, le indica al chef que los comensales casi están listos para el plato fuerte, y el chef termina su preparación. Un chef experto, que cuente con un grupo de asistentes, equilibrará la preparación de platillos para varios comensales a la vez. La meta es tener listo el platillo principal tan pronto como todos los comensales estén listos para él.”

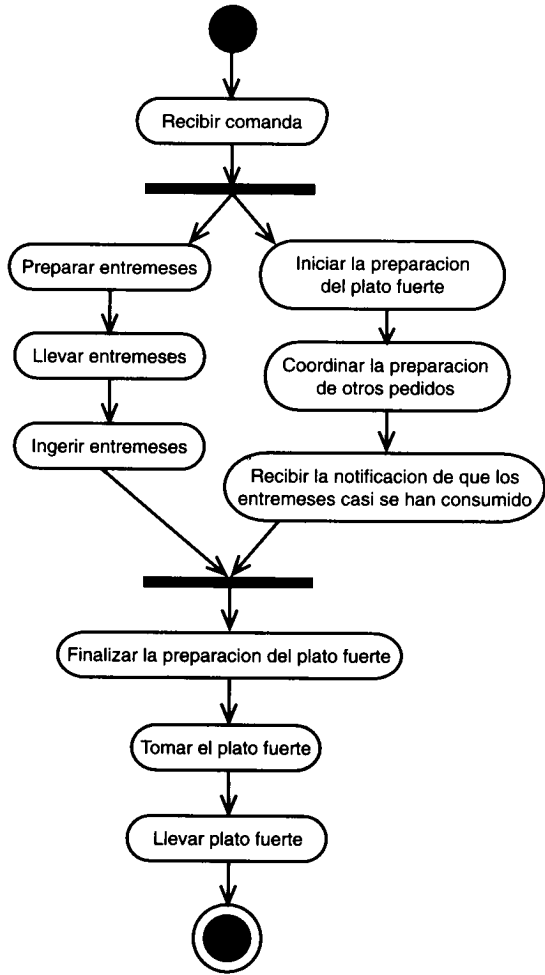
“¿Esto siempre ocurre a tiempo?”

“No, no siempre. Pero con un poco de experiencia y sentido común esto podría ser frecuente. Lo que a veces ocurre es que un comensal lento en un grupo no está del todo listo cuando llevamos el platillo fuerte, pero es un problema menor.”

“Ya entiendo. ¿Qué opina de nuestro diagrama para este proceso?” (Vea la figura 16.5.)

FIGURA 16.5

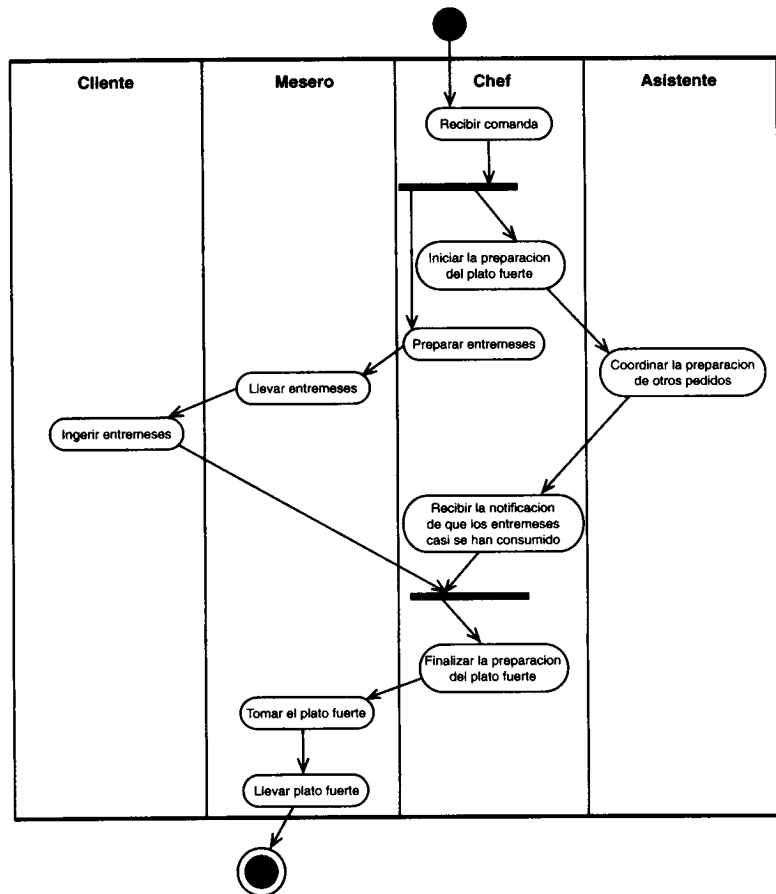
Un diagrama de actividades para "Preparación de platillos".



Como en el caso del proceso del negocio anterior, es adecuado generar un marco de responsabilidades, como el de la figura 16.6.

FIGURA 16.6

Un diagrama con marcos de responsabilidades para “Preparación de alimentos”.



Limpieza de la mesa

“Ahora vayamos a otro proceso por separado, aquél en el que el mozo de piso limpia la mesa”, dice el analista.

“Éste también requiere cierta coordinación. El mesero se asegura de que todos se hayan ido y, luego, llamará al mozo de piso para que se encargue de la mesa. En una noche ajetreada, esto tendrá que hacerse con rapidez. No tenemos tantos mozos de piso como meseros, dado que esto es un proceso casual. Los mozos de piso no siempre están cerca, por lo que tal vez el mesero se vería en la necesidad de buscar uno.”

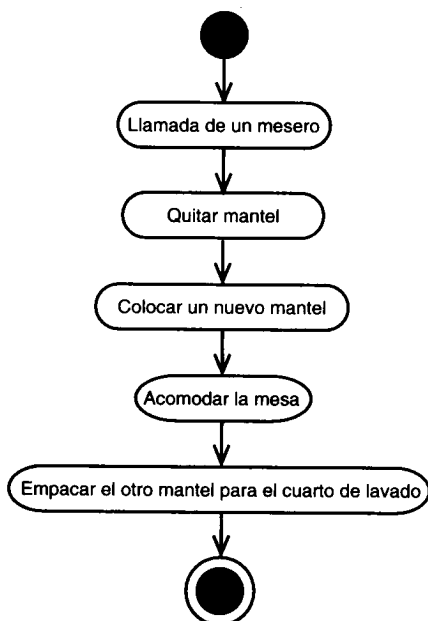
“Creo comprenderlo con ‘encargarse de la mesa’, pero ¿podría ser un poco más específico?”

“Claro. En los restaurantes que yo manejo, tenemos un mantel para cada servicio. Así que el mozo de piso tiene que quitar el mantel usado, amarrarlo, colocar uno nuevo y un nuevo juego de cubiertos en la mesa. Dobla las servilletas y acomoda los cubiertos y un plato para cada asiento de la mesa. Luego se lleva el mantel doblado a una habitación detrás de la cocina. Lo empacamos y lo enviamos al cuarto de lavado al día siguiente.”

La figura 16.7 muestra el diagrama de actividades para este proceso.

FIGURA 16.7

Un diagrama de actividades para “Limpieza de la mesa”.



Lecciones aprendidas

Si usted es aspirante a analista, recuerde las lecciones aprendidas en esta entrevista:

- Se recomienda detenerse y resumir de vez en cuando para verificar su asimilación, practicar con la terminología y hacer que el entrevistado se sienta cómodo.
- Siempre pida que el entrevistado le explique cualquier terminología que no le sea familiar. No se preocupe si ello aparenta falta de conocimiento. La razón por la que está aquí es para adquirirlo y aprender la terminología. Después de todo, va a tener que utilizar el nuevo vocabulario cuando penetre en el análisis del dominio.
- A veces, podrá hacer una pregunta basada en un tema que perciba como respuesta a preguntas anteriores. Mantenga su mente y oídos abiertos para hacer preguntas como éstas. La lógica de negocios con frecuencia emerge de las respuestas.

- Tome nota cuando aparezcan las reglas de la lógica de negocios. Lleve un registro de estas reglas. Podrían ser útiles posteriormente (usted nunca sabe: algún día podría generar una herramienta para la automatización de decisiones que dependa de tales reglas). Claro que debe generarse un registro de esto en la minuta.
- Si siente que cierta parte del proceso se torna complejo, puede optar por sacar tal complicación del contexto hacia un proceso del negocio por separado. Podría facilitar el modelado y el modelo resultante será más claro que si intenta aglomerarlo todo en un solo proceso.
- Obtenga el punto de vista del entrevistado respecto al diagrama de actividades. Haga cualquier modificación que sugiera.

Ha aprendido mucho esta hora y ahora ya cuenta con algunas valiosas técnicas. Conforme obtenga experiencia, generará sus propias técnicas.

En la siguiente hora, comprenderá el análisis de dominios.

Resumen

Esta hora le presentó el escenario para el estudio de un caso que aplicará al UML en un proceso de desarrollo. En el escenario, el consorcio ficticio LaHudra, Nar y Goniff, S. A. decide incorporar la tecnología del cómputo en el restaurante del futuro. Como analista, su trabajo es comprender el proceso del negocio involucrado, comprender el dominio y recopilar las necesidades (acciones presentes en el primer segmento de GRAPPLE).

La recién creada División de restaurantes LNG le da acceso a los expertos del dominio que necesita para comprender los procesos del negocio.

El contenido de esta hora se ha orientado al diálogo en una entrevista y la forma en que podría fluir. Se han intercalado notas que le servirán de guía para saber cómo realizar la entrevista. El objetivo fue el de mostrarle cómo convertir los resultados de la entrevista en un modelo UML.

En la siguiente hora, comprenderá lo relacionado al análisis de un dominio.

Preguntas y respuestas

- P** ¿Siempre se da el caso en que las acciones de un segmento se realicen en el orden en que las listó?
- R** No. En ocasiones podría ser sensato realizarlas en otro orden. Por ejemplo: tal vez quiera descubrir los requerimientos de un sistema antes de identificar a los elementos cooperativos. A su vez, tenga en cuenta que algunas de las acciones no siempre serán necesarias para ciertos proyectos y que algunas de ellas pueden realizarse al mismo tiempo que otras. La “G” en GRAPPLE significa “Guidelines” o “Lineamientos”. Ello no se puede interpretar como “¡Gulp! Es una obligación seguirlo así” o “Debo hacerlo exactamente así”.
- P** ¿Es necesario contar con un solo entrevistador para enterarse de los procesos de negocios de un cliente o un experto? ¿Dos funcionarían mejor que uno?
- R** Por lo general se recomienda que sea una sola persona la que hable con el experto, para que el entrevistado no se sienta como que encara a la inquisición. Podría optar por rotar a los entrevistadores en algún momento de una sesión. El segundo entrevistador podría haber sido el que tomaba las notas, mismo que alternaría su papel con el primer entrevistador.
- P** ¿Hay alguna consideración especial para las notas de la entrevista?
- R** Asegúrese que haya indicado la fecha, hora, lugar y participantes al principio. Nunca sabrá cuándo necesitará tal información y no necesitará tenerla de memoria. A su vez, dentro de las notas intente capturar tanto como le sea posible. Es como un estenógrafo en una corte. Si intenta hacer un boceto conforme avanza, podría estar pasando por alto algo.
- P** ¿No podría omitir algo al intentar tomar nota de todo?
- R** ¡Por supuesto —razón por la cual es mejor contar con dos personas que tomen notas—! Seguramente uno de ellos podría haber anotado algo que el otro no. Recuerde que las notas que tome serán parte del documento que le dará al cliente. Entre más completas estén las notas, será más fácil rastrear la evolución de una idea.

Taller

Para realmente captar todo esto, responda al cuestionario y haga los ejercicios. Las respuestas están en el Apéndice A, “Respuestas a los cuestionarios”.

Cuestionario

1. ¿Cuál diagrama del UML es adecuado para modelar un proceso del negocio?
2. ¿Cómo podría modificar a este diagrama para mostrar qué hace cada quién?
3. ¿Qué debe entenderse por “lógica de negocios”?

Ejercicios

1. Intente aplicar los principios de esta hora a otro dominio. Suponga que LaHudra, Nar y Goniff le han contratado para encabezar un equipo de desarrollo que genere un sistema para su biblioteca corporativa. Inicie el segmento Recopilación de necesidades mediante la asimilación y modelado de los procesos del negocio involucrado. Para ello, tendrá que confiar en su propio conocimiento de las bibliotecas. Haga anotaciones de su solución dado que utilizará este ejemplo en los ejercicios de las siguientes horas.
2. Revise las entrevistas de esta hora. ¿Qué partes de la lógica de negocios se hicieron evidentes?

HORA 17



Elaboración de un análisis de dominio

Ahora continuaremos con el análisis conceptual en el segmento de recopilación de necesidades de GRAPPLE.

En esta hora se tratarán los siguientes temas:

- Análisis de la entrevista del proceso del negocio
- Desarrollo del diagrama de clases inicial
- Agrupación de las clases
- Conformación de asociaciones
- Agregados y objetos compuestos
- Llenado de clases las clases

El primer par de acciones en GRAPPLE se orientan al dominio y no al sistema. Ni la hora anterior ni ésta se orientarán al sistema propuesto. Ciertamente, en lo que hemos visto hasta ahora no se ha propuesto sistema alguno. Sólo tenemos una asignación vaga de LaHudra, Nar y Goniff para valernos de la tecnología y mejorar el acto de comer en restaurantes.

El objetivo de las horas 16 y 17 es comprender el dominio. Lo que significa que tenemos que conocer los procesos específicos que intentamos mejorar y el modo en el que operan tales procesos. Al intentar descubrir los procesos del negocio, hemos empezado a alimentar el conocimiento del equipo de desarrollo en nuestro escenario. Como resultado, los miembros del equipo cuentan con un vocabulario que pueden utilizar para comunicarse posteriormente con la División de Restaurantes de LNG. Esto es de gran importancia dado que el equipo cuenta con un fundamento para aumentar y cultivar su conocimiento en el transcurso del proyecto.

Análisis de la entrevista del proceso del negocio

El equipo de desarrollo tendrá otras entrevistas con los expertos restauranteros, pero antes de ello trabajarán con el contexto de la entrevista del proceso del negocio. El objetivo es producir un diagrama de clases inicial. Un modelador de objetos hará este trabajo con el equipo durante la entrevista o mediante los resultados de ella. En este punto, el modelador buscará sustantivos, verbos y construcciones verbales. Algunos de los sustantivos se convertirán en clases dentro del modelo, y algunos otros en atributos. Los verbos y construcciones verbales podrán convertirse en operaciones o las etiquetas de asociaciones.

Examinemos los resultados de la entrevista de la hora anterior. ¿Qué sustantivos y verbos utilizó el restaurantero?

He aquí los sustantivos:

cliente, abrigo, guardarropa, vale de guardarropa, sombrero, línea de espera, lista de espera, reservación, nombre, bar, bebida, comida, área de espera, mesa, mozo de piso, mantel, capitán de meseros, mesero, servidor, área de servicio, comensal, menú, asistente, charola, pan, mantequilla, vaso, agua, persona, elección de menú, sugerencia del día, restaurante, chef, platillo, cocina, comanda, área de fumadores, formulario, hora, entremés, plato fuerte, postre, menú de postres, café, taza, cuenta, efectivo, tarjeta de crédito, cambio, voucher, propina, cubierto, servilleta, habitación, cuarto de lavado.

Observe que hemos utilizado cada sustantivo en su forma singular.

Los verbos y construcciones verbales son:

tener, ayudar, almacenar, dar, formarse, honrar, sentar, salir, aguardar, surgir, deshacerse, establecer, caminar, llamar, rondar, ver, gesticular, mostrar, preguntar, ordenar, decidir, traer, ir, obtener, finalizar, reservar, rehusar, relatar, pedir, recomendar, alentar, gustar, decir, expresar, mirar, regresar, beber, leer, permitir, seleccionar, atender, obtener un

pedido, servir, recolectar, dejar, limpiar, alistar, anticipar, hablar, venir, convocar, localizar, proveer, preferir, coordinar, recibir, verificar, depender, cuidar, limpiar, asegurarse, encargarse, buscar, quitar, atar, doblar, arreglar, empacar, enviar.

Cuando busquemos los sustantivos y verbos, deberemos procurar incluirlos todos. ¿El modelador los incluirá a todos en el modelo? No. El sentido común indicará a cuáles incluir y a cuáles no. Será de mucha ayuda tener mayor contacto con el restaurantero.

Desarrollo del diagrama de clases inicial

Ahora pongámonos en los zapatos del modelador y empecemos a desarrollar el diagrama de clases. Es aquí donde el sentido común entra en juego. Primero eliminaremos algunos de los sustantivos.

Recordemos en la entrevista que se intentó cambiar “mesero” por “servidor”, pero que no dio resultado. Así que podríamos eliminar uno de estos términos. Tanto el entrevistado como el entrevistador decidieron usar “mesero”, así que eliminaremos a “servidor”. “Cliente” y “comensal” son sinónimos, por lo que podremos eliminar otro sustantivo: nos quedaremos con “cliente”. “Persona” parece ser demasiado genérico, así que también lo eliminaremos.

¿Podríamos eliminar a otros? Algunos sustantivos son más adecuados para usarse como atributos y no como clases. En nuestra lista, “nombre”, “hora” y “reservación” caen en tal categoría. Otro de ellos, “cuarto de lavado”, no es parte física del restaurante, por lo que podríamos eliminarlo.

He aquí el otro lado de la moneda: también es posible agregar una o dos clases. Si analizamos la entrevista, veremos que el restaurantero hizo referencia a “áreas asignadas” y “rotación de meseros”. ¿Quién hace la “asignación” y la “rotación”? Pues un “gerente”, que agregaremos a la lista. Tal clase podría no haber surgido de la entrevista original, dado que el analista se enfocó al cliente, el mesero, el chef y el mozo de piso.



La adición de una clase (como lo verá en la adición de clases abstractas) refleja la evolución del entendimiento conforme avanza el proyecto.

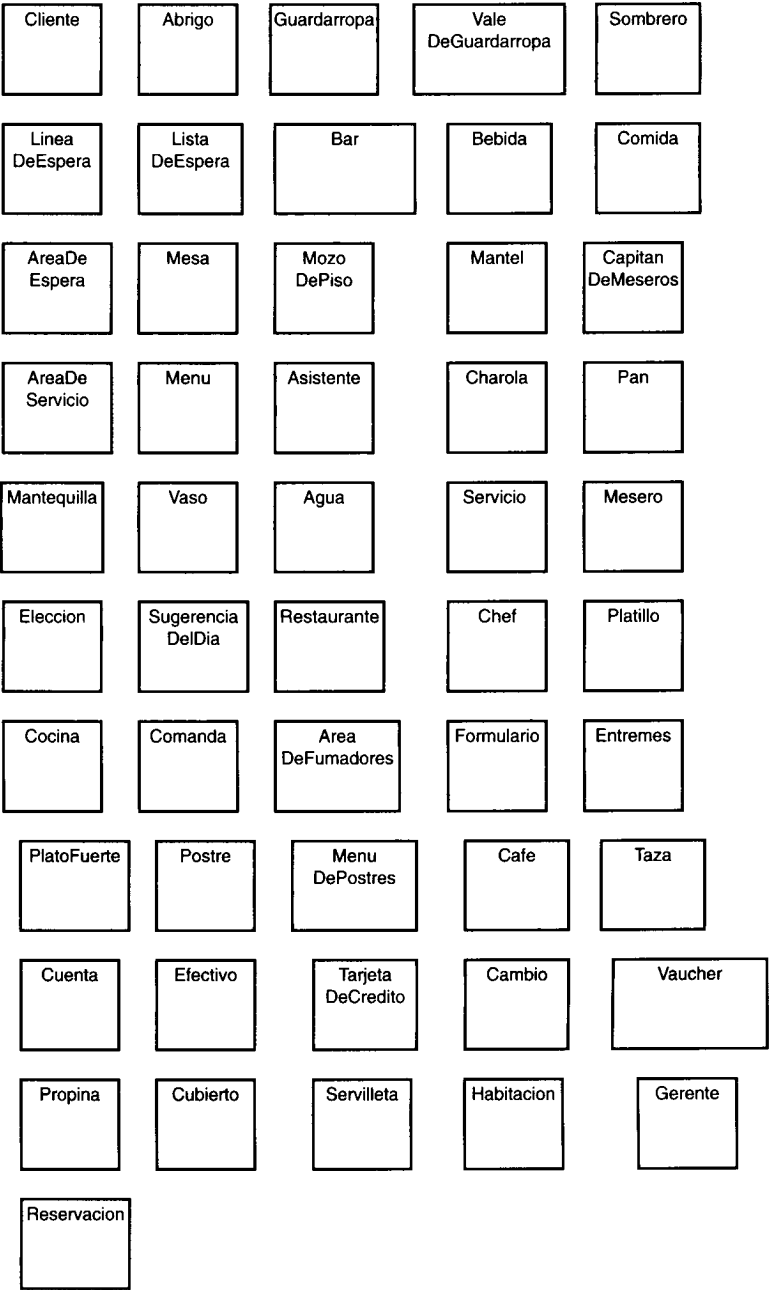
Luego de filtrar los sinónimos y atributos, así como agregar una clase, ahora ésta será nuestra lista de sustantivos que se convertirán en clases:

cliente, abrigo, guardarropa, vale de guardarropa, sombrero, línea de espera, lista de espera, bar, bebida, comida, área de espera, mesa, mozo de piso, mantel, capitán de meseros, área de servicio, menú, asistente, charola, pan, mantequilla, vaso, agua, servicio, mesero, elección, sugerencia del día, restaurante, chef, platillo, cocina, comanda, área de fumadores, formulario, entremés, plato fuerte, postre, menú de postres, café, taza, cuenta, efectivo, tarjeta de crédito, cambio, voucher, propina, cubierto, servilleta, habitación, reservación, gerente.

Nos serviremos de estos sustantivos para generar el diagrama de clases de la figura 17.1, donde pondremos en mayúscula la primera letra de cada nombre de clase. Si tal nombre tiene más de una palabra, las uniremos y pondremos con mayúscula la primera letra de cada palabra. También eliminaremos los acentos.

FIGURA 17.1

El diagrama de clases inicial para el dominio del restaurante.



Agrupación de las clases

Ahora intentaremos conformar algunos grupos significativos. Uno de los grupos consta de personas: cliente, servicio, mozo de piso, capitán de meseros, asistente, chef, mesero y gerente. Este grupo podría tener alguna división pues todos, excepto el cliente y el servicio, son empleados. Así que nos quedaremos con los grupos cliente, servicio y empleado.

Otro grupo consta de elementos relacionados con alimentos: bebida, comida, pan, mantequilla, agua, sugerencia del día, platillo, entremés, plato fuerte, postre y café.

Hay un tercer grupo que consta de utensilios: vaso, cubierto, charola, servilleta y mantel.

El cuarto grupo contiene elementos de transacción: vale de guardarropa, cuenta, efectivo, cambio, tarjeta de crédito, pagaré y propina.

Existe otro grupo que consta de áreas del restaurante: área de espera, área de fumadores, bar, guardarropa, cocina, área de servicio, mesa y habitación. “Habitación” se refiere a aquella que almacena los manteles (y asumimos que otros elementos) que el restaurante enviará a la lavandería. Para hacerlo más descriptivo, llamémoslo “cuarto de lavado”.

Finalmente, podemos agrupar los formularios del restaurante: menú, menú de postres, vale de guardarropa, cuenta y formulario. El último se refiere al que se entrega al chef con la orden, por lo que lo llamaremos “comanda”.

Observe que un par de estos elementos pueden encontrarse en dos grupos (formularios y elementos de transacción). Esto, como veremos, es admisible.

¿Ahora qué haremos con tales grupos? Cada nombre de grupo puede convertirse en una clase abstracta: una que no genera instancias por sí misma, pero que funciona como una clase principal de clases secundarias. Así, la clase abstracta `AreaDeRestaurante` tiene las siguientes clases secundarias: `Bar`, `AreaDeServicio`, `Mesa`, `AreaDeEspera`, `Guardarropa` y `Cocina`.

Podemos modificar el diagrama de clases de la figura 17.1 y producir el de la figura 17.2.

Conformación de asociaciones

Ahora, crearemos y etiquetaremos asociaciones entre algunas de las clases. Los verbos y las construcciones verbales podrán ayudarnos con las etiquetas, y no debemos limitarnos sólo a las de la entrevista. Las etiquetas que sean más descriptivas podrían sobreentenderse.

Una estrategia es la de enfocarnos en algunas de las clases, ver cómo se asocian entre sí, e ir a otro grupo hasta que hayamos desmadejado al conjunto de clases. Posteriormente, generaremos las agregaciones y objetos compuestos. A continuación, incorporaremos los verbos y construcciones verbales como operaciones de las clases.

Asociaciones con el cliente

Empecemos con la clase Cliente. ¿Cuáles clases se asocian con ella? La Reservacion sería una de ellas, y Mesero otra. Algunas otras serían Menu, Alimento, MenuDePostre, Postre, Orden, Cuenta, Propina, Abrigo y Sombrero. La figura 17.3 muestra las asociaciones.

En este punto podemos tomar algunas decisiones. ¿Es necesario incluir Sombrero y Abrigo? Después de todo, nos enfocamos en servir un alimento. Luego de debatir, probablemente concluiríamos que tales clases podrían quedarse en el modelo, porque nuestro interés se centra en todo el proceso de salir a comer. Esto nos hará generar otra clase, EncargadoDelGuardarropa, dado que alguien deberá guardar el abrigo y el sombrero del cliente.

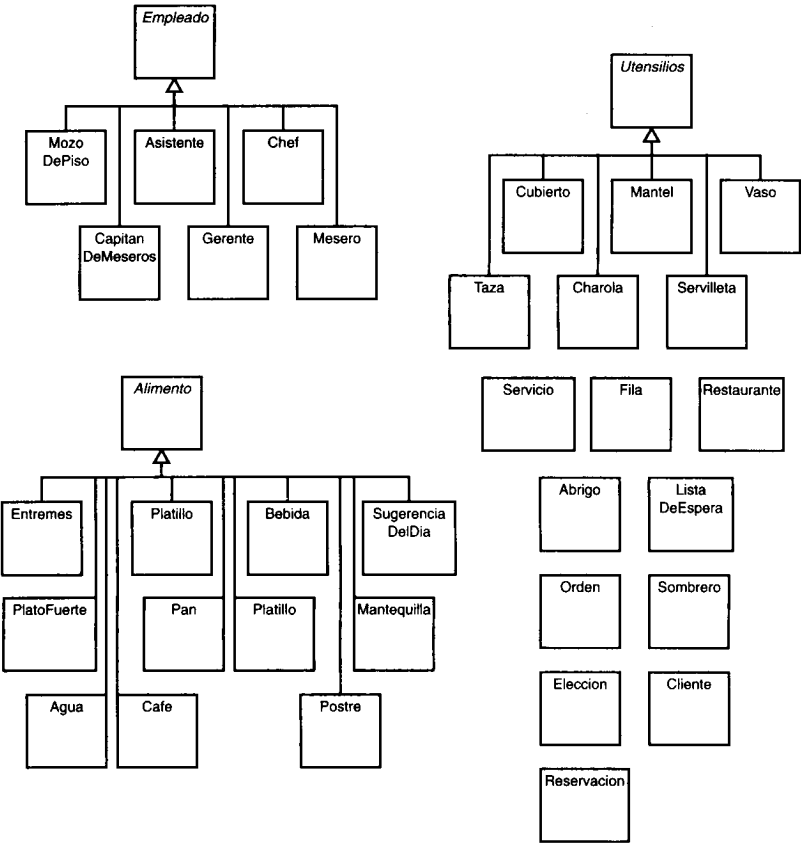
Vamos a etiquetar las asociaciones con algunas frases que las distingan. He aquí algunas de ellas:

- El Cliente hace una Reservacion
- El Cliente es atendido por un Mesero
- El Cliente ingiere un Alimento
- El Cliente ingiere un Postre
- El Cliente hace una Orden
- El Cliente elige de un Menu
- El Cliente elige de un MenuDePostre
- El Cliente liquida la Cuenta
- El Cliente deja una Propina
- El Cliente da a guardar un Abrigo a un EncargadoDelGuardarropa
- El Cliente da a guardar un Sombrero a un EncargadoDelGuardarropa

La figura 17.4 le muestra las asociaciones etiquetadas.

FIGURA 17.2

Las clases abstractas dividen al diagrama de clases en grupos significativos.



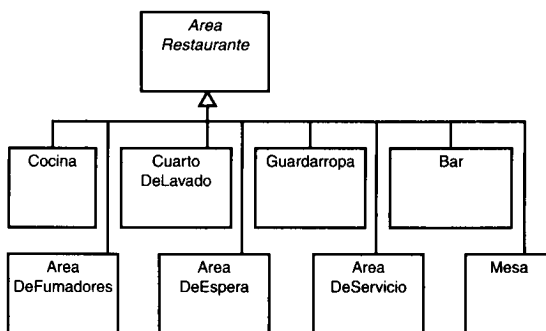
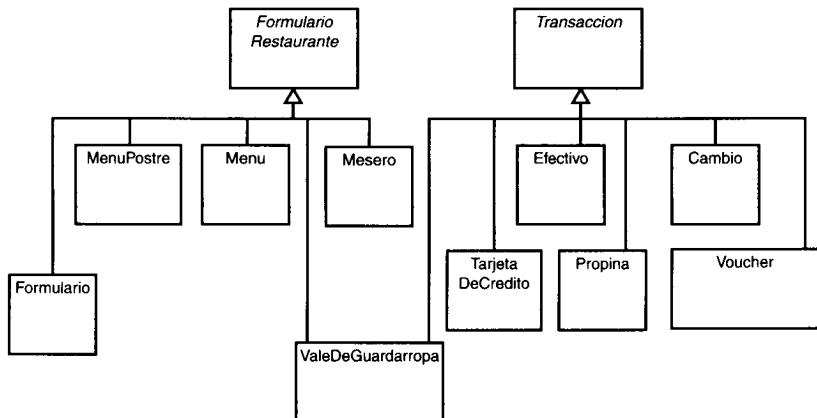


FIGURA 17.3
*Las asociaciones
iniciales a la clase
Cliente.*

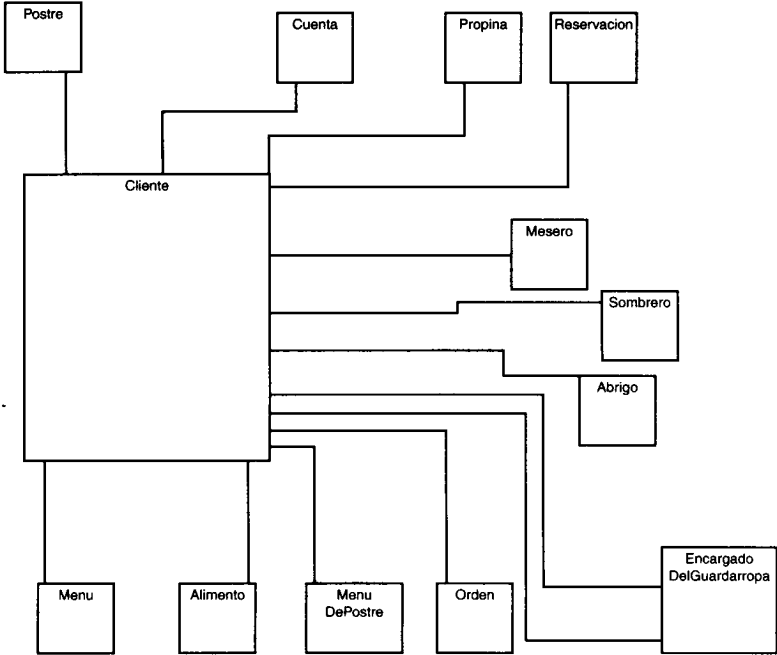
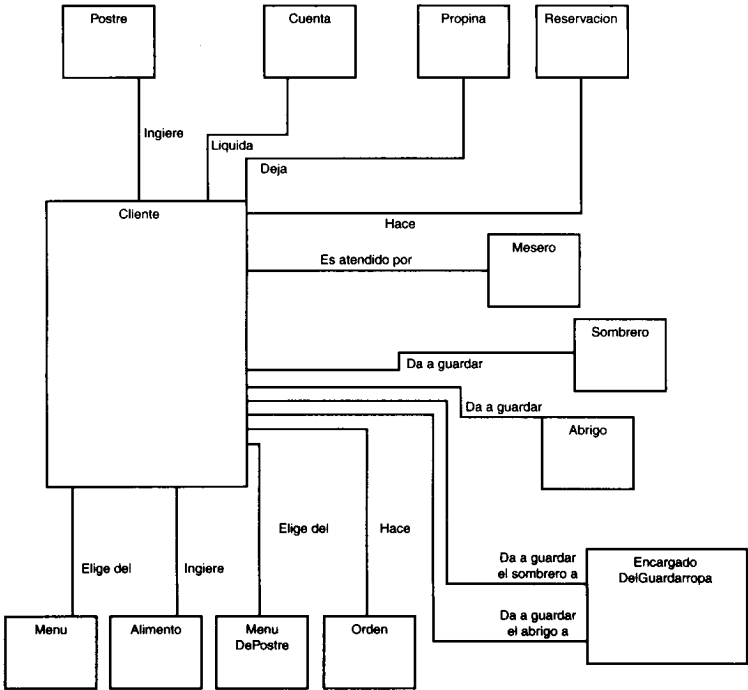


FIGURA 17.4
*Las asociaciones
a la clase Cliente
rotuladas.*



Ahora vamos a enfocarnos en las multiplicidades. Recuerde que una multiplicidad es parte de una asociación: indica cuántas instancias de la clase B se asocian con una de la clase A.

En la mayoría de las frases enumeradas, el Cliente se relaciona con una instancia de otra clase. La segunda frase es distinta de las otras. Tiene una voz pasiva (“es atendido por”) en lugar de una voz activa de otras (como “liquida” o “deja”). Esto sugiere que algo diferente podría ocurrir con tal asociación. Si la invertimos y examinamos la asociación desde el punto de vista del mesero (“El Mesero sirve a un Cliente”), es evidente que un Mesero puede atender a varios clientes.

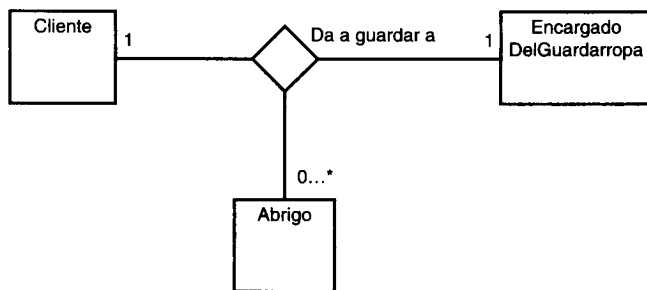
Las dos frases finales apuntan a un tipo de asociación que no habíamos visto antes:

- El Cliente da a guardar un Abrigo a un EncargadoDelGuardarropa
- El Cliente da a guardar un Sombrero a un EncargadoDelGuardarropa

¿Cómo modelaríamos esto?

A este tipo de asociación se le conoce como *tripartita*; esto quiere decir que hay tres clases involucradas. Este tipo de asociación la modelaría mediante la conexión de las clases asociadas con un rombo, y escribiría el nombre de la asociación cerca de él, como en la figura 17.5. En una asociación tripartita, las multiplicidades indican cuántas instancias de dos clases están involucradas cuando la tercera se mantiene constante. En este ejemplo, un Cliente puede dar a guardar más de un Abrigo a un EncargadoDelGuardarropa.

FIGURA 17.5
Una asociación ternaria.



En la siguiente subsección, verá otra forma de manejar esto.

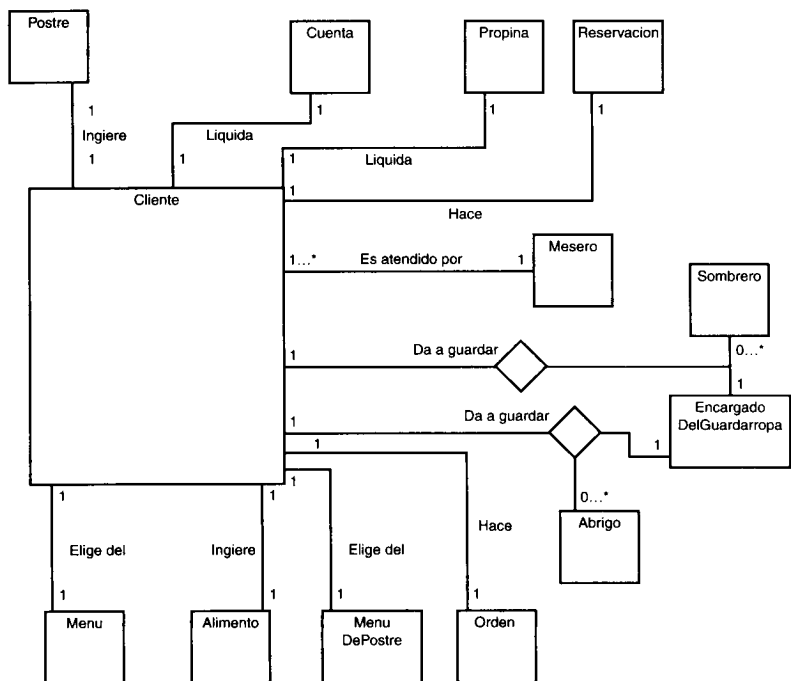


Es posible contar con más de tres clases en una asociación. En nombre de la generalización, el UML la trata como asociación múltiple.

La figura 17.6 le muestra todas las asociaciones del Cliente etiquetadas, donde se incluyen las multiplicidades.

FIGURA 17.6

Inclusión de las multiplicidades en las asociaciones de la clase Cliente.



Asociaciones con el Mesero

Ahora utilicemos la asociación Cliente-Mesero para generar las asociaciones con el Mesero. Una forma de modelar cualquiera de las asociaciones del mesero es tratarlas como tripartitas:

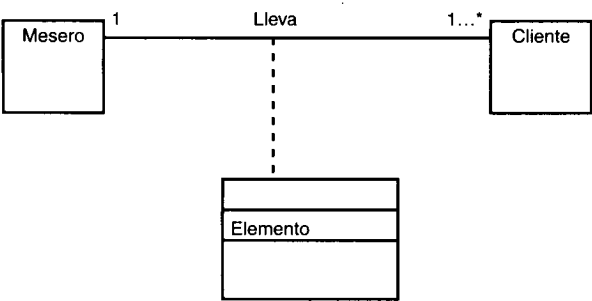
- El Mesero toma una Orden de un Cliente
- El Mesero lleva una Orden a un Chef
- El Mesero sirve un Alimento a un Cliente
- El Mesero sirve un Postre a un Cliente
- El Mesero trae un Menu a un Cliente
- El Mesero trae un MenuDePostre a un Cliente
- El Mesero trae la Cuenta a un Cliente
- El Mesero recoge el Efectivo de un Cliente
- El Mesero recoge la TarjetaDeCredito de un Cliente

Esto, sin duda, desordenará al modelo y lo hará difícil de comprender. Una forma más eficiente es examinar tales asociaciones, utilizar la menor cantidad de etiquetas y adjuntar las clases de asociación adecuadas.

El trabajo de un Mesero es, aparentemente, el de traer y llevar cosas. “Recoger” se sobreentiende como “llevar” y “servir” como “traer”. Etiquetaremos esta asociación de la clase Mesero como “llevar” y “traer”. Adjuntaremos una clase de asociación, y en ella indicaremos lo que se lleva o se trae. Para ello, le daremos un atributo llamado *elemento* y le asignaremos un tipo numérico. Los valores posibles del atributo son los diversos elementos que el Mesero podría llevar o traer.

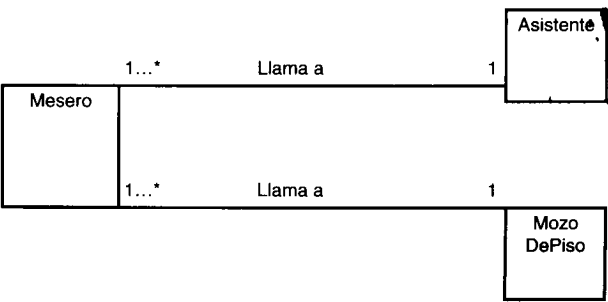
La figura 17.7 le muestra lo anterior una vez puesto en efecto.

FIGURA 17.7
El uso de clases de asociación en las asociaciones del Mesero.



El Mesero también se asocia con un Asistente y un Mozo de piso, como muestra la figura 17.8.

FIGURA 17.8
Otras asociaciones con el Mesero.

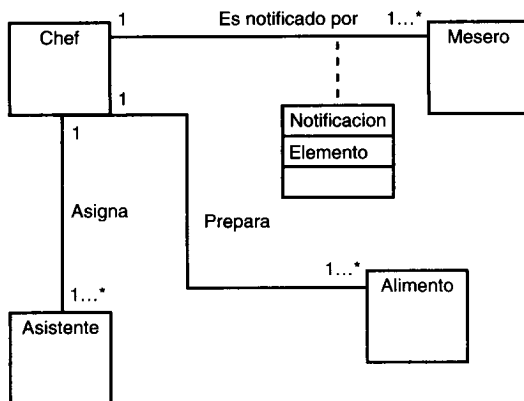


Asociaciones con el Chef

El Chef se asocia con el Asistente, el Mesero y con el Alimento, como en la figura 17.9.

FIGURA 17.9

Asociaciones con el Chef.

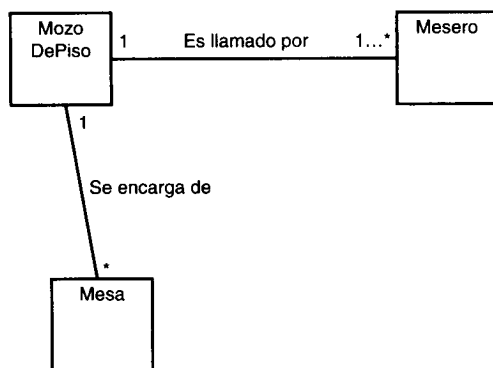


Asociaciones con el Mozo de piso

El Mozo de piso realiza algunas tareas, como se establece en la figura 17.10.

FIGURA 17.10

Asociaciones con el Mozo de piso.



Asociaciones con el Gerente

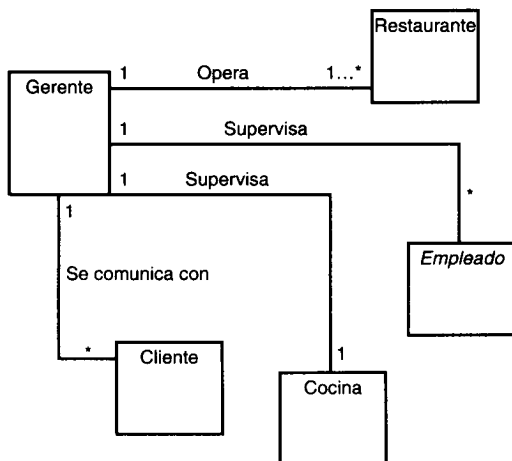
El Gerente es una clase nueva que hemos derivado a partir del análisis del dominio. Esta clase se asocia con muchas otras hacia las cuales hemos desarrollado las siguientes frases:

- El Gerente opera el Restaurante
- El Gerente supervisa a los Empleados
- El Gerente supervisa la Cocina
- El Gerente se comunica con el Cliente

La figura 17.11 modela tales asociaciones.

FIGURA 17.11

Asociaciones con el Gerente.



Una digresión

Algo que podría imaginar es que debiera eliminar los sustantivos que son roles en las asociaciones y contar sólo con una clase genérica, como Empleado. En la asociación, pondría el nombre del rol cerca de la línea de vida adecuada de la asociación.

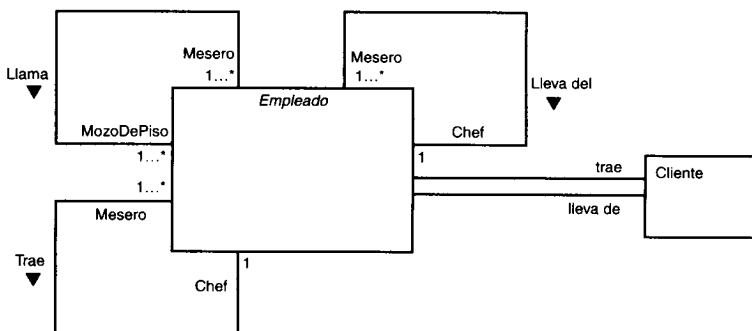
En algunos contextos (como un sistema de nóminas), ello funcionaría bien. En éste tal vez no. Analice las siguientes asociaciones:

- El Mesero trae al Cliente
- El Mesero lleva del Cliente
- El Mesero trae al Chef
- El Mesero lleva del Chef
- El Mesero llama al Mozo de piso

El diagrama luciría como en la figura 17.12.

FIGURA 17.12

Modelado con la clase Empleado.



Como puede advertir, los iconos de clase del diagrama se hacen densos y poco claros, y aún no hemos incluido las clases de asociación.

En todo lo que se relacione con el modelado, deje que le guíe la comprensión.

Formación de agregados y objetos compuestos

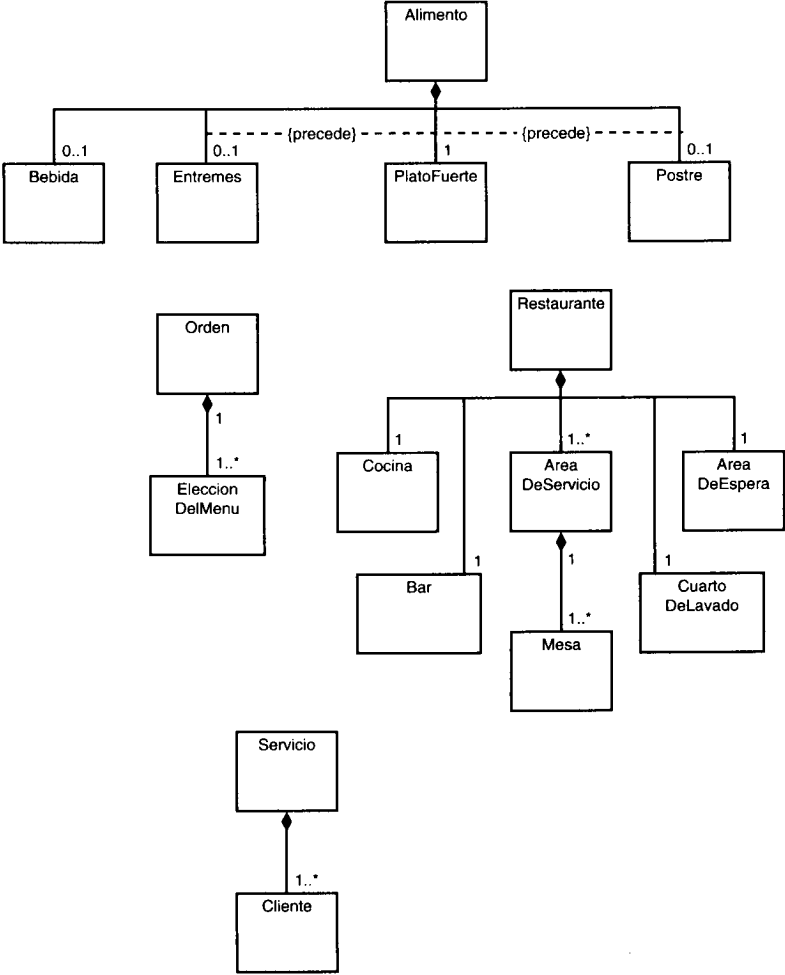
Ya hemos conformado y bautizado clases abstractas y asociaciones, pero hay otra dimensión organizacional. El siguiente paso es localizar clases que sean componentes de otras. En este dominio ello no deberá ser difícil. Por ejemplo, un Alimento consta de un Entremés, un PlatoFuerte, una Bebida y un Postre. El Entremés y el Postre son opcionales. A su vez, los componentes se encuentran en un orden específico que deseamos conservar en nuestro modelo.

He aquí algunas otras composiciones:

- Una Orden consta de una o varias EleccionesDeMenu
- Un Restaurante consta de una Cocina, una o varias AreasDeServicio, un AreaDeEspera, un Bar y un CuartoDeLavado
- Un AreaDeServicio consta de una o varias Mesas
- Un Servicio consta de uno o varios Clientes

En cada caso, el componente es miembro exclusivo de un agregado, de modo que la figura 17.13 los modela a todos ellos como objetos compuestos.

FIGURA 17.13
*Objetos compuestos
en el dominio
Restaurante.*



Llenado de las clases

Las entrevistas y sesiones consecuentes serán muy útiles para dar cuerpo a nuestras clases. Tenga en cuenta que a partir de este momento, un modelador de objetos deberá estar en cada sesión y depurará el modelo al mismo tiempo. Podemos comenzar con la depuración mediante la adición de algunos atributos y operaciones.

Nuestras clases más importantes parecen ser las de Cliente, Mesero, Chef, Gerente y Asistente. Verifique si hay alguna otra clase importante.

El Cliente

¿Cuáles son los atributos obvios para un Cliente? He aquí algunos:

- nombre
- horaLlegada
- orden
- horaServicio

¿Qué hay con las operaciones? Nuestra lista de verbos nos podría servir de guía (aunque no nos deberá limitar). Algunas operaciones del Cliente son:

- ingerir()
- beber()
- estarFeliz (¡bromeaba!)
- ordenar()
- pagar()

La figura 17.14 le muestra la clase Cliente.

FIGURA 17.14

La clase Cliente.

Cliente
nombre horaLlegada orden horaServicio
ingerir() beber() ordenar() pagar()

El Empleado

El Mesero, Chef, Gerente y Asistente son todas clases secundarias de la clase abstracta Empleado. Por ello, asignaremos atributos a Empleado y las clases secundarias los heredarán. Algunos de ellos son:

- nombre
- domicilio
- numeroSeguroSocial
- aniosExperiencia
- fechaContratacion
- salario

Para el asistente, hay cosas que son un poco más complejas. Primero, necesitaremos un atributo llamado *trabajaCon* dado que un Asistente podría ayudar a un Mesero o a un Chef. Este atributo será de tipo numérico.

Las operaciones serán específicas para cada clase secundaria. Para el Mesero, tales operaciones parecen ser adecuadas y pueden verse en la figura 17.15:

- llevar()
- servir()
- recoger()
- llamar()
- verificarEstadoDeLaOrden()

Para el Chef:

- preparar()
- cocinar()
- darPrioridad()
- crearReceta()

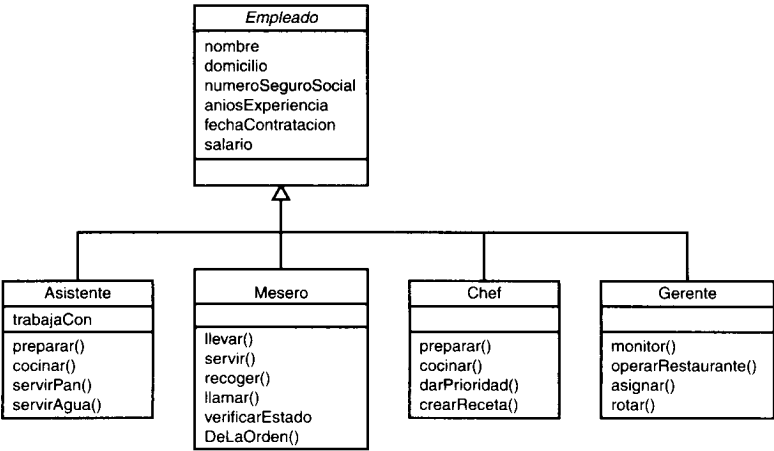
Para el Asistente:

- preparar()
- cocinar()
- servirPan()
- servirAgua()

Las operaciones del Gerente serían:

- supervisar()
- operarRestaurante()
- asignar()
- rotar()

FIGURA 17.15
La clase Empleado y sus clases secundarias.



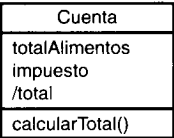
La Cuenta

La Cuenta es, obviamente, una clase importante pues contiene la información del dinero que costó la comida. Sus atributos son:

- totalAlimentos
- impuesto
- total

Dado que total es la suma de totalAlimentos e impuesto, es una variable derivada. Nuestro modelo (vea la figura 17.16) lo refleja. La operación de la Cuenta es calcularTotal(totalAlimentos, impuesto).

FIGURA 17.16
La clase Cuenta.



Detalles generales de los modelos

Hasta este punto ya ha recopilado bastante información. He aquí algunas sugerencias para ayudarle a mantenerlo todo organizado.

Diccionario del modelo

Cuando conjugue los resultados de las entrevistas, procesos del negocio y análisis de dominio, mantenga un diccionario del modelo. Éste es un glosario de terminología en el modelo. Le ayudará a mantener la consistencia y a evitar la ambigüedad.

Por ejemplo: en nuestro dominio del restaurante, el término “menú” es prominente. Este término significa una cosa para un restaurantero y otra para un desarrollador de interfaces gráficas. “Servidor”, el término propuesto pero rechazado, podría tener implicaciones similares: mientras el restaurantero podría pensar en un “mesero”, un ingeniero de sistemas podría pensar en otra cosa completamente distinta. Si tiene definiciones con las que todos estén de acuerdo, o si está al menos consciente de qué palabras podrían causar una gran confusión, evitará muchos problemas subsecuentes. La mayoría de las herramientas de modelado le permiten generar un diccionario conforme cree el modelo.

Organización del diagrama

Otra sugerencia tiene que ver con la organización del diagrama. No es recomendable tener todos los detalles de su modelo de clases en un enorme diagrama. Necesitará un diagrama principal que muestre todas las conexiones, asociaciones y generalizaciones, pero será mejor omitir los atributos y operaciones de él. Podrá enfocarse en determinadas clases si las coloca en diagramas por separado. Normalmente, las herramientas de modelado le permiten organizar sus diagramas mediante una vinculación adecuada entre ellas.

Lecciones aprendidas

¿Qué es lo que hemos aprendido en nuestro análisis de dominio?

- La entrevista del proceso del negocio otorga las bases para el análisis del dominio
- Los sustantivos resultantes pueden convertirse en candidatos para clases
- Hay que eliminar los sustantivos que son atributos, y los que son sinónimos de otros sustantivos de la lista, así como los que representen a clases fuera del ámbito del dominio
- No pierda de vista la oportunidad de agregar clases que podrían no haberse mencionado durante la entrevista del proceso del negocio
- Válgase de algunos de los verbos o construcciones verbales como etiquetas para las asociaciones
- Agrupe las clases y utilice los nombres de los agrupamientos como clases abstractas
- Agrupe clases en agregados u objetos compuestos
- Cambie el nombre de las clases para mayor legibilidad
- Algunas asociaciones pueden ser tripartitas (esto es, que involucran a tres clases)
- Válgase del sentido común para denominar asociaciones y establecer multiplicidades

En la siguiente hora nos iremos del entorno conceptual a los detalles relacionados con el sistema.

Resumen

Esta hora continuó con el análisis conceptual que inició en la hora anterior. La entrevista del proceso del negocio da por resultado los fundamentos para el análisis del dominio. Los sustantivos, verbos y construcciones verbales de la entrevista son los candidatos para el diagrama de clases inicial que definen al dominio Restaurante. El sentido común le indicará cuáles utilizar y cuáles eliminar. Es posible que agregue clases conforme haga el análisis.

El modelador de objetos agregará sustancia a este diagrama mediante la derivación de clases abstractas, asociaciones y multiplicidades. La derivación de agregados u objetos compuestos le ayudará a organizar al modelo. Serán necesarias otras entrevistas y sesiones para dar cuerpo completamente en el modelo, aunque es posible empezar a agregar atributos y operaciones en este punto.

Preguntas y respuestas

P ¿Cómo sabré cuáles clases eliminar de la lista de clases candidatas?

R Mediante el sentido común, elimine los nombres de clases redundantes y esté consciente de los nombres que sean atributos. Elimine los nombres de las clases que estén fuera del ámbito del dominio que está analizando. Recuerde que también podrá agregar clases.

Taller

Este taller verifica la muy importante aptitud para el análisis del dominio, requerida para la creación y desarrollo de un diagrama de clases. Las respuestas se encuentran en el Apéndice A, “Respuestas a los cuestionarios”.

Cuestionario

1. ¿De qué forma utilizaremos los sustantivos obtenidos en la entrevista con un experto?
2. ¿Y de qué forma utilizaremos los verbos y construcciones verbales?
3. ¿Qué es una asociación “tripartita”?
4. ¿Cómo modelaría una asociación tripartita?

Ejercicios

1. Revise las asociaciones tripartitas del cliente con el EncargadoDelGuardarropa. Utilice una clase de asociación para modelarlas de forma eficiente.
2. Si ha seguido con atención la entrevista y el análisis del dominio, tal vez habrá pensado en algunas clases que no aparecieron en ninguna de tales instancias. Una de ellas sería el Cajero. Conforme una asociación entre el Mesero y el Cajero. Utilice una clase de asociación en caso de ser necesario. Si puede pensar en otras clases, incorpórelas en el análisis del dominio.
3. Nuestro objeto compuesto Restaurante incluye sólo a clases “físicas”: áreas como la Cocina y Bar. Tal vez podría decirme que un restaurante también incluye a gente. Revise el objeto compuesto Restaurante e incluya a los empleados en el diagrama. ¿Al incluir a los empleados se convierte al objeto compuesto en un agregado?
4. Además de los atributos y operaciones, apunté en la hora 3, “Uso de la orientación a objetos”, que podría representar la responsabilidad de una clase. En la clase Mesero, agregue un panel de responsabilidades y asígnele una descripción de la responsabilidad propia del mesero.
5. Continúe con el dominio Biblioteca de los ejercicios de la hora 16, “Presentación del caso por estudiar”, y desarrolle un diagrama de clases.



HORA 18

Recopilación de las necesidades del sistema

Las dos horas anteriores trataron de cuestiones conceptuales respecto al dominio. Derivó procesos del negocio y generó diagramas de clases. Ahora iremos al sistema.

En esta hora se tratarán los siguientes temas:

- Cómo vislumbrar el sistema
- La sesión Desarrollo conjunto de aplicaciones (JAD)
- Cómo organizar las necesidades del sistema
- El aprovechamiento de los casos de uso

Los señores LaHudra, Nar y Goniff están impresionados. Han visto el resultado de su equipo de desarrollo y saben que el proyecto va en la dirección correcta. Todos parecen tener una buena comprensión del dominio Restaurante, tanto que los restauranteros de la División de Restaurantes de LNG dicen que los diagramas han aclarado su propio concepto de las operaciones de un restaurante.

Ahora el equipo tendrá que trabajar en la parte medular, es decir, en la parte técnica para el restaurante del futuro. Ya cuentan con los procesos del negocio y los diagramas de clases. Ya pueden empezar a codificar ¿cierto? ¡No! Ni siquiera están cerca de escribir un programa. Antes que nada, tienen que desarrollar una visión del sistema.

La mayoría de los proyectos empiezan con frases como “Generar una base de datos con la información del cliente y hacerla fácil de usar para que los empleados puedan utilizarla con un mínimo de capacitación” o “Crear un entorno de ayuda computarizado que resuelva los problemas en menos de un minuto”. Aquí, el equipo de desarrollo ha comenzado la misión de “usar la tecnología para construir el restaurante del futuro”. Tienen que vislumbrar a este restaurante basado en tecnología para tener una idea de la forma en que el personal del restaurante trabajará en él. Trabajan en un nivel en el que un equipo de desarrollo usualmente no llega, y LaHudra, Nar y Goniff confían en ellos.

El equipo utilizará el conocimiento del proceso del negocio y el recién adquirido conocimiento del dominio para ver en qué lugar la tecnología podría mejorar la experiencia de comer en un restaurante. Escuchemos una reunión del equipo. Los integrantes son un analista, un modelador, un restaurantero, un mesero, un chef y un ingeniero de sistemas. Habrá un moderador en la reunión.

Desarrollo de la idea

Moderador: “Al analizar nuestros diagramas de procesos del negocio, pienso que podemos ver varios lugares donde podría ser útil la tecnología basada en computadoras. Haré una lista en la pizarra. ¿Quién quiere empezar?”



El moderador distribuye copias de la figura 18.1, el diagrama de procesos del negocio para “Servir a un cliente”, y la figura 18.2, el de “Preparar un platillo”.

Analista: “Sí, aparentemente el negocio del restaurante, como casi cualquier otro, depende de la transmisión de la información. Si podemos agilizar tal transmisión (algo en lo que la tecnología se especializa) cumpliremos nuestra meta.”

Restaurantero: “No me queda muy claro. ¿Qué quieren decir con ‘transmisión de la información’? Siempre pensé que mi negocio se centraba en el movimiento de alimentos.”

Ingeniero de sistemas: “Yo le puedo explicar. Cuando el cliente hace una orden, le da información al mesero... por cierto, recordemos que mesero es alguien que sirve las mesas. Una vez que el mesero tiene la información, se la lleva al chef, con ello está transmitiendo información.”

Moderador: “¿Dónde más se transmite información en el diagrama?”

Mesero: “Creo que cuando un cliente me pide que le informe cómo va el avance de su orden y yo le pregunto al chef, ello también puede traducirse como transmisión de información ¿no?”

Analista: “Así es.”

Chef: “Transmisión, tecnicismos. No se ofendan, pero yo no me emociono para nada cuando un mesero entra y me pregunta cuánto más tardaré en terminar de preparar un platillo. Tardará lo que tenga que tardarse y no puedo ser molestado.”

Moderador: “Bueno, tal vez podríamos hacer algo por reducir su molestia. ¿Ven algunos otros puntos de transmisión de información?”



En este caso el moderador intenta suavizar las cosas con el chef para que éste continúe involucrado.

Restaurantero: “¿Qué tal cuando el mesero hace la sugerencia del día? ¿O cuando responde a una pregunta acerca del menú?”

Moderador: “¡Por supuesto!”

Chef: “En ocasiones yo también tengo que dar respuestas. Las personas me envían al mesero para pedirme alguna receta en particular. A veces se las envío mediante el mesero y, si no hay mucho trabajo, yo mismo voy y hablo con el cliente. A ellos les gusta eso.”

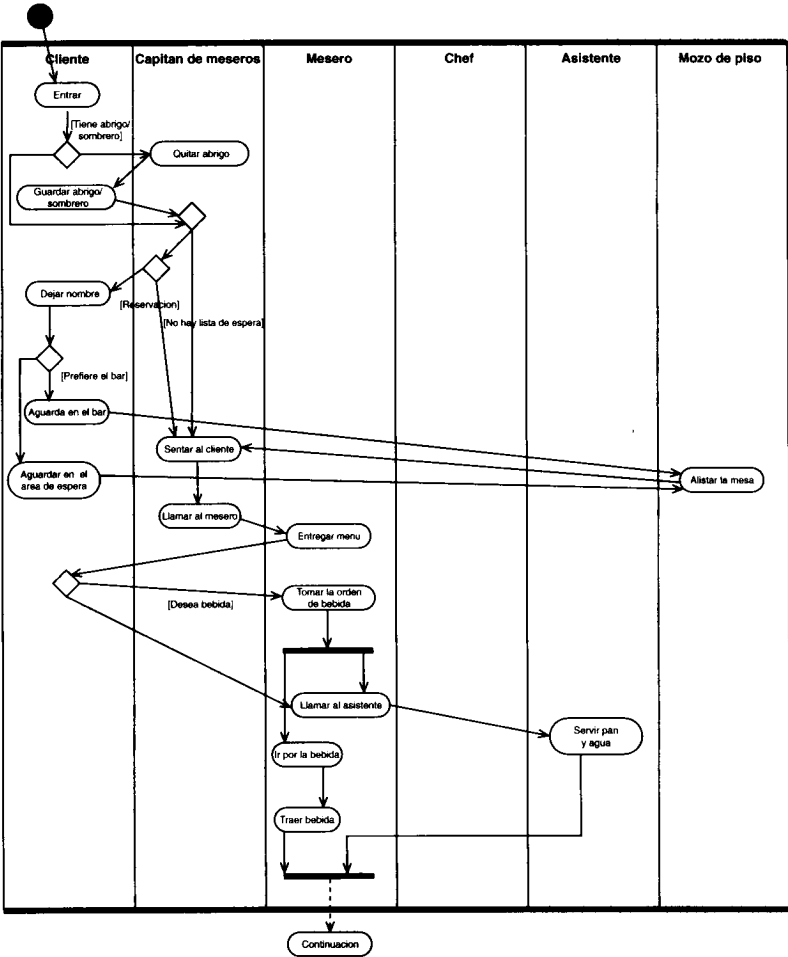
Mesero: “Pues hay cierta información que no me gusta transmitir. Un cliente hace una orden, voy y se la paso al chef, y luego de quince minutos, cuando regreso a la cocina por cualquier otra cosa, oigo que ya se acabaron los ingredientes para tal orden. Yo tengo que ir con el cliente y sugerirle que ordene alguna otra cosa. Generalmente eso molesta mucho al cliente y a mí también, debido a que reduce mi propina.”

Analista: “Tal vez podríamos agregar esto al proceso del negocio...”

Moderador: “Tal vez. Pienso que, si están de acuerdo, podríamos tratarlo en otra reunión.”

FIGURA 18.1

Un diagrama de procesos del negocio para “Servir a un cliente”.



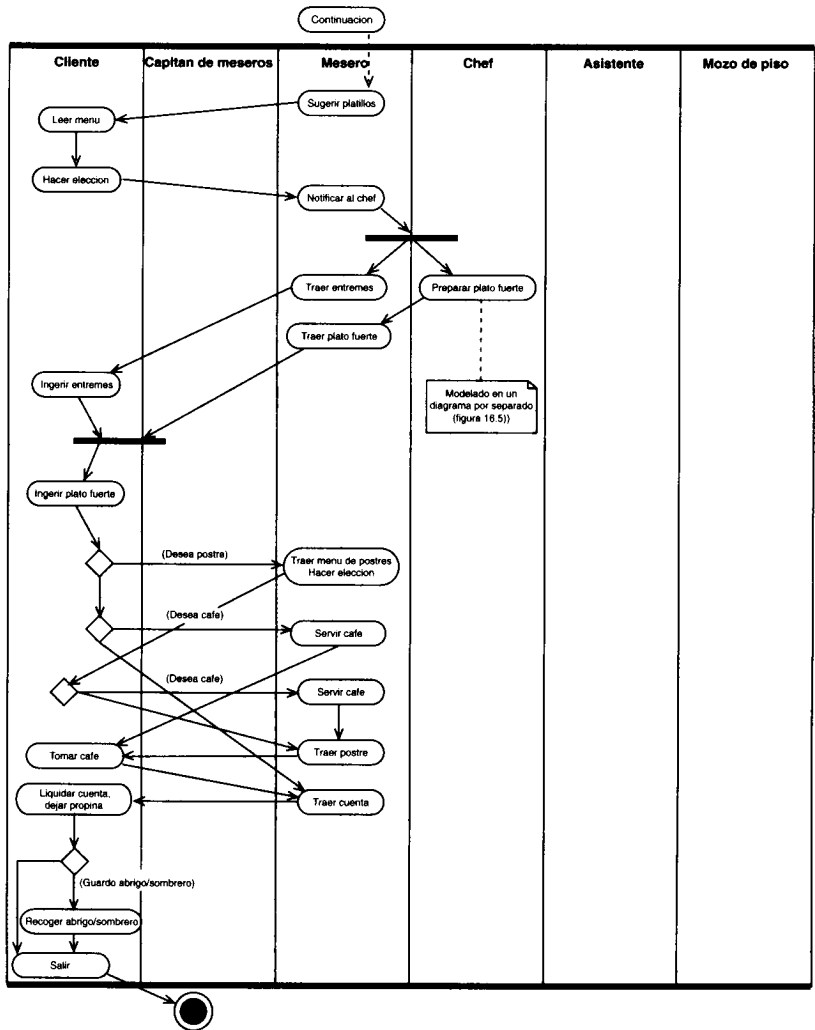
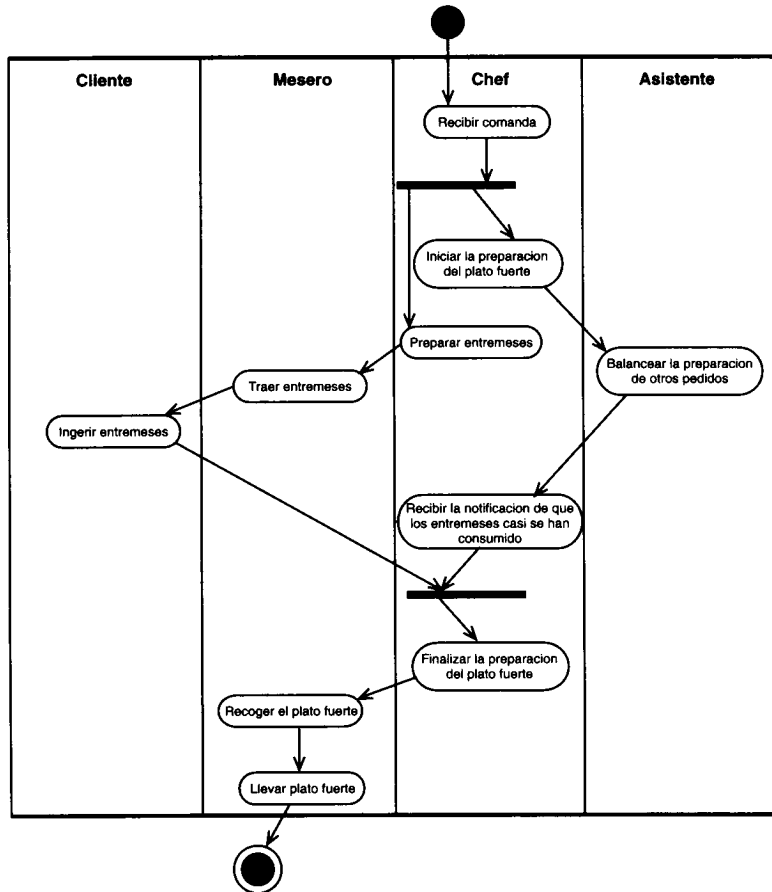


FIGURA 18.2

Un diagrama de procesos del negocio para “Preparación de platillos”.



El moderador intenta mantener bien enfocada la reunión. Observe que el moderador evita utilizar la enojosa construcción “Sí, pero...”.

Analista: “Sí, claro. No sería bueno desviarnos del tema.”

Moderador: “Veamos dónde hemos llegado. De acuerdo con la lista que he hecho, la transmisión de la información se realiza cuando

- El cliente hace una orden
- El mesero lleva la comanda al chef
- El cliente pide al mesero el grado de avance de su orden
- El mesero da la sugerencia del día

- El mesero responde a una pregunta de algo en el menú
- El chef responde a inquietudes respecto a alguna receta.”



Es muy adecuado que, en la entrevista del proceso del negocio, el moderador haga una pausa para resumir.

Analista: “Sé que no está en ninguno de nuestros diagramas de negocios, pero ¿qué acaso el cliente no tiene preguntas sobre algo en la cuenta? Cuando el mesero le da la respuesta, también se tiene transmisión de información.”

Moderador: “Correcto. ¿Algo más de los procesos del negocio?”

Ingeniero de sistemas: “Creo que sí. ¿Qué hay de la famosa coordinación entre el mesero y el chef? Es decir, ¿en qué momento saben que debe servirse el plato fuerte una vez que los clientes han terminado sus entremeses? También, esto es información que se transmite.”

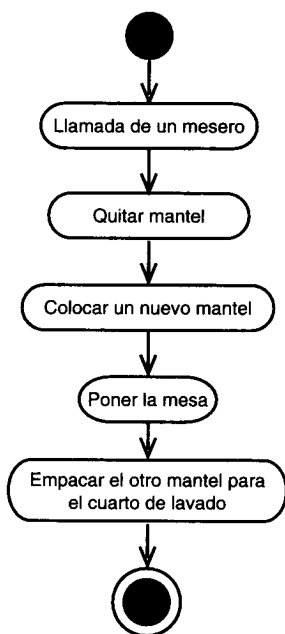
Analista: “De acuerdo. La información se transmite de distintas formas en este caso.”

Restaurantero: “Sólo nos dio dos diagramas de procesos del negocio y recuerdo que creamos otro.”

Moderador: “Así es. Aquí está el de ‘Limpieza de una mesa’” (vea la figura 18.3).

FIGURA 18.3

Un diagrama de procesos del negocio para “Limpieza de una mesa”.



Analista: “Parece que aquí sólo hay un caso de transmisión de información , pero apuesto que es muy importante: El mesero llama al mozo de piso para hacerle saber que es hora de limpiar la mesa.”

Restaurantero: “Esto es muy importante. No puede sentar a un nuevo grupo de personas en la mesa si no está lista. Si la limpieza no se realiza tan rápido como sea posible, tendremos muchos clientes hambrientos, y molestos, amontonados en el bar y en el área de espera.”

Modelador: “He ido modificando mis diagramas de clases mientras transcurre la reunión. ¿Puedo hacer una pregunta? ¿Sería bueno que nuestro sistema (como sea que luzca) nos permitiera evaluar nuestra eficiencia en conjunto para atender a nuestra clientela?”

Restaurantero: “¡Claro! De esa forma sabríamos dónde y cómo mejorar. ¿En qué está pensando?”

Modelador: “En nuestra clase Cliente hay un atributo horaLlegada y otro horaServicio. Quisiera agregar un atributo derivado que se llame tiempoEspera, que podría ser la diferencia entre horaLlegada y horaServicio. ¿Qué opinan?”

Restaurantero: “Me parece bien. Así sabremos qué tal estamos en cuestiones de eficiencia.”

Analista: “Así es. Tendrán muchos datos con los que podrán jugar, por ejemplo, tiempoEspera como una función de la hora del día, o como una función de cuántos meseros estaban laborando en ese momento o cosas como esas...”

Modelador: “Hay otra posibilidad. ¿Y si agregamos otro atributo llamado horaSalida, y un atributo derivado llamado duracionComida que sería la diferencia entre horaServicio y horaSalida?”

Moderador: “Parece ser una buena sugerencia. La agregaré. ¿Alguna otra?”

Modelador: “Ya que estamos con los atributos de tiempo, ¿qué tal si agregamos algunos atributos de tiempo a las clases Mesero y Chef que indiquen al gerente el tiempo que se tardan en realizar el trabajo?”

Restaurantero: “Huy, no... La idea de supervisar el rendimiento de los empleados no es muy agradable para ellos y para mí tampoco. No porque sean perezosos (no lo son), sino porque no es agradable sentirse acosado por una especie de capataz. Es mejor que todos se sientan a gusto en el trabajo; de esta manera, el restaurante será mejor y los clientes también se sentirán más a gusto.”

Chef: “Sí, estoy de acuerdo. Como lo dije, cuando se prepara un platillo se tardará lo que tenga que tardarse. No quiero tener un manojito de comandas y tener un capataz que me diga que tengo que tardarme cuatro minutos y medio menos en preparar una trucha a la almandina.”

Mesero: “Y yo no quiero oír que me he tardado mucho en llevar el menú de postres cuando los clientes hayan terminado con sus alimentos. Hay muchas cosas involucradas.”

Modelador: “Bien, bien, idea descartada. De hecho, ya que lo mencionan, creo que debo quitar ‘supervisar’ como operación de la clase Gerente. Entretanto, he aquí la forma en que luce la clase Cliente ahora.” (Vea la figura 18.4.)

FIGURA 18.4

La clase Cliente actualizada.

Cliente
nombre
horaLlegada
orden
horaServicio
/tiempoEspera
horaSalida
/duracionComida
ingerir()
beber()
ordenar()
pagar()



Las ideas del modelador muestran que se encuentra modificando constantemente los diagramas de clases.

El debate entre el modelador, el restaurantero y el mesero dejan entrever un punto crucial: es necesario que las personas del negocio participen en el desarrollo del sistema. Sin las opiniones tanto del restaurantero como del mesero, el proyecto de desarrollo podría haber gastado tiempo y dinero en instaurar algunas características para la supervisión del rendimiento que no servirían a posteriori. Los empleados podrían haber reaccionado de manera negativa, lo que causaría repercusiones en el sistema y, eventualmente, en el restaurante.

Moderador: “Por lo que he escuchado, parece que podemos distinguir dos tipos de agilidad. Uno de ellos involucra la velocidad con que se transmite la información, y la otra a la rapidez con que cada empleado realiza una tarea. El sentir del grupo deja entrever que la segunda es una molestia, pero la primera no. ¿Estoy en lo correcto?”

(Todos asienten).

Analista: “Bien, ya que estamos de acuerdo en ello, ¿podríamos ver algunas otras ideas respecto a lo que el sistema haría específicamente?”

Moderador: “Claro. ¿Alguna idea?”

Mesero: “Cuando transmito toda esta información, camino mucho en el transcurso de una tarde. En ocasiones tengo que trabajar en un área que está lejos de la cocina. El caminar tanto se lleva tiempo —sin contar las suelas de mis zapatos—.”

Analista: “Bien, pues parece que tendremos que hacer algo para eliminar, o al menos reducir, el tiempo muerto. Con ello podríamos agilizar la transmisión de la información.”

Moderador: “¿Tiempo muerto?”

Analista: “Sí. Nuestro sistema debe evitar que los meseros caminen de más. Es obvio que tendrán que ir a la cocina para llevar la orden a los clientes, pero ¿y qué tal si logramos que sea a lo único que vayan a la cocina? ¿Y suponga que pudieran ir a ella justo a tiempo para traer la orden?”

Ingeniero de sistemas: “Pues creo que ya nos estamos concentrando en algo importante. Se me ocurre que podríamos establecer una pequeña red que conecte a los meseros con la cocina, incluso con los mozos de piso. Así la información se transmitiría con mucha rapidez.”

Analista: “No quisiera sonar demasiado analítico aquí pero... ¿Una pequeña red? Tendrán que brincar cables para llegar a las terminales. En lugar de caminar a la cocina, tendrán que buscar una computadora. A mí me parece que es como matar moscas a cañonazos. ¿Qué nos ahorraría?”

Ingeniero de sistemas: “Tal vez nada, como lo has descrito. Incluso hasta el servicio podría empeorar. Pero así no era lo que yo pensaba.”

Analista: “¿Entonces?”

Ingeniero de sistemas: “Bueno. Suponga que cada mesero y mozo de piso llevan consigo una terminal. Una pequeña, del tamaño de la palma de su mano. Y también supón que no usamos cables. Tendríamos una terminal de escritorio en la cocina y otra en la oficina del gerente.”

Analista: “Ah, ya veo... El sistema del que hablas podría resolver muchas cosas. Por ejemplo, si el cliente hace sus órdenes, el mesero podría marcarlas en su computadora de mano y se irían a una terminal en la cocina. Ello eliminaría el paso, y los pasos, de caminar del área de servicio a la cocina.”

Mesero: “Me gusta. ¿Qué tal si el cliente casi ha acabado sus entremeses y se lo indico a la cocina apretando algo en la computadora que dicen? Ello me evitará el tener que ir y decirle al chef que termine de preparar el plato principal.”

Chef: “Y así veré el mensaje en la cocina. De hecho, todos mis asistentes podrían ver el mensaje al mismo tiempo, y podríamos tener los mensajes en una, dos o tres pantallas grandes. No tendré que ver cuál asistente estaba cocinando qué platillo ni decirles qué tanto deberían de haber avanzado. Ellos podrían tomar tal responsabilidad por sí mismos.”

Ingeniero de sistemas: “Y cuando terminen la orden, podrían enviar un mensaje al mesero para hacérselo saber. No tendrá que estar regresando a la cocina para verificar.”

Mesero: “Huy, pues suena muy bien. Incluso podría enviarle una señal a un mozo de piso para que limpie una mesa. No tendré que buscar a uno. Ello podría agilizar mucho las cosas.”

Restaurantero: “¿Cómo podrían ustedes hacer esto?”

Ingeniero de sistemas: “No se preocupe de ello por el momento.”

Moderador: “Bueno, entonces nuestro sistema será una red de área local inalámbrica con computadoras palmtop para los meseros y los mozos de piso, y con computadoras de escritorio en la cocina y en la oficina del gerente. Pero nos estamos olvidando de algo.”

Analista: “¿De qué?”

Moderador: “De un nombre para el sistema.”

Chef: “¿Qué tal ‘EL EXPERTO CHEF’?”

Moderador: “¿Y tiene algún significado en particular? ¿Forman alguna sigla?”

Chef: “No. Sólo me gustó.”

Analista: “¿Qué tal Red Interactiva Inalámbrica para Restaurantes? Podríamos llamarla RIIR.”

Moderador: “Bueno... No lo sé.”

Ingeniero de sistemas: “Bueno, tal vez podría ser Red Inalámbrica para la Comunicación Organizada. ¡Miren! Y ello deja las siglas RICO, que sería muy adecuado para el servicio en un restaurante.”

Chef: “Me gusta.”

Analista: “A mí también. El nombre deja entrever que el sistema nos enriquecería a nosotros, y tiene un significado muy directo.”

Moderador: “Entonces, ¿Estamos de acuerdo con RICO? Bien, pienso que hemos terminado por el momento.”

Preparación para la recopilación de las necesidades

El equipo pasa los resultados de su reunión a los funcionarios corporativos. LaHudra no puede creer en su buena fortuna al explorar una nueva área. Nar está completamente abrumado. Goniff ve visiones de signos monetarios que bailan ante sus ojos. Los tres le dicen al equipo que prosiga.

Ahora que el equipo tiene una idea del sistema, ¿podrán iniciar su trabajo los programadores y los ingenieros de sistemas? Por supuesto que no. El equipo deberá adaptar el sistema RICO a las necesidades de los usuarios, no a la tecnología. Aunque ya cuentan con algunas ideas de la reunión del equipo, aún no han expuesto el concepto de RICO a un grupo de empleados y gerentes para obtener información e ideas desde su punto de vista como usuarios.

La siguiente acción GRAPPLE hace exactamente eso. En una sesión de Desarrollo conjunto de aplicaciones (JAD), el equipo obtendrá y documentará las necesidades del sistema. Con esto, podrán hacer algunas estimaciones de tiempo y dinero.

La sesión JAD se realiza en una sala de conferencias. Encabezada por un moderador, se denomina sesión “conjunta” dado que incluirá a miembros del equipo de desarrollo y a usuarios potenciales del sistema, así como a expertos del dominio. Los miembros del equipo de desarrollo en esta reunión son dos analistas que toman notas, un modelador, dos programadores y un ingeniero de sistemas. Los usuarios potenciales son tres meseros, dos chefs, dos restauranteros y dos mozos de piso.

El objetivo de esta reunión es producir un diagrama de paquetes que muestre los principales segmentos de funcionalidad del sistema. Cada paquete representará un segmento y contendrá casos de uso que detallarán de qué se trata cada segmento de funcionalidad.

Vamos a la sesión.

La sesión JAD de necesidades

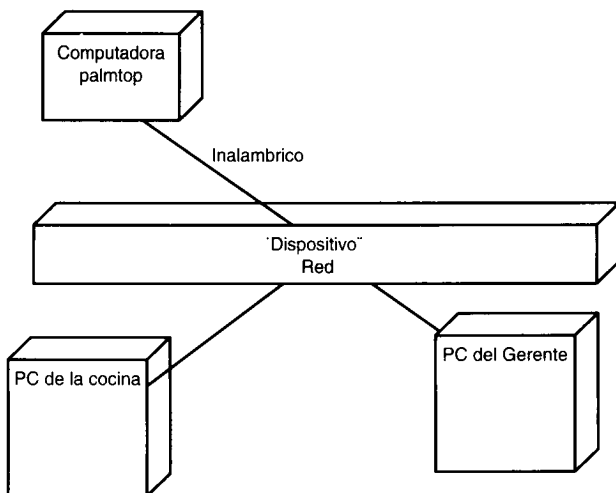
Moderador: “Para empezar, quiero agradecerles a todos haber venido a nuestra sesión. Estas sesiones nos pueden llevar algo de tiempo, pero también pueden ser muy divertidas. Lo que intentamos hacer es conocer las necesidades para un sistema llamado RICO: Red Inalámbrica para la Comunicación Organizada.”

“El concepto de RICO es muy directo. La forma en que lo imaginamos es que los meseros tendrán en su poder una computadora palmtop y la utilizarán para comunicarse con la cocina y sus mozos de piso. Estos últimos también tendrán computadoras de este tipo y las utilizarán para comunicarse. La cocina tendrá una terminal de escritorio y una o varias pantallas. El gerente también tendrá una en su oficina. He aquí una figura de lo que estoy hablando.” (Vea la figura 18.5.)

“Esperamos instalar a RICO en todos los restaurantes LNG, y queremos ayudarles a hacer su trabajo. Para lograrlo, necesitamos que nos digan qué es lo que desean que haga el sistema. Es decir, si el sistema ya estuviera en funcionamiento ¿qué querrían que hiciera?”

FIGURA 18.5

El sistema RICO.



“Haremos esta pregunta una y otra vez. Al final de la sesión, tendremos un conjunto organizado de necesidades con las que todos estaremos de acuerdo. Piensen que es su carta a Santa Claus. Nos basaremos en esa carta para crear un proyecto que los programadores utilizarán para generar el sistema. Hay algo que quisiera que tuvieran en cuenta: necesitamos la comprensión e ideas de cada uno de ustedes, no importa cuál sea el nombre de su puesto.”

Analista1: “¿Podríamos empezar por imaginarnos cuáles serían los principales segmentos de funcionalidad?”

Moderador: “Sí. ¿Están de acuerdo los miembros del grupo?”

Restaurantero2: “Si, mire... Yo no estuve en los debates anteriores, pero creo que esto es una buena idea. ¿Podríamos organizarlo de acuerdo a, por decir, las áreas del restaurante? Es decir, las áreas de servicio tienen ciertas necesidades, la cocina otras, el área de espera otras...”

Moderador: “Es una posibilidad.”

Analista2: “Hum... Al ver los diagramas de procesos del negocio yo ya veo una organización.”

Programador1: “¿Perdón?”

Analista2: “Por tarea. El chef tiene que realizar ciertas cosas, el mesero otras, y así.”

Moderador: “Suena bien. ¿Estaríamos de acuerdo en organizarlo por tarea?”

(Todos asienten).

Moderador: “¡Bien! En los diagramas de procesos del negocio y de clases, las tareas que tenemos son Mesero, Chef, Mozo de piso, Asistente y Gerente.”

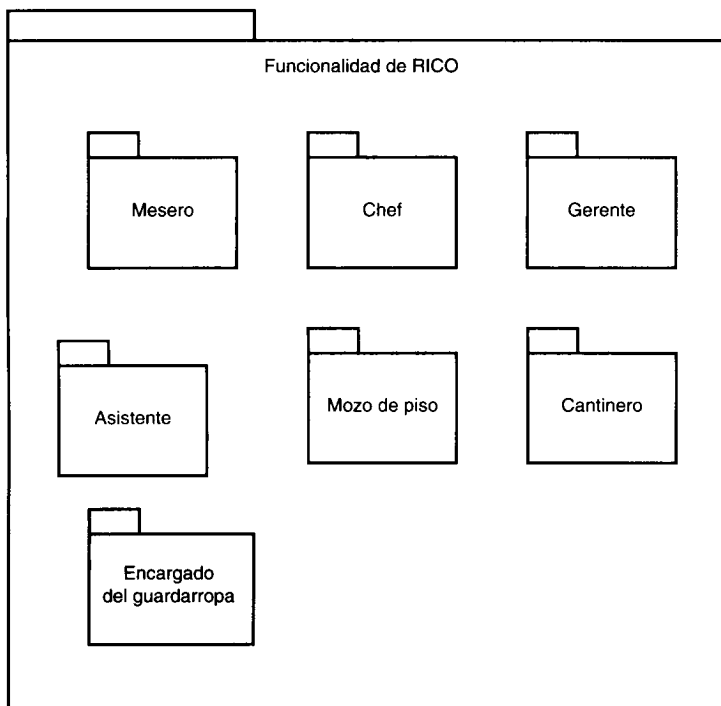
Restaurantero2: “¿No olvidaron algunas? No están el encargado del guardarropa y el cantinero.”

Restaurantero1: “¡Vaya! ¿Cómo se me pudieron olvidar?”

Moderador: “Los agregaré a nuestra lista, y utilizaré los símbolos de paquetes del UML para estar al tanto.” (Vea la figura 18.6.)

FIGURA 18.6

Los paquetes de funcionalidad de RICO.



Modelador: “Ya estoy en ello. Agregué información a nuestros diagramas de clases. Ya estaba la clase EncargadoDelGuardarropa. Lo que agregué fue la clase Cantinero.”

Restaurantero2: “Me pregunto qué hace en su computadora laptop. ¿Me podría mostrar sus, hum, ‘clases’?”

Modelador: “Sí, claro. Aquí están.” (Vea la figura 18.7.)

FIGURA 18.7

Las clases
EncargadoDelGuardar
ropa y Cantinero.

EncargadoDel Guardarropa	imprimirVale
guardarAbrigo() guardarSombbrero() imprimirVale	tomarOrdenDeBebida() prepararBebida() imprimirCuentaDeBar

Restaurantero2: “Muy interesante. Espero que en algún descanso se dé tiempo de explicarme lo que significa.”

Moderador: “Ahora que contamos con las piezas principales, ¿alguien prefiere empezar con alguna tarea en particular?”

Mesero1: “¿Qué tal con la del Mesero?”

Moderador: “Me parece adecuado. Bien, ¿qué tipo de funcionalidad quisiera ver en este paquete? Recuerden, miembros del grupo, que aunque tratemos una tarea que no sea la de ustedes, pueden participar. Todas las opiniones son bienvenidas.”

Mesero2: “Quisiera poder tomar una orden en mi pequeña computadora y enviarla a la cocina.”

Moderador: “Muy bien. ¿Qué más?”

Mesero1: “Quisiera saber el grado de avance de una orden.”

Chef2: “¿Puedo indicarle a un mesero cuando ya esté lista su orden?”

Moderador: “Sí y sí. Estos conceptos, como ven, los estoy escribiendo dentro de elipses rotuladas. Las llamaré ‘casos de uso’. Les pediremos a algunos de ustedes que regresen y que nos ayuden a analizar tales casos de uso, pero ello será en otra reunión.”

El resultado

La sesión JAD continuó por el resto del día. Cuando los participantes terminaron, ya contaban con un conjunto de necesidades que aparecieron como casos de uso organizados en los paquetes.

Para el paquete mesero, los casos fueron:

- Tomar una orden
- Transmitir la orden a la cocina
- Cambiar una orden
- Recibir una notificación de la cocina

- Llevar un control del progreso de la orden
- Notificar al chef el progreso de los clientes en sus alimentos
- Totalizar una cuenta
- Imprimir una cuenta
- Llamar a un asistente
- Llamar a un mozo de piso
- Tomar una orden de bebida
- Transmitir una orden de bebida al bar
- Obtener un acuse de recibo
- Recibir una notificación del bar

Para el paquete chef, los casos de uso fueron:

- Almacenar una receta
- Recuperar una receta
- Notificar al mesero
- Recibir una petición del mesero
- Dar acuse de recibo a una petición del mesero
- Indicar el periodo de preparación
- Asignar una orden

Los casos de uso del mozo de piso fueron:

- Recibir una petición del mesero
- Dar acuse de recibo a una petición
- Indicar que una mesa ha sido limpiada

Los casos de uso para el asistente fueron:

- Recibir una petición del mesero
- Recibir una petición del chef
- Dar acuse de recibo de una petición
- Notificar que la petición se ha completado

Para el cantinero:

- Capturar una receta de bebida
- Obtener la receta de una bebida
- Recibir una notificación del mesero

- Recibir una petición del mesero
- Dar acuse de recibo de una petición
- Notificar que la petición se ha completado

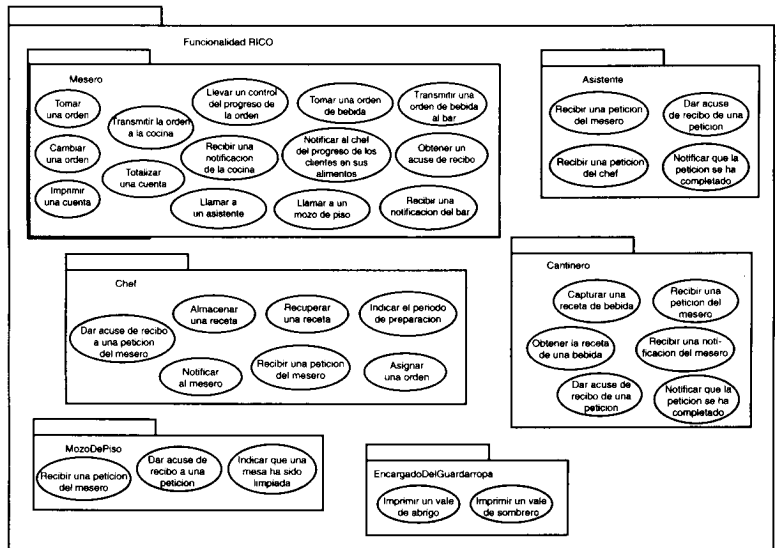
Y para el encargado del guardarropa:

- Imprimir un vale de abrigo
- Imprimir un vale de sombrero

La figura 18.8 muestra la forma en que todo lo anterior luce en el UML.

FIGURA 18.8

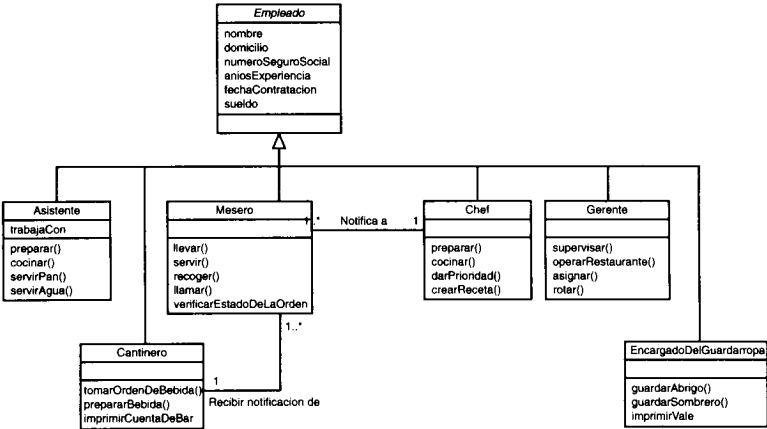
El diagrama de paquetes de funcionalidad.



El modelador hará el desarrollo ulterior de los diagramas de clases mediante la adición de dos clases y asociaciones como en la figura 18.9.

FIGURA 18.9

La información de las clases recién agregada.



¿Ahora qué?

El documento de diseño que entregará el equipo a su cliente crece a pasos agigantados. Ahora incluye procesos del negocio, diagramas de clases y un conjunto de paquetes de funcionalidad.

¿Ahora empezará a codificar el equipo? En absoluto. En la siguiente hora empezarán a analizar el contenido de los paquetes.

Resumen

En el contexto de la reunión, el equipo de desarrollo ha generado una visión del sistema de cómputo para el restaurante del futuro. Los miembros del equipo decidieron que la agilización en la transmisión de la información es la clave para el éxito del sistema, y traen a colación tipos de tecnología para lograrlo.

En una sesión JAD, el equipo de desarrollo se reúne con usuarios potenciales y expertos de dominio para obtener los requerimientos del sistema. El resultado es un diagrama de paquetes en donde cada paquete representa una sección principal de funcionalidad. Los casos de uso dentro de un paquete se basan en tal funcionalidad.

Preguntas y respuestas

- P ¿Podría ser que algunos de los participantes de la sesión JAD sean los mismos que participaron en la reunión anterior en equipo?**
- R** Sí, y, de hecho, es recomendable. Tales personas podrían recordar detalles vitales que podrían no ser muy claros en las minutas.
- P Vi que los señores LaHudra, Nar y Goniff no participan en tales reuniones. ¿Podría alguien de tal nivel tomar parte en las reuniones y las sesiones JAD?**
- R** Estos en particular no. Sin embargo, en algunas empresas, los altos directivos podrían participar de manera activa al menos en parte de una sesión. Es difícil que un alto directivo tome parte en toda una sesión JAD.
- P ¿Siempre se dará el caso de que se tenga que organizar al sistema por tareas, como lo hicimos en este dominio?**
- R** No, no siempre. Aquí se hizo así porque fue conveniente para este dominio. De hecho, podríamos buscar una forma alternativa de hacerlo si nos concentramos en ello. Otros tipos de sistemas podrían requerir un trato distinto. Por ejemplo: un área de asistencia podría tener Recepción de llamadas, Resolución de problemas y Regreso de llamadas como paquetes. Nuevamente, dentro de cada paquete, tendrá un conjunto de casos de uso.

Taller

Verifique su conocimiento en la recopilación de necesidades y localice las respuestas en el Apéndice A, “Respuestas a los cuestionarios”.

Cuestionario

1. ¿Cómo hemos representado las necesidades del sistema?
2. ¿Una vez que se hace el análisis del dominio ya finaliza el modelado de clases?
3. ¿Qué es el “tiempo muerto”?

Ejercicio

Continúe con el dominio de la Biblioteca. ¿Cuáles son los principales paquetes de funcionalidad? ¿Cuáles son los casos de uso que los componen?

HORA 19



Desarrollo de los casos de uso

Ahora que ha visto lo que está involucrado en la recopilación de necesidades, veremos el análisis de tales necesidades y las llevaremos a cabo en un sistema.

En esta hora se tratarán los siguientes temas:

- Cuidado y provisión de casos de uso
- La especificación de descripciones, condiciones previas y resultantes.
- La especificación de pasos
- La diagramación de los casos de uso

Los casos de uso del diagrama de paquetes de la hora 18, “Recopilación de las necesidades del sistema”, nos dio una buena idea de lo que el sistema tendrá que hacer. El equipo tendrá que llevar a cabo cada uno de ellos. Poco a poco han pasado de comprender el dominio a comprender el sistema. Los casos de uso establecieron el puente.

Si cree que el proyecto del desarrollo del sistema se orienta a los casos de uso, ya habrá comprendido todo el proceso.

Observe que en ningún momento de la sesión JAD el equipo de desarrollo habló de la forma en que el sistema realizaría todas las actividades indicadas en los diversos casos de uso. La idea fue la de enumerar todos los casos de uso posibles. Conforme se realicen los casos de uso en esta hora, verá la forma en que los componentes del sistema RICO empezarán a materializarse. En este punto del desarrollo, el sistema empieza a ser el centro de atención.

Vamos a ponernos en los zapatos del equipo de desarrollo y veremos algunos de los casos de uso obtenidos.

Cuidado y provisión de los casos de uso

Para analizar a los casos de uso, tendremos que realizar otra sesión JAD. El debate en esta sesión se orienta a derivar un análisis para cada caso de uso.

A modo de advertencia: la sesión JAD de los casos de uso es, por lo general, la más compleja, dado que pide a los participantes (usuarios potenciales del sistema terminado) que se conviertan en analistas. En su propio nicho, cada uno es un experto del dominio, y tendrá que profundizar en su experiencia. Por lo general, no están acostumbrados a expresar o analizar su conocimiento. De hecho, también es probable que tampoco hayan formado parte de algún proyecto de diseño, y que no se sientan muy a gusto al intentar especificar lo que un sistema debería hacer para ayudarles en su trabajo.

Para reducir estas presiones, es mejor organizar la sesión JAD para que el equipo trate con un grupo a la vez; por ejemplo: sólo los meseros. De esa forma, los demás no estarán ociosos mientras los meseros analizan sus casos de uso. Quienes conocen las generalidades del dominio, los restauranteros, podrían dar una mano en todos los grupos. Una selección de usuarios sería adecuada cuando se trate el tema del paquete del Cliente.

Son muchos los casos de uso, y para no extendernos mucho en esta hora, nos enfocaremos en los primeros nueve de ellos, propios del paquete Mesero. Luego de que vea cómo se harán tales análisis, podrá hacer el resto de los casos de uso de los Meseros, así como los de los demás paquetes.

El análisis de los casos de uso

Por lo que se mencionó anteriormente (hora 7, “Diagramas de casos de uso”), recuerde que, cada caso de uso es una colección de situaciones, y cada una de éstas es una secuencia de pasos. Para cada escenario en cada caso de uso, queremos mostrar:

- Una breve descripción del escenario
- Conjeturas del escenario

- El actor que inicia el caso de uso
- Condiciones previas para el caso de uso
- Pasos relacionados con el sistema en el escenario
- Condiciones resultantes una vez terminado el escenario
- El actor beneficiado del caso de uso



En su análisis también podrá incluir cualquier condición excepcional o alternativas. No obstante, los presentes escenarios se han mantenido simples.

Ninguna forma específica es “correcta” para establecer un análisis de casos de uso. Los elementos que he listado establecen, por lo general, un panorama del caso de uso.



En su documento de diseño (el que le da a su cliente y programadores), cada uno de estos análisis de casos de uso deberán estar en páginas por separado. Posiblemente quiera incluir un diagrama del caso de uso, con todo y actores, en esta página.

Los pasos, en el escenario, que están relacionados con el sistema son muy importantes. Mostrarán la forma en que se supone que funcionará el sistema. Cuando los participantes de la sesión JAD nos digan los pasos, de hecho nos estarán diciendo cómo va a lucir el sistema. Después de esta sesión JAD, deberíamos tener una firme idea de los componentes del sistema.

También es importante hacer conjeturas. En la lista de conjeturas, podrá indicar algunas consideraciones de diseño, como lo verá posteriormente.

Esto era lo que quise decir cuando dije que el proyecto sería “orientado a casos de uso”. Los casos de uso crearán, a fin de cuentas, la ruta de acceso al sistema.

El paquete Mesero

La clase Mesero se distingue por ser la de mayor actividad, lo cual no debería sorprendernos, dado que el mesero interactúa prácticamente con las demás clases.

Los casos de uso del Mesero son:

- Tomar una orden
- Transmitir la orden a la cocina
- Cambiar una orden
- Recibir una notificación de la cocina
- Sondear el progreso de la orden
- Notificar al chef del progreso de los clientes en sus alimentos
- Totalizar una cuenta
- Imprimir una cuenta
- Llamar a un asistente
- Llamar a un mozo de piso
- Tomar una orden de bebida
- Transmitir una orden de bebida al bar
- Obtener un acuse de recibo
- Recibir una notificación del área del bar

Tomar una orden

Empecemos con “Tomar una orden”. Vamos a preguntarle a los meseros experimentados, para que nos den una descripción, conjeturas, condiciones previas, pasos y condiciones resultantes. El paquete y subpaquete ya indica al actor que inicia (Mesero) y al beneficiado (Cliente).

Una buena descripción en una sola frase podría ser: “El mesero captura la orden del cliente en la palmtop y la transmite a la cocina.” Se asume que un cliente desea un platillo, que ha leído el menú y que ya hizo su elección. También se puede asumir que la palmtop del mesero cuenta con una interfaz para capturar órdenes.

Las condiciones previas son que un cliente se haya sentado y haya leído el menú. La condición resultante es que tal pedido se haya capturado en RICO.

Los pasos en el caso de uso son:

1. En la palmtop, el Mesero activa la interfaz para capturar una orden.
2. Aparece la interfaz para capturar órdenes.
3. El Mesero registra la selección del Cliente hecha del menú en RICO.
4. El sistema transmite la orden a la PC de la cocina.

Aunque hemos asumido que ya hay una interfaz para registrar órdenes, aún no hemos especificado cómo luciría o cómo sería el proceso físico de registrar la orden. Todavía no sabemos cómo lucirá la interfaz de la PC de la cocina, ni hemos dicho nada de los detalles técnicos de transmitir una orden.

El punto es que, conforme ponemos de manifiesto nuestras conjeturas en cuanto al diseño, empezamos a tener una idea de lo que se supone que debe hacer el sistema y empezamos a dar forma a nuestras ideas para ponerlo en práctica. Los pasos en los casos de uso nos fuerzan a asumir la idea acerca de los componentes del sistema. Recuerde que los casos de uso pretenden mostrar como se ve el sistema para un usuario.

Transmitir la orden a la cocina

¿Listo para otro? Éste será incluido en (esto es, “usado por”) al menos dos casos de uso: la anterior y “Cambiar una orden”.

La descripción es: “Tomar una orden que ha sido registrada en la palmtop, colocarla en la red inalámbrica y enviarla a la PC de la cocina”. Se asume que ya tenemos medios para comunicar la orden (mediante una red inalámbrica) y, nuevamente, que contamos con una interfaz para registrar una orden. ¿Tenemos que repetir esta conjetura? Sí. Cada caso de uso aparecerá eventualmente en una página por separado en el documento de diseño, lo que servirá como una referencia del sistema. Para mayor claridad, las conjeturas deberían aparecer en cada caso de uso, aun si tenemos que repetirlas en cada uno de ellos.

La condición previa es una orden registrada en una palmtop. La condición resultante es que la orden llegue a la cocina. El actor beneficiado es el Cliente.

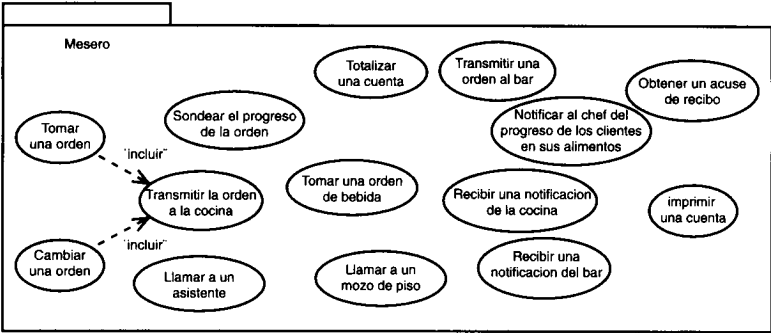
Los pasos son:

1. Hacer clic en un botón en la interfaz para el registro de órdenes indica “Enviar a la cocina”.
2. RICO transmitirá la orden mediante la red inalámbrica.
3. La orden llega a la cocina.
4. La interfaz del usuario para registrar las órdenes en la palmtop indica que la orden ha llegado a la cocina.

Claro está que tendremos que modificar nuestro diagrama de casos de uso para el subpaquete Cliente. Tiene que mostrar la dependencia «incluir» entre este caso de uso y “Tomar una orden”, y entre este caso de uso y “Cambiar una orden”. La figura 19.1 muestra los diagramas actualizados del caso de uso para el paquete Mesero.

FIGURA 19.1

Los diagramas actualizados de casos de uso para el paquete Mesero.



Cambiar una orden

Dentro de este contexto, vayamos a “Cambiar una orden”. La descripción es “Modificar una orden ya registrada en RICO”. Se asume que ya se ha registrado y enviado a la cocina una orden y que, subsecuentemente, el cliente desea cambiarla. También asumiremos que RICO cuenta con una base de datos de órdenes que muestra al mesero quién ha capturado cada orden y la mesa de donde provino, que el mesero podrá acceder a tal base de datos en la palmtop, que RICO puede transmitir y recibir de la palmtop a la PC de la cocina, y que la palmtop tiene una interfaz de usuario capaz de cambiar una orden.

La condición previa es la orden registrada con anterioridad. La condición resultante es que la orden modificada haya llegado a la cocina. El actor beneficiado es el Cliente.

Los pasos en este caso de uso son:

1. En la palmtop, el mesero activa la interfaz del usuario para cambiar una orden.
2. La interfaz del usuario trae una lista de órdenes realizadas y enviadas a la cocina por el mesero.
3. El mesero selecciona la orden por cambiar.
4. El mesero registra la modificación a la orden.
5. El sistema transmite la orden a la PC de la cocina.

El paso 5 incluye al caso de uso anterior “Transmitir la orden a la cocina”.

Sondeo del progreso de la orden

Como podrá recordar, los debates anteriores respecto al restaurante del futuro incluyeron la posibilidad de descubrir en qué momento la orden de un cliente saldría de la cocina. Este caso de uso hace exactamente eso. Si se instaura en el sistema se facilitará enormemente el trabajo del mesero.

La descripción es: “Sondear el progreso (el tiempo para completar) de una orden que ya se haya registrado en RICO”. Se asume que ya se ha realizado, registrado y enviado una orden a la cocina, y que el cliente desea saber cuánto más esperará para que llegue su platillo. Repetimos dos conjeturas de diseño anteriores: una base de datos de pedidos y la facultad de transmitir mensajes entre la palmtop y la PC de la cocina. También asumimos una interfaz del usuario en la palmtop para el sondeo de las órdenes y otra en la PC de la cocina con el mismo propósito.

La condición previa es la orden registrada con anterioridad. La condición resultante es que el estado de la orden ha llegado a la palmtop del mesero. El actor beneficiado es el Cliente.

Los pasos son:

1. El mesero activa la interfaz en la palmtop para sondear una orden registrada.
2. La interfaz le muestra al mesero una lista de las órdenes que tiene registradas en la cocina.
3. El mesero elige la orden que desea sondear.
4. El sistema transmite el mensaje de sondeo a la PC de la cocina.
5. La PC de la cocina recibe el mensaje.
6. El chef selecciona la orden de la cual se quiere conocer su avance.
7. El chef teclea un tiempo estimado para completar la orden.
8. El sistema transmite el tiempo estimado a la palmtop del mesero.

Notificar al chef del progreso de los clientes en sus alimentos

A partir de este caso de uso, utilizaré subtítulos para indicar los aspectos del análisis del caso de uso y viñetas para establecer frases para cada subtítulo con dos excepciones: Seguiré numerando los pasos y no utilizaré viñetas para la descripción.

Descripción

Mediante la red, el mesero le indica al chef que un cliente ya casi ha finalizado con su entremés.

Conjeturas

- El mesero se encuentra en el área de servicio del cliente
- El mesero puede medir el progreso del cliente
- El sistema cuenta con una interfaz para el estado del cliente
- El sistema transmite mensajes de la palmtop a la PC de la cocina y viceversa.

Condiciones previas

- El cliente ha finalizado de manera parcial el entremés

Condiciones resultantes

- El chef ha iniciado los pasos finales para completar el plato principal

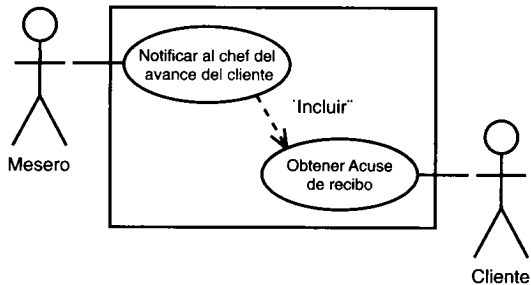
Pasos

1. En la palmtop, el mesero activa la interfaz para el estado del cliente.
2. La interfaz de usuario trae una lista de las mesas en el área de servicio del mesero.
3. El mesero elige la mesa indicada.
4. El mesero envía un mensaje de “ya casi terminó con el entremés” a la PC de la cocina.
5. La PC de la cocina recibe el mensaje.
6. El mesero obtiene un acuse de recibo de la PC de la cocina.

Este último paso utiliza el caso de uso “Acuse de recibo”, que se encuentra en el paquete Mesero. La figura 19.2 le muestra un diagrama de este caso de uso.

FIGURA 19.2

Diagrama de caso de uso para “Notificar al chef del progreso de los clientes en sus alimentos”.



Actor beneficiado

- Cliente

Totalizar una cuenta

Descripción

Sumar los elementos en la orden.

Conjeturas

- Existe una base de datos de pedidos accesibles mediante la palmtop
- Cada elemento de la orden se relaciona con su precio

Condiciones previas

- El servicio ya ha terminado con sus alimentos

Condiciones resultantes

- Se totaliza la cuenta

Pasos

1. El mesero trae una lista de órdenes activas en la palmtop.
2. El mesero elige la orden adecuada.
3. El mesero presiona un botón en la palmtop para totalizar la cuenta.
4. El sistema calcula el total de los precios en la cuenta.

Actor beneficiado

- Cliente

Imprimir una Cuenta

Aunque podría parecer trivial, es una parte importante de la transacción.

Descripción

Imprimir la cuenta totalizada.

Conjeturas

- Una impresora conectada a la red inalámbrica en el área de servicio

Condiciones previas

- Una cuenta totalizada

Condiciones resultantes

- Una cuenta impresa

Pasos

1. El mesero presiona un botón para imprimir la cuenta.
2. La impresora conectada a la red en el área de servicio imprime la cuenta.
3. El mesero presiona un botón en la palmtop para quitar esta orden de la lista de órdenes activas.

Actor beneficiado

- Cliente

Llamar a un Asistente

Descripción

Solicitar a un asistente que limpie la mesa para el siguiente cliente.

Conjeturas

- El sistema permite una comunicación inalámbrica entre dos empleados en movimiento
- El sistema cuenta con una interfaz para enviar un mensaje a un asistente

Condiciones previas

- Una mesa vacía tendrá que ser limpiada y preparada

Condiciones resultantes

- El asistente llega a la mesa para limpiarla y prepararla

Pasos

1. El mesero activa la interfaz para enviar un mensaje a un asistente.
2. El mesero obtiene un acuse de recibo del asistente.

Como en el caso de uso “Notificar al chef del progreso de los clientes en sus alimentos”, el último paso utiliza el caso de uso “Acuse de recibo”.

Actor beneficiado

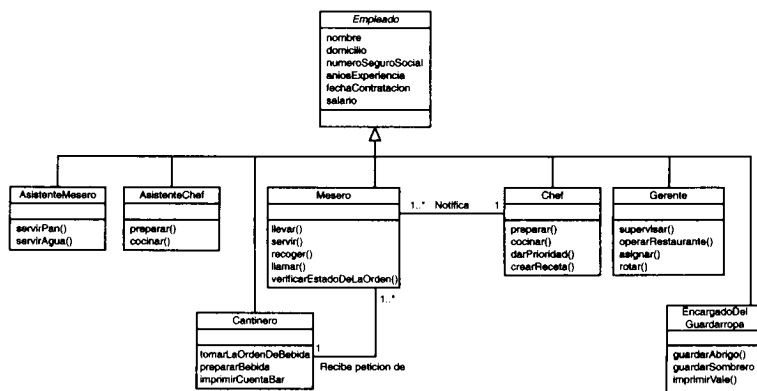
- Asistente

Al analizar este caso de uso, así como los casos de uso en el paquete del Asistente, tal vez pensemos que separar la clase Asistente en dos, AsistenteMesero y AsistenteChef, sería una buena idea (aclararía las cosas). ¿Podrían ambas ser clases secundarias de una clase abstracta Asistente? Podrían, pero posiblemente no se obtendría mucho de establecer esta clase abstracta.

La creación de estas dos clases requiere una revisión al análisis del dominio. Tendremos que volver a trabajar los diagramas de clases, en particular el diagrama de Empleado, como muestra la figura 19.3.

FIGURA 19.3

El diagrama actualizado de la clase Empleado.



También tendríamos que actualizar nuestros diagramas de paquetes para incluir el de Asistente Mesero y Asistente Chef.

Esto es un ejemplo de la forma en que los segmentos de GRAPPLE se intercomunican. Los conocimientos obtenidos durante el análisis de los casos de uso han ayudado a desarrollar el análisis del dominio.

Casos de uso restantes

Los restantes casos de uso en el paquete Mesero son muy similares a los que hemos analizado. Le dejaré como ejercicio finalizar el análisis de este paquete (vea el ejercicio 1 del taller).

Componentes del sistema

Un aspecto importante del análisis del caso de uso es que empezó a descubrir componentes del sistema. Antes de finalizar con esta hora, tome nota de los componentes que hemos visto durante nuestro análisis de los casos de uso en el paquete Mesero. Los verá en la sección Conjeturas de cada análisis de los casos de uso (habrá otros componentes que aparecerán cuando haga los ejercicios).

En el lado del software, es obvio que son necesarias varias interfaces de usuario. RICO necesitará interfaces del usuario basadas en palmtops para registrar órdenes, cambiarlas, sondearlas, dar el estado del cliente y enviar mensajes para un asistente. Como buena medida, algo como una interfaz tipo “página principal” será necesario para tener organizadas las demás interfaces. RICO también necesitará una interfaz del usuario en la PC de la cocina para permitir al chef verificar el avance de una orden.

También parece que necesitaremos una base de datos que contenga a todos los pedidos. Cada registro contendrá la mesa, el pedido, la hora en que se realizó, el mesero, si el pedido está activo, y cosas así.

Del lado del hardware, necesitaremos una red inalámbrica, palmtops para los empleados en movimiento (meseros, asistentes y mozos de piso) y una PC de escritorio en la cocina, así como otra en la recepción. Necesitaremos una impresora conectada a la red en cada área de servicio. Probablemente, también necesitaremos una palmtop y una impresora para el encargado del guardarropa.

Ya está tomando forma un mejor documento de diseño. En la hora siguiente, profundizará aún más en los casos de uso.

Resumen

No basta con listar todos los casos de uso. Un equipo de desarrollo deberá comprender cada uno detalladamente para empezar a comprender el sistema. En esta hora vimos detalladamente el análisis de cada parte de un caso de uso.

Un análisis de casos de uso involucra la especificación de una descripción del propio caso de uso, derivar las condiciones previa y resultante, y especificar los pasos. Un aspecto importante del análisis de los casos de uso es que los componentes del sistema empiezan a ser evidentes.

Preguntas y respuestas

- P** En el segmento inicial de GRAPPLE, noté que pasó por alto la acción “Identificación de los sistemas cooperativos”. ¿A qué se debe?
- R** Como recuerda, este equipo de desarrollo inició un proyecto sin precedentes. No hay sistemas cooperativos. No obstante, el siguiente sistema que alguien diseñe para Restaurantes LNG tal vez tendría que acceder a RICO de alguna forma.
- P** En esta hora modificó los diagramas de casos de uso y el de clases. ¿Esto sucede con frecuencia?
- R** Así es. No deberá ser reactivo a hacer modificaciones conforme evolucione su conocimiento. La lista original de casos de uso capturó todo el conocimiento en cierto punto del proyecto, y captura una imagen de tal momento. Los diagramas modificados representarán las ideas subsecuentes del equipo de desarrollo.

Taller

El taller para esta hora verificará su conocimiento para realizar casos de uso. Para ver las respuestas reales, vea el apéndice A, “Respuestas a los cuestionarios”.

Cuestionario

1. ¿Cuáles son las partes de un diagrama de casos de uso típico?
2. ¿A qué se refiere que un caso de uso “incluya” (o “utilice”) a otro?

Ejercicios

1. Genere el diagrama de casos de uso para “Llamar a un mozo de piso”.
2. Analice los restantes casos de uso del paquete Mesero, y dibuje sus diagramas.
3. Analice los casos de uso del paquete Chef, y dibuje los casos de uso.
4. Haga lo mismo para los paquetes Cantinero, Asistente y Mozo de piso.

Analice los casos de uso que haya obtenido de su proyecto Biblioteca.