2.6 Análisis Funcional

Por medio del análisis funcional:

- Se muestra las operaciones de los objetos y sus dependencia de datos por medio de los diagramas de flujo de datos.
- Se descompone las operaciones hasta que se obtienen funciones sencillas.
- Se provee una descripción de la funcionalidad y limitaciones de cada función sencilla.
- Se identifican las restricciones entre objetos.
- Se especifica un criterio de optimización.

Para lograr el análisis funcional se requiere que se haya completado previamente:

- La descripcion del problema
- El diccionario de datos
- El análsis de objetos
- El análisis dinámico

El proceso de análisis para el modelo funcional se muestra en la Figura 2.237.

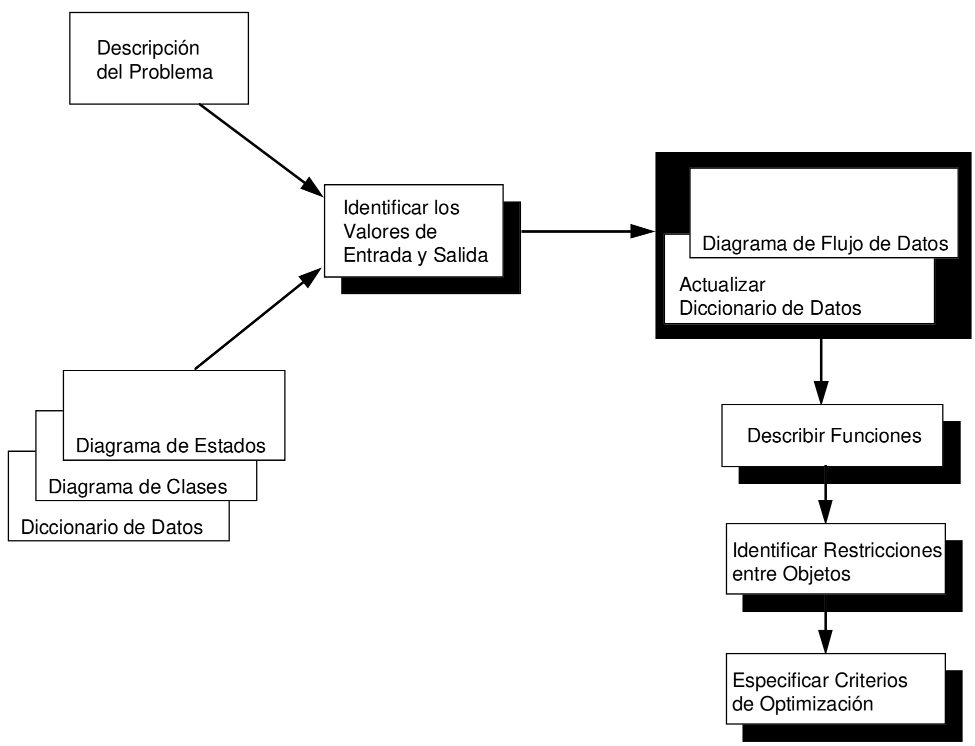

Figura 2.237. Pasos a seguir en el análisis funcional.

Los siguientes son los pasos a seguir:
1) Establecer los bordes del sistema
2) Identificar los datos de entrada y salida
3) Construir el diagrama de contexto
4) Construir los diagramas de flujo de datos

5) Describir los procesos y funciones
6) Identificar las restricciones
7) Actualizar el diccionario de datos
8) Comparar los modelos

2.6.1 Establecer los Bordes del Sistema

Para establecer los bordes del sistema se debe identificar cuales objetos pertencen al sistema y cuales son externos a él.

- Durante esta etapa se identifican los objetos externos al sistema que luego aparecerán en el diagrama de contexto.

Ejemplo Cajero: El borde del sistema para el sistema de cajeros se muestra en la Figura 2.238.

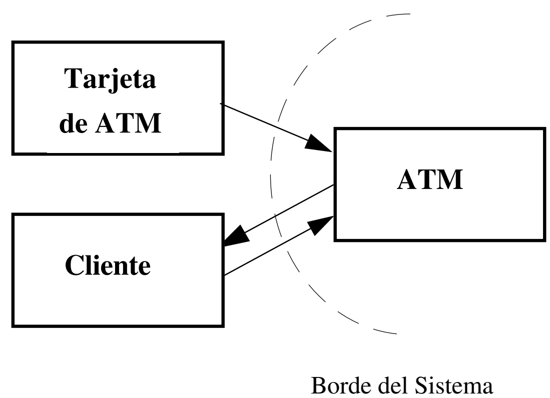

Figura 2.238. Borde del sistema para el sistema de cajeros.

Todas las interacciones entre el sistema y los objetos del mundo externo pasan por la máquina de cajero (ATM) o por la estación de cajero.

2.6.2 Identificar los Datos de Entrada y Salida

Se examina los bordes del sistema y se lista los valores de entrada y salida, los cuales corresponden a parámetros de los posibles eventos entre el sistema y el mundo exterior. Estos valores luego aparecerán en el diagrama de contexto.

- Se busca transacciones, mensajes, e interacciones de interfaces entre el sistema y el mundo externo.

- Se examina la descripción del problema por cualquier valor que no se haya incluido.

Ejemplo Cajero: Los valores de entrada y salida para el sistema, identificados de la descripción del problema y de los diagramas de estado elaborados en el análisis dinámico, son
Tarjeta
código de banco

código de tarjeta
<u>Cliente</u>
contraseña,
tipo de transacción
tipo de cuenta
cantidad
<u>ATM</u>
efectivo
recibos
mensajes

todos los valores de entrada y salida son parámetros de eventos del ATM. Eventos de entrada que solo afectan el flujo de control, como *cancelar*, *terminar*, o *continuar*, no aportan valores de entrada, al igual que eventos de confirmación como *tomar dinero*, o *tomar tarjeta*.

2.6.3 Construir el Diagrama de Contexto

Tomando en cuenta los dos pasos anteriores, la identificación de los bordes del sistema y de los datos de entrada y salida, se puede construir el diagrama de contexto para el sistema, el que corresponde al diagrama de flujo de datos de más alto nivel.

- El diagrama de contexto coresponde al diagrama de flujo de datos de más alto nivel de las clases con interacción externa.

<u>Notación OMT</u>

Ejemplo Cajero: El diagrama de contexto para el sistema de cajeros se muestra en la Figura 2.239.

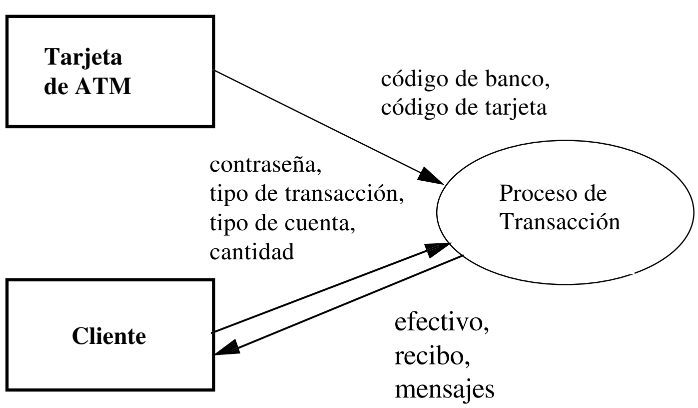

Figura 2.239. Diagrama de contexto para el sistema de cajeros.

2.6.4 Construir los Diagramas de Flujo de Datos

Se debe construir los diagramas de flujo de datos, un diagrama por clase, mostrando cómo los valores de salida se computan de los valores de entrada.

- Se construyen los diagrama de flujos de datos correspondiendo a las capas intermedias

Ejemplo Cajero: El diagrama de flujo de datos de más alto nivel, después del diagrama de contexto, para la máquina de cajero (ATM) se muestra en la Figura 2.240. Los valores de entrada y salida son proporcionados y consumidos por los objetos externos: *cliente* y *tarjeta de ATM*.

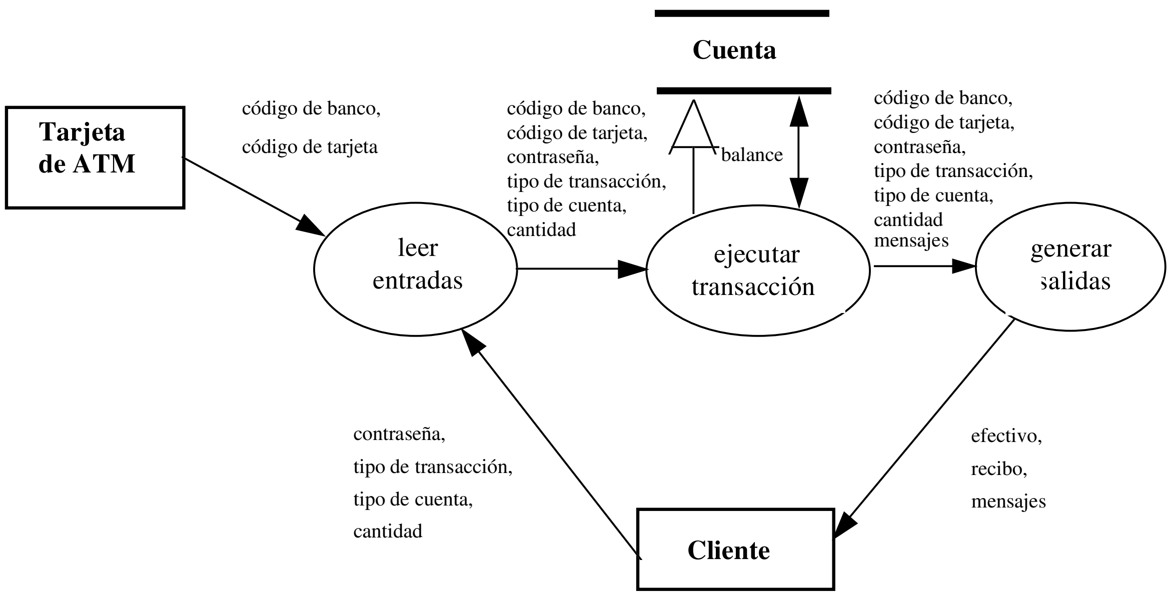

Figura 2.240. Diagrama de flujo de datos de alto nivel para el ATM.

- Dentro de cada capa se debe determinar los valores de salida de cada capa. Se debe hacer el trazo de variables hacia atrás para cada valor de salida para determinar la operación que lo computa.

- Se debe determinar las operaciones necesarias para producir los valores de salida.

- Se debe determinar los valores de entrada necesarios para ejecutar las operaciones.

- Si los valores de entrada a las operaciones corresponden a todas las entradas del diagrama entonces ya se ha terminado. Algunas de las entradas de las operaciones son valores intermedios que deben ser trazados hacia atrás hasta llegar a las entradas del diagrama. También se pueden trazar hacia adelante, de entradas a salidas, pero es más complicado.

- Se debe añadir las operaciones necesarias y continuar el trazo de valores hacia atrás.

- Se debe expandir cada proceso no trivial en el diagrama, de arriba hacia abajo, en diagramas de flujo de datos de más bajo nivel.

- Se debe expandir los diagramas hasta un nivel que produzca funciones simples

Ejemplo Cajero: el diagrama de flujo de datos para el proceso *ejecutar transacción* se muestra en la Figura 2.241. El proceso *ejecutar transacción* selecciona la cuenta apropiada basado en el *código de banco*, *código de tarjeta*, y *tipo de cuenta*. El proceso de *actualización* se aplica al *tipo de transacción* y *cantidad*. La máquina de cajero automático lee y escribe valores al objeto *cuenta*. *Verificar código* es una operación de decisión, se muestra la señal de error que se puede producir pero se deja implícito la flecha de control al proceso *actualización de cuenta*.

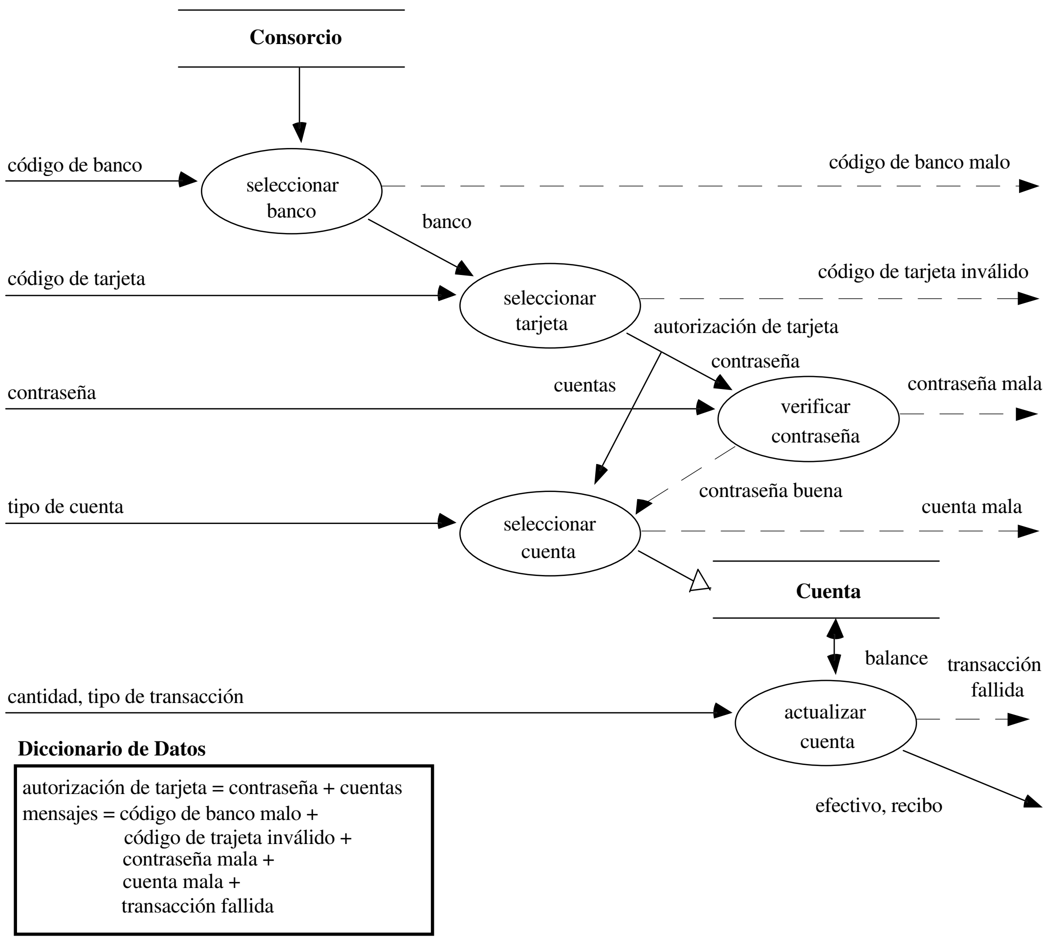

Figura 2.241. Diagrama de flujo de datos para el proceso *ejecutar transacción*.

2.6.5 Describir los Procesos y Funciones

Cuando el diagrama de flujo de datos ya ha sido refinado lo suficiente, se debe escribir una descripción para cada función, usando el formato más apropiado.

Ejemplo Cajero: La descripción de la función para *actualizar cuenta* es la siguiente:

actualizar cuenta(*cuenta*, *cantidad*, *tipo-de-transacción*)->*efectivo*, *recibo*, *mensaje*
- Si la *cantidad* de un retiro excede el balance actual de la *cuenta*, se rechaza la transacción y no se da *efectivo*.
- Si la *cantidad* de un retiro no excede el balance actual de la *cuenta*, se resta de la *cuenta* y se da la *cantidad* de *efectivo*.
- Si el *tipo-de-transacción* es un depósito, se acredita la *cuenta* y no se da *efectivo*.
- Si el *tipo-de-transacción* es un pedido de estado no se da *efectivo*.
- En cualquier caso,
  los *recibos* muestran el número de ATM, fecha, hora, número de cuenta, *tipo-de-transacción*, *cantidad* (si hay) y balance nuevo.

2.6.6 Identificar las Restricciones

Se debe identificar las restricciones en las funciones impuestas por los requisitos del problema, estableciendo tiempos y condiciones correspondientes.

Ejemplo Cajero:
"Ningún balance de cuenta puede ser negativo en ningún momento".
Si se añade cuentas con privilegio de sobregiro, la restricción sería
"Ningún balance de cuenta puede exceder el límite de crédito para la cuenta".
Estas restricciones no especifican que se debe hacer si el cliente trata de hacer un retiro excesivo. Para tal caso, el analista debe incorporar las restricciones en los modelos dinámicos y funcionales.

2.6.7 Actualizar el Diccionario de Datos

El diccionario de datos debe ser actualizado con todos los datos, definidos en los flujos de datos y flujos de control, definiendo entradas, salidas, y datos intermedios, incluyendo almacenamiento de datos.

2.6.8 Comparar Modelos

Se debe comparar el modelo funcional con el modelo de objetos y el modelo dinámico, actualizando el diccionario de datos.

- Se debe tratar de simplificar las operaciones lo máximo posible.
- Se debe buscar operaciones similares.
- Se debe buscar variaciones en una sola operación. Las partes comunes se deben reemplazar con una nueva operación más generalizada.
- Se debe usar herencia para reducir el número de operaciones distintas.
- Se debe ubicar las operaciones lo más alto posible en la jerarquía de herencia.
- Se debe definir operaciones más generales para incluir variaciones y casos especiales.

- Se deben definir operaciones generales que provean una base para la definición de un objeto más allá del problema inmediato.
- Se debe añadir al modelo de objetos cualquier objeto o atributos que se hayan identificado durante la etapa del análisis funcional.
- Se debe verificar si existen transformaciones funcionales del modelo dinámico que hayan sido omitidas.
- Se debe verificar si existen restricciones funcionales que requieran la introducción acciones condicionales en el modelo dinámico.

En la Figura 2.242 se muestra una tabla de relaciones entre los tres modelos.

Modelo de Objetos	Modelo Dinámico	Modelo Funcional
Clases		Actores Almacenamiento de datos
Atributos		Flujos de datos Almacenamiento de datos
Operaciones Métodos	Actividades Acciones Eventos	Procesos Funciones Flujos de Control

Figura 2.242. Tabla de relaciones entre los tres modelos.

2.6.9 Especificar Criterios de Optimización

El analista debe decidir si es necesario especificar algún criterio de optimización al sistema.

- Se debe examinar la descripción del problema nuevamente.
- Se debe especificar que aspectos del sistema deben ser maximizados, minimizados, u optimizados.
- Si hay varios criterios de optimización conflictivos, se debe indicar como decidir entre ellos.

Ejemplo Cajero: Uno de los criterios en el desarrollo del sistema es

"Minimizar el número de mensajes físicos enviados entre diferentes lugares."
"Minimizar el tiempo que una cuenta está bloqueada por razones de concurrencia."
"Minimizar el tiempo que el banco está bloqueado por razones de concurrencia."

2.7 Análisis General

Luego de haber completado el proceso de análisis para cada modelo es necesario revisar el sistema completo por medio de nuevas iteraciones, en particular es necesario definir las operaciones finales del sistema.

2.7.1 Identificar Operaciones Adicionales

Identificar las operaciones verificando los diferentes modelos:
1) operaciones del modelo de objetos
2) operaciones del modelo dinamico
3) operaciones del modelo funcional

2.7.1.1 Operaciones del Modelo de Objetos

Las operaciones en las clases incluyen consultas, lectura de valores de atributos, y accesos a ligas. Estas operaciones no tienen que ser mostradas explícitamente en el modelo de objetos.

- Durante análisis se asume que se puede consultar los valores de todos los atributos. Durante el diseño se define que atributos se pueden consultar de forma *pública* y cuales solo de forma *privada*.

- Las operaciones de acceso a las ligas de una objeto no son necesarias durante el análisis. Durante el diseño se especifica cuales accesos a ligas son *públicos* y cuales *privados*.

2.7.1.2 Operaciones del Modelo Dinámico

Las operaciones del modelo dinámico incluyen, acciones, actividades, y eventos. Durante el análisis del modelo funcional es necesario describir de forma completa las acciones y actividades. Los eventos pueden ser representados como etiquetas en transiciones de estados y no deben ser listados de forma explícita en el modelo de objetos.

2.7.1.3 Operaciones del Modelo Funcional

Cada función en el diagrama de flujo de datos corresponde a una operación en un objeto (o en varios objetos).

Ejemplo Cajero: En el proceso *ejecutar transacción* en el diagrama de alto nivel, las operaciones *seleccionar* son accesos a asociaciones en el modelo de objeto. Las únicas operaciones importantes son *verificar contraseña* y *actualizar cuenta*. Se puede definir *verificar contraseña* como una operación de la clase *autorización tarjeta,* y *actualizar cuenta* puede definirse como una operación en la clase *cuenta*. Puede ser conveniente expandir la definición de *actualizar cuenta* definiendo las siguientes funciones para la clase *cuenta* según el tipo de transacción:

retiro(*tipo de transacción,cantidad*)->*balance*
depósito(*tipo de transacción,cantidad*)->*balance*

Otras funciones para *cuenta*:

cerrar()
autorizar-tarjeta-dinero(*autorización-tarjeta*)

Las siguientes son algunas de las funciones para el *banco*:

crear-cuenta-ahorro(*cliente*)->*cuenta*
crear-cuenta-cheques(*cliente*)->*cuenta*
crear-tarjeta-atm(*cliente*)->*autorizacion-tarjeta*

2.7.2 Iterar el Análisis

El análisis final sirve como base para la arquitectura del sistema, diseño, e implementación. La descripción del problema original debe ser revisada para incorporar correcciones y cambios descubiertos durante el análisis. No hay análisis perfecto ni una separación clara con el diseño.

- Se debe verificar los resultados con los expertos.
- Se debe buscar objetos simples que tengan dos roles distintos definiendo nuevos objetos para cada identidad.
- Se debe restructurar los modelos para capturar mejor las restricciones, y remover objetos o asociaciones que no parecen importantes.