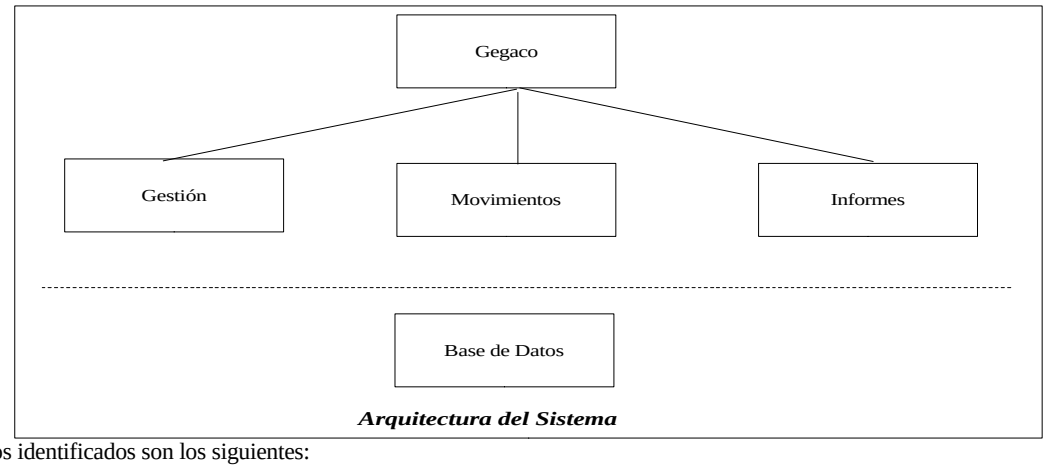
4. DISEÑO DEL SISTEMA

4.1 Metodología de diseño de alto nivel

Se utiliza la metodología de Diseño Estructurado, basada en la descomposición funcional del sistema.

4.2 Descomposición del sistema

La estructura modular del sistema aparece representada en la figura siguiente.



Arquitectura del Sistema

Los módulos identificados son los siguientes:

- Gegaco: Es el programa principal, realiza el diálogo con el operador.
- Gestión: Módulo que realiza las funciones de gestión del programa
- Movimientos: Módulo que realiza las funciones de actualización
- Informes: Módulo que se encarga de realizar los informes gráficos
- Base de Datos: Módulo que contiene la base de datos del sistema.

El módulo base de datos está realizado físicamente por el SGBD de código abierto que se utiliza.

5. DESCRIPCION DE COMPONENTES

5.1Módulo: Base de Datos.

5.1.1*Tipo*: Base de datos relacional

5.1.2*Objetivo*: Este módulo contiene la base de datos relacional que almacena la información persistente del sistema.

5.1.3*Función*: Almacenar los datos que se indican

5.1.4*Subordinados*: Ninguno

5.1.5*Dependencias*: Gestión, Movimientos, Informes.

5.1.6*Interfaces*: El modelo fisico de datos es el siguiente:

CLIENTES				
Campo	Tipo	Long	Indice	Descripción
Idcliente	Char	9	SI	NIF del cliente
Nombre	Char	15	No	Nombre del cliente
Apellido	Char	15	No	Primer apellido del cliente
Seg_apellido	Char	15	No	Segundo apellido del cliente

PROVEEDOR				
Campo	Tipo	Long	Indice	Descripción
Idproveedor	Char	9	SI	CIF del proveedor
Nombre	Char	50	No	Nombre comercial del proveedor

PRODUCTOS				
Campo	Tipo	Long	Indice	Descripción
Idproducto	Numero	4	SI	Código de referencia del producto

Descripción	Char	50	No	Breve descripción del producto
Precio	Número	10	No	Precio en € del producto

DIRECCION				
Campo	Tipo	Long	Indice	Descripción
Ident	Char	9	SI	NIF o CIF de identificación
Calle	Char	30	No	Calle del domicilio
Número	Número	5	No	Número de la calle
Piso	Char	5	No	Piso del edificio
Localidad	Char	30	No	Localidad de la dirección

MUNICIPIO				
Campo	Tipo	Long	Indice	Descripción
Código	Número	5	Si	Código postal de la localidad
Localidad	Char	30	No	Localidad de residencia
Municipio	Char	30	No	Municipio de la localidad
Provincia	Char	30	No	Provincia que pertenece el municipio

PERIODO				
Campo	Tipo	Long	Indice	Descripción
Dias	Número	4	Si	Número de dias que componen el periodo
Tipo	Char	20	No	Identificación de un tipo de periodo

CUENTA				
Campo	Tipo	Long	Indice	Descripción
Saldo	Número	12	Si	Saldo total de la cuenta
Fecha	Tiempo	8	No	Fecha de realización del movimiento
Descripción	Char	50	No	Descripción del movimiento
Cantidad	Número	12	No	Cantidad del abono o ingreso en cuenta
SERVICIO				
Campo	Tipo	Long	Indice	Descripción
Idservicio	Número	6	Si	Número de referencia de un servicio
Idcliente	Char	9	Si	NIF del cliente que contrata el servicio
Idproveedor	Char	9	Si	CIF del proveedor del servicio
Descripción	Char	50	No	Descripción del servicio

DESCRIPCION				
Campo	Tipo	Long	Indice	Descripción
Idservicio	Número	6	Si	Número de referencia del servicio
Idproducto	Número	4	Si	Número de referencia del producto
Cantidad	Número	6	No	Unidades del producto en el servicio

PREVISION				
Campo	Tipo	Long	Indice	Descripción
Idproducto	Número	4	Si	Número de referencia del producto
Idproveedor	Número	9	Si	CIF del proveedor
Fecha	Tiempo	8	Si	Fecha de consulta de precios
Precio	Número	12	No	Precio del producto en la fecha

APROXIMACION				
Campo	Tipo	Long	Indice	Descripción
Idproducto	Número	4	Si	Número de referencia del producto
Idproveedor	Número	9	Si	CIF del proveedor
Precio_Estimado	Número	12	No	Precio estimado del producto

MOVIMIENTO				
Campo	Tipo	Long	Indice	Descripción
IdMovimiento	Número	6	Si	Número de referencia del movimiento
Descripción	Char	50	No	Descripción
Fecha_base	Tiempo	8	No	Inicio del movimiento

MOVIMIENTO-PREVISTO				
Campo	Tipo	Long	Indice	Descripción
IdMovimiento	Número	6	Si	Número de referencia de movimiento
Periodo	Char	20	No	Identificación de periodicidad
Actor	Numero	9	No	Identificación del cliente o proveedor
Importe	Numero	12	No	Importe del movimiento

5.1.7 *Recursos*: Ninguno.

5.1.8 Referencias: Ninguna

5.1.9 Proceso: Ninguno

5.1.10 *Datos:* Ver interfaces

5.2Módulo : Gedaco

5.2.1 *Tipo*: Abstracción funcional (programa principal).

5.2.2 Objetivo: Es el programa principal de la aplicación

5.2.3 Función: Este módulo se encarga de realizar el diálogo con el usuario en lo referente a la seleción de la función deseada en cada momento. También realizará las funciones oportunas al comienzo y final de la sesión de trabajo.

5.2.4 Subordinados: Gestión, Movimientos, Informes.

5.2.5 Dependencias: Ninguna

5.2.6 *Interfaces*: No aplicable

5.2.7 Recursos: Ninguno

5.2.8 Referencias: Ninguna

5.2.9 *Proceso*:

- Iniciar sesión
- REPETIR:

Presentar menú principal
Elegir opción
CASO opción de
Gestión
Presentar menú gestión
Elegir opción
CASO opción de
Alta producto: Realizar alta producto
Alta clientes: Realizar alta de clientes
Alta proveedores: Realizar alta proveedor
Alta previsiones: Realizar alta previsión
Alta servicio: Realizar alta servicio
Baja producto: Realizar baja de producto
Baja cliente: Realizar baja cliente
Baja proveedores: Realizar baja proveedor
Baja previsiones: Realizar baja previsión
Baja servicio: Realizar baja servicio
Modificación producto: Actualizar producto
Modificación cliente: Actualizar datos cliente
Modificación de proveedores: Actualizar datos proveedor
Modificación previsiones: Actualizar previsión
Modificación servicios: Actualizar servicio
Listado productos: Presentar listado de productos
Listado productos-proveedor: Listado
Listado clientes: Presentar listado de clientes
Listado servicios: Presentar listado de servicios
Listado de compras: Calcula el precio de una compra
FIN-CASO
Movimientos
Presentar menú movimientos
Elegir opción
CASO opción de
Ingresos: Realizar apunte ingreso
Abonos: Realizar apunte de abono
Ultimos movimientos: Presentar listado movimientos
Ultimos dias: Listado de movimientos de X días
Previsión: Realizar previsión gastos
Compras: Predicción de la compra
FIN-CASO
Informes
Presentar menú informes
Elegir opción
CASO opción de
Tabular: Realizar gráfico tabular
Sectores: Realizar gráfico de sectores
Barras: Realizar gráfico de barras
Impresión: Imprimir el gráfico
FIN-CASO
HASTA Fin de sesión
Terminar sesión

5.2.10 Datos: Ver base de datos

5.3Módulo: Gestión

5.3.1 *Tipo*: Abstracción funcional (colección de funciones)

5.3.2Objetivo: Realizar las distintas operaciones de mantenimiento del sistema Gestransa correspondientes al registro de clientes, productos, proveedores, servicios y generar listados de los mismos.

5.3.3 Función:

Alta de productos: Da de alta un nuevo producto.

Entrada:

Salida:

Usa: Proveedores

Actualiza: Productos, Previsiones

Efecto: Compone una nueva ficha de producto, se le asignará un proveedor de un listado.

Excepciones: Si el código del producto existe se mostrará un mensaje de error y se saldrá.

Proceso:

Editar la ficha del producto mediante un formulario en pantalla.

El proveedor del producto se selecciona de la lista de proveedores, si éste no existe se deberá salir de la función y darlo de alta como proveedor.

Pedir conformidad con los datos

SI se confirma ENTONCES

Registrar el nuevo producto en las tablas Productos y Previsión

SI NO

Anular la asignación de nuevo número de referencia

FIN SI

Alta de clientes: Da de alta un nuevo cliente.

Entrada:

Salida:

Usa: Clientes, Municipio

Actualiza: Clientes, Dirección

Efecto: Compone una nueva ficha de cliente, la localidad de residencia debe de existir en el listado de municipios y códigos postales.

Excepciones: Si existe un cliente con el mismo NIF que se introduce se mostrará un mensaje de error y se saldrá de la función.

Proceso:

Editar la ficha del cliente mediante un formulario en pantalla.

La localidad de residencia se elige de una lista desplegable.

Pedir confirmación

SI se confirma ENTONCES

Registrar el nuevo cliente en las tablas Clientes y Dirección

SI NO

Anular la asignación de datos del cliente dejando el formulario vacío

FIN SI

Alta de proveedores: Da de alta un nuevo proveedor.

Entrada:

Salida:

Usa: Proveedor, Municipio

Actualiza: Proveedor, Dirección

Efecto: Compone una nueva ficha de proveedor, la localidad de residencia debe de existir en el listado de municipios y códigos postales.

Excepciones: Si existe un proveedor con el mismo CIF que se introduce se mostrará un mensaje de error y se saldrá de la función.

Proceso:

Editar la ficha del proveedor mediante un formulario en pantalla.

La localidad de residencia se elige de una lista desplegable.

Pedir confirmación

SI se confirma ENTONCES

Registrar el nuevo proveedor en las tablas Proveedor y Dirección

SI NO
Anular la asignación de datos del proveedor dejando el formulario vacío
FIN SI

Alta de previsiones: Da de alta una nueva previsión .
Entrada:
Salida:
Usa: Clientes, Proveedores, perdiodo, movimiento
Actualiza: Movimiento-previsto
Efecto: Crea una nueva ficha de previsiones de abonos o ingresos
Excepciones:
Proceso:
Editar la ficha de previsión mediante un formulario en pantalla.
Los clientes o proveedores se elegiran de una lista desplegable.
Los productos o servicios causantes del movimiento previsto deben existir
Pedir confirmación
SI se confirma ENTONCES
Registrar la nueva previsión en la tabla de movimientos-previstos
SI NO
Anular asignación de previsión
FIN SI

Alta de servicios: Da de alta un nuevo servicio.
Entrada:
Salida:
Usa: Cliente, Proveedor, Producto
Actualiza: Descripción
Efecto: Crea un nuevo servicio contratado por un cliente que consta de una serie de productos.
Un cliente sólo puede contratar un producto en un determinado servicio
El cliente deberá existir como tal.
El proveedor debe estar dado de alta y como distribuidor del producto
Excepciones: Si el cliente ya tiene contratado un servicio con estos productos se indicará mediante un mensaje de error.
Proceso:
Editar la ficha de servicio mediante un formulario en pantalla.
Los clientes, proveedores y productos se eligiran de una lista desplegable.
Pedir confirmación
SI se confirma ENTONCES
Registrar el nuevo servicio en las tablas de servicio y la descripción del mismo en la tabla descripción
SI NO
Anular asignación de servicio
FIN SI

Baja de productos: Da de baja un producto.
Entrada:
Salida:
Usa: Servicio, Proveedores, Previsión
Actualiza: Productos, Proveedores
Efecto: Da de baja un producto servido por un determinado proveedor.
Excepciones: Si el producto está asignado a un determinado servicio se enviará un mensaje indicando si la modificación supone la supresión de dicho servicio, caso de que sea el único producto de ese tipo, o bien si se debe de modificar las condiciones del servicio, en ambos casos se pedirá confirmación.
Proceso:
Mostrar listado de productos

Buscar las relaciones del producto seleccionado
SI el producto está asignado a un servicio ENTONCES
CASO último producto
Abrir dialogo de dar de baja servicios
Pedir conformidad
CASO modificar condiciones servicio
Abrir dálogo modificar servicio
Pedir conformidad
FIN CASO
SI NO
Pedir conformidad
SI se confima ENTONCES
Dar de baja el producto y previsiones sobre él
SI NO
Anular selección y volver a pantalla de baja de producto
FIN SI
FIN SI

Baja de clientes: Da de baja un cliente.

Entrada:

Salida:

Usa: Movimiento, Vive_en, Servicio

Actualiza: Cliente, Direccion, Servicio

Efecto: Da de baja un cliente seleccionado de una lista desplegable, dará también de baja los servicios que tuviera contratados así como los datos de dirección y movimientos que tuviera asignados.

Excepciones: Si el cliente no tiene puesto al día los pagos se indicará esta circunstancia y no se realizará la acción.

Proceso:

Mostrar listado de clientes
Buscar las relaciones del cliente seleccionado
SI el cliente tiene contratado un servicio ENTONCES
CASO pagos al día
Pedir conformidad
CASO pagos pendientes
Mostrar mensaje
Salir de la función
FIN CASO
SI NO
Pedir conformidad
FIN SI
SI se confima ENTONCES
Dar de baja el cliente, su dirección y servicios contratados
SI NO
Anular selección y volver a pantalla de baja de cliente
FIN SI

Baja de proveedores: Da de baja un proveedor.

Entrada:

Salida:

Usa: Vive_en, movimiento, servicio, aproximación

Actualiza: Proveedor, Dirección

Efecto: Da de baja un proveedor seleccionado de una lista desplegable, se dan también de baja los servicios que distribuye, así como los datos de dirección y movimientos que tuviera asignados.

Excepciones: Si el proveedor no tiene puesto al día los pagos se indicará esta circunstancia y no se realizará la acción. Si el proveedor prest un servicio, se indicará con un mensaje y se permitirá deshacer la acción.

Proceso:

Mostrar listado de proveedores
Buscar las relaciones del proveedor seleccionado
SI el proveedor distribuye un producto de un sevicio ENTONCES
CASO el producto es reemplazable
abrir diálogo de modificar servicio
CASO el producto sólo es servido por éste distribuidor
Mostrar mensaje
Pedir conformidad de eliminación
SI no se confima
Salir de la función
FIN SI
FIN CASO
FIN SI
SI el proveedor tiene los pagos puestos al día
Pedir conformidad
FIN SI
SI se confima ENTONCES
Dar de baja el proveedor, su dirección y servicios contratados
SI NO
Anular selección y volver a pantalla de baja de proveedor
FIN SI

Baja de prevision: Da de baja una previsión.
Entrada:
Salida:
Usa: Clientes, proveedores, periodo, movimiento
Actualiza: Movimiento-previstos
Efecto: Da de baja una determinada previsión.
Excepciones:
Proceso:
Mostrar listado de previsiones
Pedir conformidad
SI se confima ENTONCES
Dar de baja la previsión seleccionada
SI NO
volver menú de baja de previsión
FIN SI

Baja de servicios: Da de baja un servicio .
Entrada:
Salida:
Usa: Cliente, producto, proveedor
Actualiza: Descripción.
Efecto: Da de baja (borra) un determinado servicio
Excepciones:
Proceso:
Mostrar listado de servicios
Pedir conformidad
SI se confirma ENTONCES
dar de baja el servicio seleccionado
dar de baja la relación con los productos que lo componian
SI NO
volver al menú de baja de servicios
FIN SI

Modificación de productos: Modifica los datos de un determinado producto.

Entrada:

Salida:

Usa: Proveedores

Actualiza: Productos, Previsiones, Servicio

Efecto: Modifica la ficha del producto seleccionado.

Excepciones:

Proceso:

Seleccionar un determinado producto

Editar la ficha del producto mediante un formulario en pantalla.

Modificación por parte del usuario de los datos del producto (a excepción de identificación)

Pedir conformidad con los datos

SI se confirma ENTONCES

Registrar el producto en las tablas Productos

Registar los cambios en servicio

Sumar el valor del producto en la tabla de previsión, y calcular el valor a guardar

SI NO

Volver al menu de modificar producto

FIN SI

Modificación de clientes: Modifica los datos del cliente.

Entrada:

Salida:

Usa: Clientes, Municipio

Actualiza: Clientes, Dirección

Efecto: Modifica la ficha de cliente, la localidad de residencia debe de existir en el listado de municipios y códigos postales.

Excepciones: Si existe un cliente con el mismo NIF que se introduce se mostrará un mensaje de error y se saldrá de la función.

Proceso:

Seleccionar un determinado cliente

Editar la ficha del cliente mediante un formulario en pantalla.

La localidad de residencia se elige de una lista desplegable.

Pedir confirmación

SI se confirma ENTONCES

Registrar los cambios en las tablas afectadas

SI NO

Anular la asignación de datos del cliente dejando los datos anteriores

FIN SI

Modificación de proveedores: Modifica los datos de un proveedor.

Entrada:

Salida:

Usa: Proveedor, Municipio

Actualiza: Proveedor, Dirección

Efecto: Modifica la ficha de proveedor, la localidad de residencia debe de existir en el listado de municipios y códigos postales.

Excepciones: Si existe un proveedor con el mismo CIF que se introduce se mostrará un mensaje de error y se saldrá de la función.

Proceso:

Seleccionar un determinado proveedor

Editar la ficha del proveedor mediante un formulario en pantalla.

La localidad de residencia se elige de una lista desplegable.

Pedir confirmación

SI se confirma ENTONCES

Registrar los cambios en las tablas afectadas

SI NO

Anular la asignación de datos del proveedor dejando los datos anteriores

FIN SI

Modificación de previsiones: Modifica los datos de una previsión .

Entrada:

Salida:

Usa: Clientes, Proveedores, periodo, movimiento

Actualiza: Movimiento-Previsto

Efecto: Modifica una ficha de previsiones de abonos o ingresos

Excepciones:

Proceso:

Seleccionar una determinada previsión

Editar la ficha de previsión mediante un formulario en pantalla.

Los clientes o proveedores se eligiran de una lista desplegable.

Pedir confirmación

SI se confirma ENTONCES

Registrar los cambios en las tablas afectadas

SI NO

Anular la asignación de datos de la previsión dejando los datos anteriores

FIN SI

Modificación de servicios: Modifica los datos de un servicio.

Entrada:

Salida:

Usa: Cliente, Proveedor, Producto

Actualiza: Descripción

Efecto: Modifica un servicio contratado por un cliente que consta de una serie de productos.

Un cliente sólo puede contratar un producto en un determinado servicio

El cliente deberá existir como tal.

El proveedor debe estar dado de alta y como distribuidor del producto

Excepciones: Si el cliente ya tiene contratado un servicio con estos productos se indicará mediante un mensaje de error.

Proceso:

Seleccionar un determinado servicio

Editar la ficha de servicio mediante un formulario en pantalla.

Los clientes, proveedores y productos se eligiran de una lista desplegable.

Pedir confirmación

SI se confirma ENTONCES

Registar los cambios en las tablas afectadas

SI NO

Anular la asignación de datos del cliente dejando los datos anteriores

FIN SI

Listado de productos: Muestra un listado de productos.

Entrada:

Salida:

Usa: Producto

Actualiza:

Efecto: Muestra un listado por pantalla de todos lo productos que están registrados en la base de datos

Excepciones:

Proceso:

Mostrar listado

Listado de productos-proveedor: Muestra un listado de productos.

Entrada:

Salida:

Usa: Producto, proveedor

Actualiza:

Efecto: Muestra un listado por pantalla de todos lo productos que están registrados en la base de datos ordenado por proveedores o productos según voluntad del usuario.

Excepciones:

Proceso:

Seleccionar orden

CASO por productos

Mostrar listado ordenado por productos

CASO por proveedores

Mostrar listado ordenado por proveedores

Listado de clientes: Muestra un listado de clientes.

Entrada:

Salida:

Usa: Clientes, vive_en

Actualiza:

Efecto: Muestra un listado por pantalla de todos lo clientes que están registrados en la base de datos

Excepciones:

Proceso:

Mostrar listado

Listado de servicios: Muestra un listado de servicios.

Entrada:

Salida:

Usa: Cliente, Proveedor, Descripción, Producto, Servicio

Actualiza:

Efecto: Muestra un listado por pantalla de todos lo productos que están registrados en la base de datos

Excepciones:

Proceso:

Mostrar listado

5.3.4 Subordinados: Base de Datos

5.3.5 Dependencias: Gegaco

5.3.6 Interfaces: Ver función

5.3.7 Recursos: Ninguno

5.3.8 Referencias: Ninguna

5.3.9 *Proceso*: Ver función

5.3.10 Datos: Ver módulo Base de Datos

5.4Módulo: Movimientos

5.4.1 *Tipo*: Abstracción funcional (colección de funciones)

5.4.2 Objetivo: Realiza las distintas operaciones sobre el estado contable de Gestransa.

5.4.3 *Función*:

Ingresos: Realiza las actualizaciones correspondientes a un ingreso.

Entrada:

Salida:

Usa: Cliente, Proveedor, Movimiento

Actualiza: Cuenta

Efecto: Realiza el apunte de un ingreso en cuenta

Excepciones:

Proceso:

Leer el importe del ingreso y su descripción.

Leer fecha del sistema

Operar con el saldo e importe del ingreso

Perdir conformidad

SI se confirma ENTONCES

Actualizar movimiento en tabla cuenta

SI NO

Anular la asignación de datos volver al menú de movimientos.

FIN SI

Abonos: Realiza las actualizaciones correspondientes a un abono.

Entrada:
Salida:
Usa: Cliente, Proveedor, Movimiento
Actualiza: Cuenta
Efecto: Realiza el apunte de un abono en cuenta
Excepciones:
Proceso:

Leer el importe del abono y su descripción.
Leer fecha del sistema
Operar con el saldo e importe del abono
Perdir conformidad
SI se confirma ENTONCES

Actualizar movimiento en tabla cuenta

SI NO

Anular la asignación de datos volver al menú de movimientos.

FIN SI

Ultimos movimientos: Listado de los movimientos indicados por el operador.

Entrada:
Salida:
Usa: Cuenta
Actualiza:
Efecto: Muestra por pantalla un listado de los últimos movimientos de la cuenta
Excepciones:
Proceso:

Leer en número de movimientos a mostrar
Mostrar listado de cuenta

Ultimos días: Listado de los movimientos realizados en los dias indicados por el operador.

Entrada:
Salida:
Usa: Cuenta
Actualiza:
Efecto: Muestra por pantalla un listado de los movimientos de la cuenta en los *n* dias indicados
Excepciones:
Proceso:

Leer en número de dias a mostrar
Mostrar listado de cuenta

Previsión: Listado de la previsión de movimientos de los 30 próximos dias.

Entrada:
Salida:
Usa: Movimiento-previsto, cuenta
Actualiza:
Efecto: Muestra por pantalla un listado de la previsión de movimientos en la cuenta
Excepciones:
Proceso:

Leer fecha actual
Calcular movimientos de los 30 próximos días según periodicidad.

Mostrar listado de cuenta

Compras: Muestra una previsión sobre la compra de una serie de artículos

Entrada:

Salida:

Usa: Proveedor, Previsión, Producto

Actualiza

Efecto: A partir de los datos acumulados sobre el precio de los productos, calcula el importe apróximado de una compra indicada por el operador.

Excepciones:

Proceso:

Mostar lista de productos

Recoger selección de productos

Estimar precios

Mostrar listado por pantalla

5.4.4 Subordinados: Base de Datos

5.4.5 Dependencias:Gegaco

5.4.6 Interfaces: Ver función

5.4.7 Recursos: Ninguno

5.4.8 Referencias: Ninguna

5.4.9 Proceso: Ver función

5.4.10 Datos: Ver módulo Base de Datos

5.5Módulo: Informes

5.5.1 *Tipo*:Abstracción funcional (colección de funciones)

5.5.2 Objetivo: Realiza las distintas operaciones sobre el estado contable de Gestransa.

5.5.3 *Función*:

Informe tabular: Genera un informe de los proveedores.

Entrada:

Salida:

Actualiza:

Usa: Proveedores, Servicio

Efecto: Muestra en pantalla un gráfico tabular de los proveedores

Excepciones:

Proceso

Realizar gráfico

Sectores: Genera un informe de los ingresos.

Entrada:

Salida:

Actualiza:

Usa: Clientes, Servicio

Efecto: Muestra en pantalla un gráfico de sectores de los ingresos agrupados por clientes

Excepciones:

Proceso

Cálculos de los ingresos

Realizar gráfico

Barras: Genera un informe de los gastos reales con respecto a los presupuestados.

Entrada:
Salida:
Actualiza:
Usa: Cuenta, Movimiento-previsto
Efecto: Muestra en pantalla un gráfico de barras indicando las diferencias encontradas entre lo presupuestado y lo real
Excepciones:
Proceso
Calcular valores
Realizar gráfico

Impresión: Impresión de un gráfico
Entrada: Gráfico a imprimir
Salida:
Actualiza:
Usa:
Efecto: Realiza la impresión en la impresora determinada del último gráfico generado.
Excepciones: Si no hay ningún gráfico generado no será accesible desde el menú
Proceso
Imprimir gráfico

5.5.4*Subordinados*: Base de Datos

5.5.5*Dependencias*:Gegaco

5.5.6*Interfaces*: Ver función

5.5.7*Recursos:* Ninguno

5.5.8*Referencias*: Ninguna

5.5.9*Proceso*: Ver función

5.5.10*Datos*: Ver módulo Base de Datos

6. VIABILIDAD Y RECURSOS ESTIMADOS

Esta aplicación puede ejecutarse en una máquina tipo PC de gama media o superior. Los recursos mínimos estimados son los siguientes:

- Procesador: 80686
- Sistema Operativo: GNU-LINUX
- Memoria: 64 kb
- Pantalla gráfica.
- Disco duro:
  - 1 Gb para el SGBD
  - 4 Mb para los programas de aplicación
  - 4 Mb para los datos de aplicación
- Entorno gráfico (GNOME)
- Se precisa que se instale GTK y Glib para el entorno gráfico
- Se precisa la instalación previa de un motor de base de datos (SGDB) tipo MySql.

7. MATRIZ REQUISITOS COMPONENTES

	Base de datos	Gegaco	Gestión	Movimiento	Informe
R 1.1	X		-	-	-
R 1.2	-	X	-	-	-
Función 1.1	-	-	X	-	-
Función 1.2	-	-	X	-	-
Función 1.3	-	-	X	-	-
Función 1.4	-	-	X	-	-
Función 1.5	-	-	X	-	-
Función 1.6	-	-	X	-	-
Función 1.7	-	-	X	-	-
Función 1.8	-	-	X	-	-
Función 1.9	-	-	X	-	-
Función 1.10	-	-	X	-	-
Función 1.11	-	-	X	-	-
Función 1.12	-	-	X	-	-
Función 1.13	-	-	X	-	-
Función 1.14	-	-	X	-	-
Función 1.15	-	-	X	-	-
Función 1.16	-	-	X	-	-
Función 1.17	-	-	X	-	-
Función 1.18	-	-	X	-	-
Función 1.19	-	-	X	-	-
Función 1.20	-	-	X	-	-
Función 2.1	-	-	-	X	-
Función 2.2	-	-	-	X	-
Función 2.3	-	-	-	X	-
Función 2.4	-	-	-	X	-
Función 2.5	-	-	-	X	-
Función 2.6	-	-	-	X	-
Función 3.1	-	-	-	-	X
Función 3.2	-	-	-	-	X
Función 3.3	-	-	-	-	X
Función 3.4	-	-	-	-	X
R 2.1	X	-	-	-	-
* R 2.2	-	*	*	*	-
R 2.3	-	-	-	-	X
R 2.4	-	-	-	-	*
R 4.1	-	R	-	-	-
**R 4.2	-	-	-	-	-
R 5.1	-	-	-	-	-
R 6.1	X	-	-	-	-
R 6.2	-	-	-	-	-
R 7.1	-	-	-	-	-
R 7.2	-	-	-	-	-
**R 8.1	-	-	-	-	-
**R 8.2	-	-	-	-	-

NOTA 1: Los requisitos marcados con (*) se deberán comprobar

NOTA 2: Los requisitos marcados con (**) no se cumplen

8. UML

8.1 Introducción

UML es una notación que se produjo como resultado de la unificación de la técnica de modelados de objetos; ha sido diseñado para un amplio rango de aplicaciones, proporcionando construcciones para un amplio rango de sistemas y actividades. El desarrollo de sistemas se enfoca en tres modelos diferentes del sistema:

- Modelo funcional: Funcionalidad del sistema desde el punto de vista del usuario, es representado con diagramas de caso.
- Modelo de objetos: Estructura del sistema desde el puntode vista de objetos, atributos..., se representa con diagramas de clase.
- Modelo dinámico: Comportamiento interno del sistema, se representa con diagramas de secuencia, diagrama de gráfica de estado y diagramas de actividad.

8.2 Notación

8.2.1Diagramas de caso de uso

Los casos de uso se utilizan durante la obtención de requerimientos y el análisis para representar la funcionalidad del sistema; se enfocan en el comportamiento del sistema desde el punto de vista externo, describiendo una función proporcionada por el sistema que produce un resultado visible para un actor. Un actor describe cualquier entidad que interactúa con el sistema (un usuario, otro sistema, un ambiente físico del sistema,...): La identificación de los actores y los casos de uso da como resultado la definición de la frontera del sistema. Cuando los actores y los casos de uso intercambian información, se dice que se comunican.

La descripción de un caso de uso se hace mediante una plantilla de seis campos:

- **Nombre**Identificador único.
- **Actores participantes**ctores que interactúan con el caso de uso
- **Condiciones iniciales**Aquellas que se deben satisfacer antes de que se inicie el caso de uso.
- **Flujo de eventos**Secuencia de acciones del caso de uso. Numeradas
- **Condiciones de salida**escripción de las condiciones que se satisfacen cuando termina el caso de uso.
- **Requerimientos especiales**Aquellos que no están relacionados con la funcionalidad del sistema. Restricciones sobre el desempeño del sistema, plataformas, herramientas, etc.

Los casos de uso se describen en lenguaje natural. Cuando nos referimos a un caso de uso en particular, es decir, una descripción del conjunto de acciones concretas, hablamos de un escenario, esto es una instacia de un caso de uso; para referirnos a un escenario se usa una plantilla de tres campos:

- Nombre: Identificador único, se subraya para indicar que es una instacia
- Instancias de actores participantes: Instancias de los actores involucrados, subrayados.
- Flujo de eventos: Secuencia de eventos paso a paso.

8.2.2.1 Relaciones de los casos de uso

- Comunicación: Intercambio de información entre actores y casos de uso. Línea contínua.
- Inclusión: Una forma de simplificación del modelado, tras identificar las cosas comunes de diferentes casos de uso, estos las incluyen (evitando inconsistencias y redundancias). Dos casos de uso están relacionados por una relación de inclusión si alguno de ellos incluye al segundo en su flujo de eventos. Se representa con una flecha discontínua, que se inicia en el caso que incluye al otro, se etiqueta con <<incluye>>, son los casos de uso con comportamiento compartido.
- Extensión: Un caso de uso puede extender a otro caso de uso mediante la adición de eventos. Una relación extendida indica que una instancia del caso de uso extendido puede incluir el comportamiento especificado por el caso de uso que se extiende, de forma excepcional. Suelen tratarse como extensiones las ayudas, los errores, etc. Se representan por una flecha discontinua, desde la extensión hacia el caso de uso con etiqueta <<extiende>>.
- Generalización/Especialización: Un caso de uso puede especializar a otro más general, añadiendo más detalles.

8.2.2 Diagramas de clase.

Los diagramas de clase se usan para describir la estructura de un sistema, formalizan el conocimento del dominio de la aplicación. Las clases son abstracciones que especifican los atributos y comportamientos de un conjunto de objetos. Los objetos son entidades que encapsulan estado y comportamiento. Cada objeto tiene una identidad: se puede hacer referencia a él de manera individual y es distinguible con respecto a otros objetos.

En UML las clases y objetos se represetan en cuadrados, con el nombre, atributos y operaciones (estas dos se pueden omitir, por claridad). Los nombres de objetos estarán subrayados para indicar que son instancias. Los nombres de clase comienzan con mayúscula.

La conexión entre objetos se llaman vínculos, las conexiones entre clases asociaciones (que serán grupos de vínculos), con el fin de aclarar el propósito de la asociación, se pueden etiquetar con papeles.

Cada extremo de una asociación puede ser etiquetado con números para indicar la cantidad de vínculos que se pueden originar a partir de una instancia de la clase.

Cuando una asociación tiene atributos y operaciones se denomina clase asociación y se conecta a la asociación mediante una línea de guiones.

Cuando una asociación tiene atributos y operaciones se denomina clase asociación y se conecta a la asociación mediante una línea de guiones. Cuando se quiere representar la composición (estado,región, pueblo) en UML se utiliza la agregación, una línea con un rombo en el estremo del contenedor de la asociación. La agregación enfatiza los aspectos jerarquicos de la relación.

La generalización es la aplicación típica de objetos, con la superclase, subclases y relaciones de herencia. La clase abstracta se nombra en cursiva.

8.2.3 Diagramas de secuencia

Los diagramas de secuencia se usan para formalizar el comportamiento del sistema y para visualizar la comunicación entre objetos. Son útiles para la identificación de objetos adicionales que participen en los casos de uso.

Un objeto interactúa con otro objeto enviando mensajes. La recepción de un mensaje por parte de un objeto activa la ejecución de una operación, la cual, a su vez puede enviar mensajes a otros objetos

En los diagramas de secuencia cada columna representa un objeto que participa en la interacción. El eje vertical representa el tiempo de arriba a abajo. Los mensajes se indican con flechas, cuyos rótulos son el nombre del mensaje y pueden contener argumentos.

Mediante los diagramas de secuencia se pueden indicar situaciones con un nivel de detalle variable, cuando la secuencia es abstracta (es decir, describe todas las interacciones posibles) los diagramas de secuencia también proporcionan notaciones para condiciones e interadores. Una condición en un mensaje se indica por una expresión entre corchetes de forma que si la condición es cierta se envía en mensaje. La repetición se indica mediante un asterisco '*'.

8.2.4 Diagramas de gráfica de estado

Los diagramas de gráfica de estado describen el comportamiento de un objeto individual como varios estados y transiciones entre esos estados, es decir, es una notación para la descripción de la secuencia de estados por los que pasa un objeto en respuesta a eventos externos. Un estado es una condición que satisface un objeto, una abstracción de los valores de atributo de una clase. Una transición representa cambios de estado activados por eventos, condiciones o tiempo. Para reducir la complejidad se pueden utilizar transiciones internas o gráficas de estado aisladas. Las transiciones internas son transiciones que permanecen dentro de un sólo estado.

8.2.5 Diagramas de actividad

Un diagrama de actividad describe un sistema desde el punto de vista de las actividades. Una actividad se puede definir como estados que representan la ejecución de un conjunto de operaciones, cuya terminación dispara una transición hacia otra actividad (se pueden asimilar al DFD y al DFC).

Los diagramas de actividad pueden representar decisiones (ramificaciones del flujo de control) como un rombo con una o más flechas entrantes y dos o más flechas salientes, rotuladas con las condiciones que permiten seleccionar la rama.

La sincronización de varias actividades o la división del flujo de control en varios hilos se indican con transiciones complejas, que son acciones que pueden suceder en paralelo. Estas acciones se pueden separar en carriles si interesa indicar el objeto o subsistema que realiza las acciones.

8.2.6 Organización de los diagramas

Los modelos crecen en complejidad conforme se van refinando, esta complejidad puede manejarse agrupando en paquetes los elementos relacionados. Un paquete es un agrupamiento de elementos del modelo (similar a una organización de ficheros y directorios). Los paquetes no son necesariamente jerarquicos, es decir, la misma clase puede aparecer en más de un paquete. Se debe de tener en cuenta que un paquete es una forma de organización por ello no tiene los comportamientos asociados a un objeto (mensajes ...) Se puede asociar una nota a un diagrama con el fin de agregar información a los modelos y a los elementos de los modelos.

8.2.7 Extensiones al diagrama.

Si algo se ha aprendido en la historia del modelado de sistemas software, es que difícilmente se consigue modelar una amplia clase de sistemas mediante una notación fija. Por esta razón UML proporciona varios mecanismos de extensión del lenguaje.

Un *estereotipo* es un texto encerrado entre parentesis angulares <<texto>>, que se añade a un elemento UML, como una clase o una asociación, se usa para crear nuevos tipos de bloque de construcción que se necesitan en el dominio.

Una *restricción* es una regla que se añade a un bloque de construcción UML, permite representar fenómenos. La notación es un texto entre corchetes.

8.3 Diseñando con UML.

8.3.1 Obtención de requerimientos

8.3.1.1 Identificación de los actores

Es el primer paso de la obtención de requerimientos; sirve para definir las fronteras del sistema y para encontrar todas las perspectivas

desde las cuales los desarrolladores necesitan considerarlo. En una compañia, por lo general, ya existen la mayoría de los actores correspondiéndose a los papeles dentro de la organización. Como se dijo los actores representan entidades externas que interactúan con el sistema, son abstracciones de los papeles y no necesariamente tienen una correspondencia directa con personas.

Para tratar de identificar a los actores se pueden hacer las siguientes preguntas:

- ¿Cuáles son los grupos de usuarios apoyados por el sistema para realizar su trabajo?
- ¿Cuales son los grupos de usuarios que ejecutan las funciones principales del sistema?
- ¿Cuales son los grupos de usuarios que realizan funciones secundarias? (administración, etc).
- ¿Interactuará el sistema con algún otro sistema hardware o software.

El siguiente paso es determinar la funcionalidad a la que tiene acceso cada actor, información que se puede extraer usando escenarios y formalizando el uso de casos de uso.

En una aplicación de gestión de una biblioteca comarcal los actores que podríamos identificar son: socios, lectores , el bibliotecario y bibliotecas ajenas (prestamo interbibliotectario).

8.3.1.2 Identificación de escenarios

Un escenario es una descripción concreta e informal de una sola característica del sistema desde el punto de vista de un sólo actor. Los escenarios pueden tener muchos usos diferentes durante la obtención de requerimientos y durante otras actividades del ciclo de vida, algunas características son:

- Los escenarios describen una situación real, pueden ser validados con los usuarios.
- Los escenarios describen un sistema futuro, pueden usarse como prototipos baratos.
- Los escenarios de evaluación describen las tareas del usuario contra las que se va a evaluar el sistema.
- Los escenarios sirven como cursos prácticos para introducir a los nuevos usuarios en el sistema.

En la obtenión de requerimientos, los desarrolladores y usuarios escriben y refinan una serie de escenarios para obtener una comprensión compartida de lo que debe ser el sistema. Se suelen utilizar las siguientes preguntas para identificar escenarios:

- ¿Cuáles son las tareas que el actor quiere que realice el sistema?.
- ¿Qué información consulta el actor?, ¿Quién crea los datos?. ¿Se les puede modificar o eliminar?. ¿Quién lo hace?.
- ¿Qué cambios externos necesita informar el actor al sistema?. ¿Con cuánta frecuencia?.
- ¿Qué cambios necesita el actor que le informe el sistema?. ¿Con cuánta latencia?.

Normalmente estas preguntas se responden con los documentos existentes sobre el dominio de la aplicación. Los desarrolladores deben redactar los escenarios usando términos del dominio de la aplicación. El paso siguiente es formalizar los escenarios hacia los casos de uso.

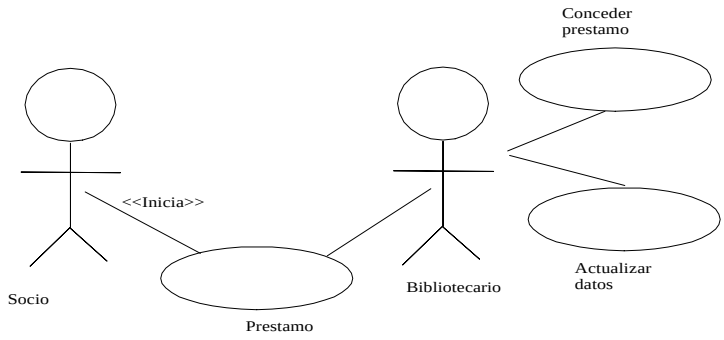
Nombre del escenario :	*Petición_interbiblioteca*
Instancias de actores:	*Roberto: bibliotecario*
participantes:	*Pepe: Socio* *Juana: Bibliotecario externo*
Flujo de eventos:	*1. Pepe está buscando un libro en el sistema. El volumen en cuestión no está disponible en la biblioteca de su pueblo, pero el mismo sí que aparece en una de la comarca perteneciente a la red interbibliotecaria.* *2. Pepe captura los datos y rellena una ficha de solicitud de prestamo interbibliotecario.* *3. Roberto recibe la petición, comprueba su corrección y realiza la petición a la biblioteca externa.* *4. Juana recibe la petición de Roberto, gestiona la viabilidad del prestamo, registra los datos y procede al envío del libro solicitado.* *5. Roberto tras recibir la aceptación del prestamo interbibliotecario hace saber a Juan que su petición ha sido aceptada y cursada.*

8.3.1.3 Identificación de casos de uso

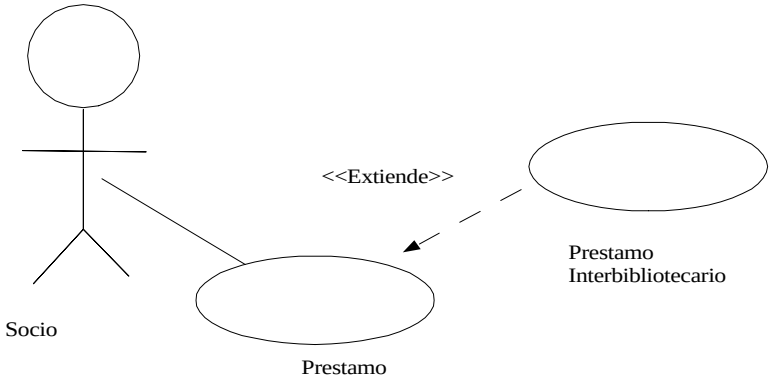
Las relaciones entre actores y casos de uso permiten que los desarrolladores y usuarios reduzcan la complejidad del modelo e incrementen su comprensibilidad. Las relaciones de comunicación entre actores y casos de uso representan el flujo de información durante el caso de uso. Se debe distinguir entre el actor que inicia el caso de uso y los demás actores con los que se comunica el caso de uso. Un caso de uso extiende otro caso de uso si puede incluir el comportamiento de la extensión bajo determinadas condiciones. Mediante la extensión

separamos el flujo común de eventos del excepcional. Podemos evitar redundancias en los casos de uso mediante las relaciones de inclusión.

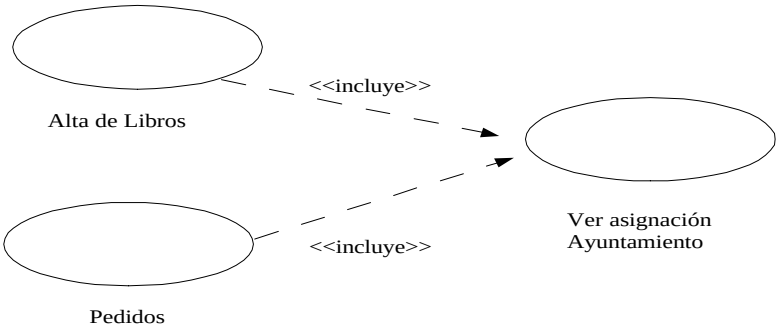
Relación de comunicación.



Relación extendida Crédito interbibliotecario



Relación de inclusión



8.3.2 Documentación de la obtención de requerimientos

UML no es una notación exclusivamente gráfica, los resultados de la actividad de obtención de requerimientos se documenta en el Documento de Análisis de Requerimientos (RAD), que describirá por completo al sistema desde el punto de vista de los requerimientos funcionales y no funcionales, y sirve de base contractual entre el cliente y los desarrolladores (no hay que olvidar que el software de código abierto también puede [y debería] tener clientes). Una plantilla para un RAD puede ser la siguiente:

1. Introducción
   1.1 Propósito del sistema
   1.2 Alcance del sistema
   1.3 Objetivos y criterios de éxito del proyecto
   1.4 Definiciones, siglas y abreviaturas
   1.5 Referencias

1.6 Panorama

2. Sistema actual
3. Sistema propuesto
   3.1 Panorama
   3.2 Requerimientos funcionales
   3.3 Requerimientos no funcionales
      3.3.1 Interfaz de usuario
      3.3.2 Documentación
      3.3.3 Consideraciones Hardware
      3.3.4 Características de desempeño
      3.3.5 Manejo de errores y condiciones extremas
      3.3.6 Cuestiones de calidad
      3.3.7 Modificaciones al sistema
      3.3.8 Ambiente físico
      3.3.9 Cuestiones de seguridad
      3.3.10 Cuestiones de recursos
   3.4 Pseudorequerimientos (restricciones de diseño e implementación impuestas)
   3.5 Modelos del sistema
      3.5.1 Escenarios
      3.5.2 Modelo de casos de uso
      3.5.3 Modelo de objetos
         3.5.3.1 Diccionario de datos
         3.5.3.2 Diagramas de clase
      3.5.4 Modelos dinámicos
      3.5.5 Interfaz de usuario: Rutas de navegación y maquetas de pantallas

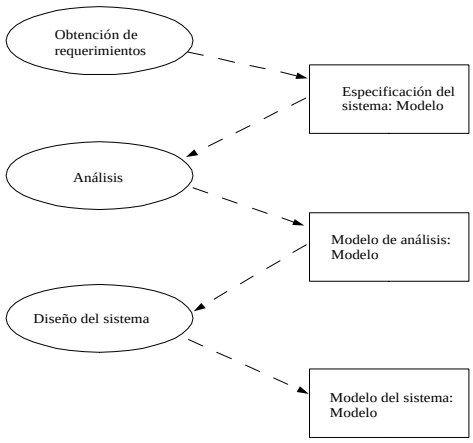
8.4 Análisis y UML

8.4.1 Panorama de análisis

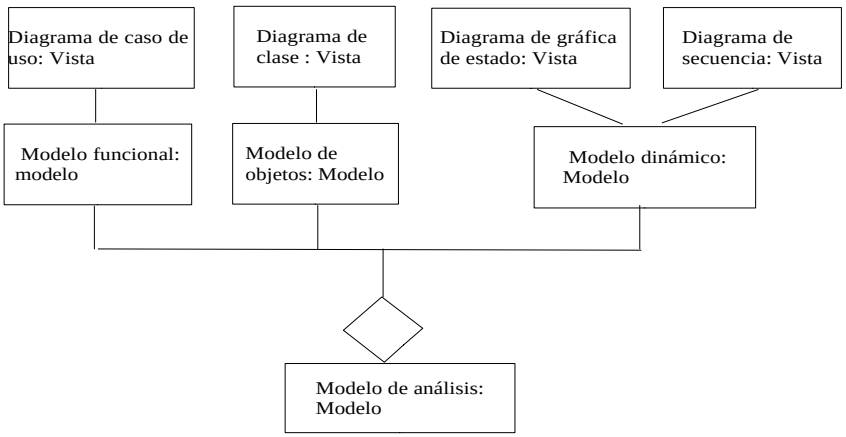
El modelo de análisis utilizando UML está compuesto por tres modelos individuales: el modelo funcional (casos de uso y escenarios), el modelo de objetos de análisis (diagramas de clase y objetos), y el modelo dinámico (gráficas de estado y diagramas de secuencia). Tras obtener los requerimientos de los usuarios, descritos como casos de uso y escenarios, se debe refinar el modelo funcional y derivar el modelo de objetos y el dinámico, con el fin de obtener una descripción más precisa y completa conforme se añaden detalles al modelado de análisis.

Productos de obtención de requerimientos y análisis.

Diagrama de actividad UML.



Composición del modelo de análisis:



8.4.2 Conceptos de análisis

8.4.2.1 Objetos entidad, frontera y control

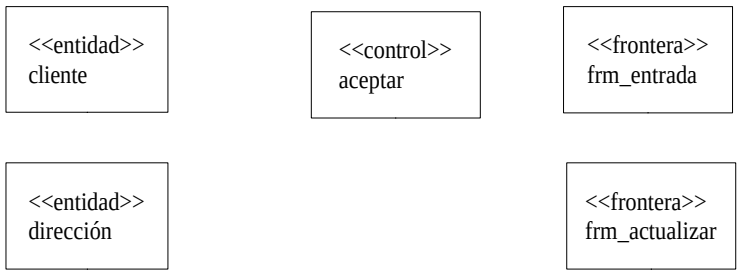
El modelo de objetos de análisis consiste en identificar los objetos de entidad, frontera y control:

Objetos de entidad: Representan la información persistente rastreada por el sistema.

Objetos frontera: representan la interacción entre los actores y el sistema.

Objetos control: representan las tareas realizadas por el usuario y soportadas por el sistema.

Modelar de esta forma proporciona una forma simple para distinguir conceptos diferentes pero relacionados. Da como resultado objetos más pequeños y especializados y se producen modelos más adaptables. Para distinguir estos objetos utilizamos estereotipos para introducir esta metainformación a los elementos de modelo. Por ejemplo:



8.4.2.2 Revisión de la multiplicidad de asociación.

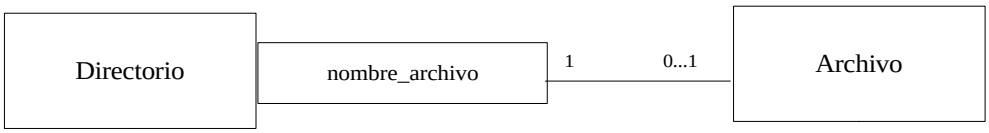
La multiplicidad indica la cantidad de vínculos que pueden originiarse desde una instancia de la clase conectada al extremo de la asociación. En UML, un entremo de una asociación puede tener como multiplicidad un conjunto de enteros arbitrarios, sin embargo las más normal son los siguientes tipos:

- Asociación uno a uno: Una persona tiene exactamente un DNI, y un DNI se corresponde con una sola persona.
- Asociación de uno a muchos: indica agregación, por ejemplo una persona puede tener varios coches, o no tener ninguno, pero el coche si existe estará puesto al nombre de una sóla persona. Un caso especial de agregación es la *composición*, donde cada componente pertenece a un todo y sólo a uno (se representa con un rombo relleno); por ejemplo un ascensor se compone de un motor, un cable y una cabina.
- Asociación de muchos a muchos: Indica la posibilidad de una cantidad arbitraria de vinculos entre instancias de dos clases. Un suministrador puede proporcionar varios productos, y un producto a su vez puede provenir de varios suministradores.

La adición de multiplicidad a las asociaciones incrementa la cantidad de información que capturamos del dominio de la aplicación o del dominio de la solución, por ello la especificación de la multiplicidad de una asociación puede ser crítica para determinar los casos de uso que se necesitan para manipular objetos del dominio de la aplicación.

8.4.2.3 Asociaciones calificadas

Un calificador permite reducir las asociaciones de uno a muchos o de muchos a muchos, de forma que el modelo sea más claro y se tengan que tomar en cuenta menos casos. Un ejemplo claro es una estructura de archivos donde un archivo pertenece a un directorio, muchos archivos pueden tener el mismo nombre en el contexto del sistema de archivos, pero sólo puede existir uno en un directorio. La representación con calificación es la siguiente:



8.4.3 Actividades de análisis: desde los casos de uso hasta los objetos

8.4.3.1 Identificación de objetos de entidad.

Tras tener identificados los objetos participantes en la obtención de requerimientos, debemos obtener los objetos entidad, una heurística comúnmente aceptada para su identificación es la siguiente:

- Términos que los desarrolladores o usuarios necesitan aclarar para comprender el caso de uso
- Nombres recurrentes en los casos de uso
- Entidades del mundo real de las que necesita llevar cuenta el sistema.
- Actividades del mundo real de las que necesita llevar cuenta el sistema.
- Casos de uso
- Fuentes o destinos de datos.

8.4.3.2 Identificación de objetos frontera

Como ya se ha comentado los objetos de frontera representan la interfaz del sistema con los actores. En cada caso de uso, cada actor interactúa con al menos, un objeto de frontera, el cuál, recopila la información del actor y la traduce a una forma neutral de interfaz que puede ser utilizada por los objetos de entidad y también por los objetos de control. A pesar de que los objetos de frontera modelan la interfaz de usuario a un nivel burdo, no describen con detalle los aspectos visuales de la interfaz de usuario, por ello no hay que llegar a detalles como *botón* o *elemento de menú,* ya que dada la evolución del interfaz a lo largo del diseño e incluso en versiones funcionales estables, la actualización del modelo de análisis cada vez que se hace un cambio visual a la interfaz consume tiempo y no produce ningún beneficio sustancial.

Como método de identificación para los objetos frontera tenemos:

- Identificar formularios y ventanas que el usuario necesita para dar datos al sistema.
- Identificar mensajes y noticias que el sistema usa para responder al usuario.
- No modelar aspectos visuales de la interfaz con objetos de frontera, usar maquetas.
- Describir las interfaces con términos de usuario.

8.4.3.3 Identificación de objetos de control.

Los objetos de control son responsables de la coordinación entre objetos de frontera y de entidad, normalmente no tienen una contraparte concreta en el mundo real. Por lo general se crean al inicio de un caso de uso y desaparecen cuando termina. Son los responsables de recopilar información de los objetos de frontera y enviarla a los objetos de entidad. Por ejemplo describe el comportamiento asociado a la secuencia de formularios.

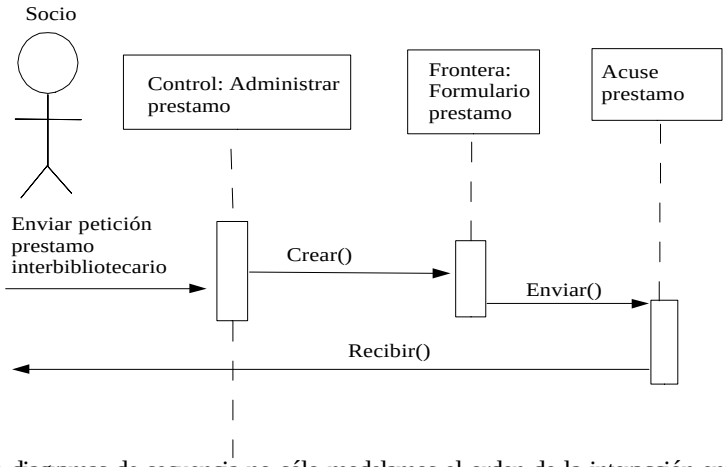
Una heurística aceptada para su identificación es:

- Identificar un objeto de control por caso de uso, si éste es completjo puede tener asociados varios objetos de control, en éste caso estudiar si se puede dividir en flujos de eventos más cortos.
- Identificar un objeto de control por actor en el caso de uso.
- La vida de un objeto de control es la del caso de uso o la de sesión de usuario. Si esta identificación no está clara normalmente es debido a que el caso de uso no tiene las condiciones de entrada y salida bien definidas.

8.4.3.4 Modelado de interacción entre objetos: diagramas de secuencia.

Un diagrama de secuencia une los casos de uso con los objetos. Muestra cómo se distribuye el comportamiento de un caso de uso (o escenario) entre sus objetos participantes. Para su trazado podemos utilizar el siguiente método:

- La primera colúmna debe corresponder al actor que empieza el caso de uso.
- La segunda columna debe ser un objeto de frontera (que usa el acto para iniciar el caso de uso).
- La tercera columna debe ser un objeto de control que maneja el resto del caso de uso.
- Los objetos de control son creados por objetos de frontera que inician casos de uso
- Los objetos de frontera son creados por objetos de control.
- Los objetos de entidad son accedidos por objetos de control y de frontera.
- Los objetos de entidad *nunca* tienen acceso a los objetos de frontera o control, esto hace que sea más fácil compartir objetos de entidad entre casos de uso.



Mediante la construcción de diagramas de secuencia no sólo modelamos el orden de la interacción entre objetos sino que también distribuimos el comportamiento del caso de uso, es decir, se asignan responsabilidades a cada objeto en forma de un conjunto de operaciones. Durante el análisis permiten identificar nuevos objetos participantes y comportamiento faltantes.

8.4.3.5 Identificación de asociaciones

Los diagramas de clase permiten que los desarrolladores describan la conectividad especial de los objetos, una asociación muestra una relación entre dos o más clases. Las asociaciones tienen varias propiedades:

- Un *nombre* para describir la asociación entre las dos clases
- Un *papel* a cada extremo, que identifica la operación de cada clase con respecto a las asociaciones.
- Una *multiplicidad* a cada extremo, que identifica la cantidad posible de instancias.

En un principio, las asociaciones entre objetos de entidad son lo más importante, ya que se revela más información acerca del dominio de la aplicación., cada asociación debe tener un nombre y papeles asignados a cada extremo.

Heurística para la identificación de asociaciones:

- Examinar frases verbales.
- Nombrar con precisión a las asociaciones y papeles
- Usar calificadores con tanta frecuencia como sea posible para identificar espacios de nombres y atributos principales.
- Eliminar cualquier asociación que pueda derivarse de otras asociaciones.
- No valorar la multiplicidad hasta disponer de un conjunto de asociaciones estable.
- Hay que evitar el exceso de asociaciones, pueden producir un modelo ilegible.

8.4.3.6 Identificación de atributos.

Los atributos son propiedades de objetos individuales, cuando éstas se identifican, sólo deben considerarse los atributos relevantes para el sistema. Las propiedades que están representadas por objetos no son atributos. Por esto es importante indentificar la mayor cantidad de asociaciones posibles antes de identificar los atributos para no confundirlos con objetos. Los atributos tienen:

- Un *nombre* que los identifica dentro de un objeto.
- Una *descripción* breve.
- Un *tipo* que describe los valores legales que puede adoptar.

Los atributos representan la parte menos estable del modelo de objetos. Con mucha frecuencia se descubren o añaden más adelante en el desarrollo. A menos que estos atributos añadidos estén asociados con funcionalidad adicional, no implican cambios importantes en la estructura del objeto (ni del sistema).

Un modelo que permite identificar los objetos (Rumbaugh):

- Examinar las frases posesivas.
- Representar el estado guardado como atributo de un objeto de entidad.
- Describir cada atributo
- No representar atributos como objetos, usar una asociación
- No refinar antes de tener una estructura estable del objeto.

8.4.3.7 Modelado del comportamiento no trivial

Los diagramas de secuencia se usan para distribuir un comportamiento entre objetos y para identificar operaciones. Los diagramas de secuencia representan el comportamiento del sistema desde la perspectiva de un solo caso de uso. Los diagramas de gráfica de estado representan el comportamiento desde la perspectiva de un solo objeto. La visión del comportamiento desde la perspectiva de cada objeto permite que el desarrollador identifique casos de uso faltantes y que construya una descripción más forma del comportamiento del objeto. No es necesario construir gráficas de estado para cada clase del sistema, sólo es necesario para los objetos que tienen una vida extendida y un comportamiento no trivial.

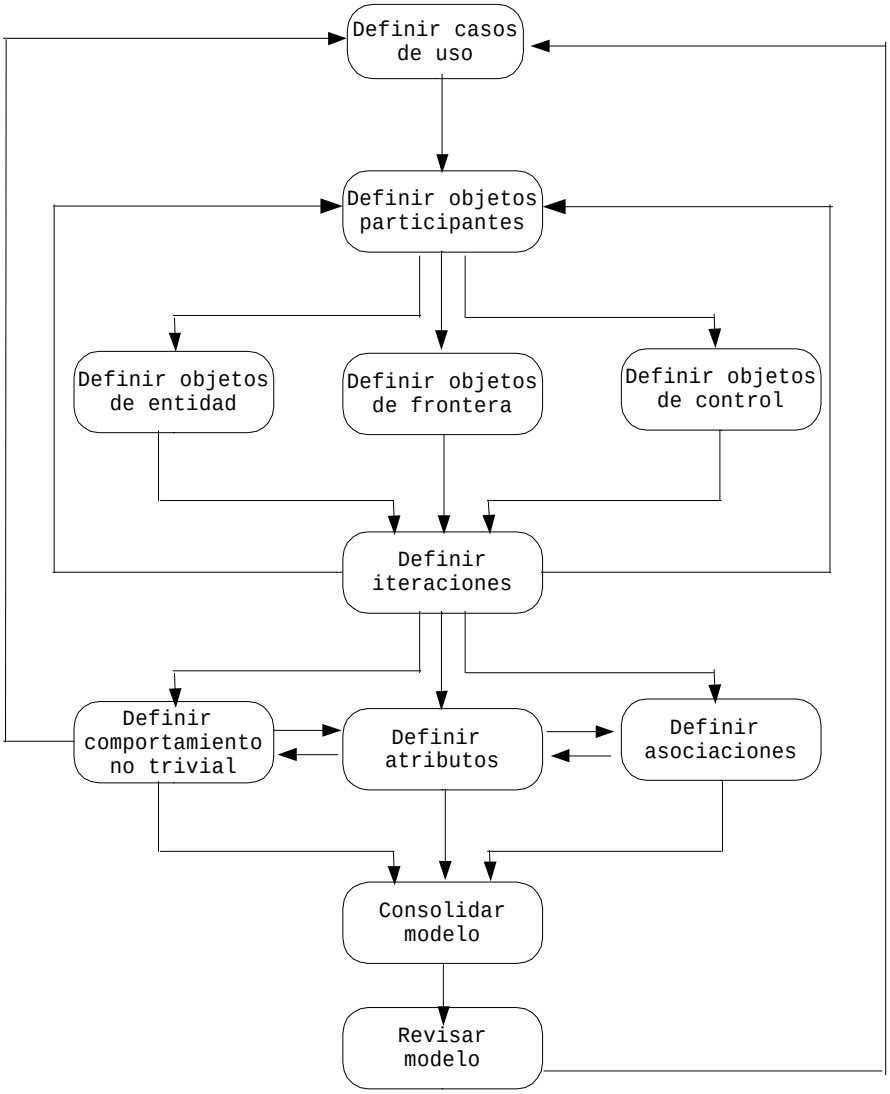
8.4.3.8 Modelado de las relaciones de generalización entre objetos

La generalización se usa para eliminar redundancias en el modelo de análisis. Si dos o más clases comparten atributos o comportamientos, la similitudes se consolidan en una super clase.

8.4.3.9 Resumen del análisis

La actividad de requerimientos es muy iterativa e incremental, conforme crece la descripción del sistema y los requerimientos se hacen más concretos, los desarrolladores necesitan extender y modificar el modelo de análisis de forma ordenada para administrar la complejidad de la información.

La siguiente figura indica la secuencia típica de las actividades de análisis mediante un diagrama de actividades UML:



8.5 Actividades de diseño en UML

8.5.1 Panorama de diseño

Tras la especificación de requerimientos y la fase de análisis que hemos realizado tenemos los siguientes productos:

- Un conjunto de requerimientos no funcionales y restricciones
- Un modelo de casos de uso
- Un modelo de objetos
- Un diagrama de secuencia

Nos falta, pues, la manera en la que se debe realizar el sistema. El primer paso del diseño se da en esta dirección, y nos dará como resultado:

- Una lista de objetivos de diseño que describe las cualidades del sistema que deben optimizar los desarrolladores.
- Una arquitectura del software, que describe la descomposición en subsitemas.

8.5.2 Conceptos del diseño de sistemas

Para reducir la complejidad del dominio de la aplicación identificamos partes más pequeñas, llamadas clases, y las organizamos en paquetes. Igualmente, para reducir la complejidad del dominio de solución descomponemos un sistema en partes más simples, llamadas subsistemas, compuestas de varias clases del dominio de solución.

Un subsistema se caracteriza por los servicios que proporciona a otros subsistemas. Un servicio es un conjunto de operaciones relacionadas que comparten un propósito común. El conjunto de operaciones de un subsistema que está disponible para otros subsistemas, forma la interfaz del subsistema, que incluye el nombre de las operaciones, sus parámetros, sus tipos y sus valores de retorno. El diseño de sistemas se enfoca a la definición de los servicios proporcionados por cada subsistema, el diseño de objetos se enfocará a la definición de las interfaces de los subsistemas (tipo de parámetros y valor de retorno). Esto no es gratuito ya que una buena definición de la interfaz, con el mínimo de información sobre su implementación, minimiza el impacto de los cambios. Para el desarrollo de subsistemas debe de tenerse en cuenta los aspectos de coherencia y acoplamiento indicados en este documento (punto 5.2.4 Diseño Modular Efectivo).

8.5.3 Actividades del diseño de sistemas.

Las principales actividades a realizar son las siguientes:

- Identificar los objetivos de diseño para los requerimientos no funcionales.
- Diseño de la descomposición inicial en subsistemas
- Establecer la correspondencia entre los subsistemas y los procesadores y componentes.
- Decidir el sistema de almacenamiento de datos
- Definición de las políticas de control de acceso.
- Selección del flujo de control.
- Identificación de las condiciones de frontera.

8.5.4 Documentación del diseño del sistema

El diseño del sistema se plasma en el Documento de Diseño del Sistema (SDD), qué describe los objetivos del diseño propuestos para el proyecto, la descomposición en subsistemas (Diagramas de Clase en UML), correspondencia entre hardware y software (Diagramas de Despliegue UML), y demás datos obtenidos. Un modelo de plantilla puede ser el siguiente:

1 Introducción
  1.1 Propósito del sistema
  1.2 Objetivos del diseño
  1.3 Definiciones, siglas y abreviaturas.
  1.4 Referencias
  1.5 Panorama
2 Arquitectura del software actual (si procede).
3 Arquitectura del software propuesto.
  3.1 Panorama
  3.2 Descomposición en subsistemas
  3.3 Correspondencia entre hardware y software
  3.4 Administración de datos persistentes
  3.5 Control de acceso y seguridad
  3.6 Control de software global
  3.7 Condiciones de frontera
4 Servicios de subsistema
5 Glosario

De todos estos apartados, el único que no ha sido tratado es la tercera sección, *Arquitectura del software propuesto*, que está dividida en siete subsecciones:

- Panorama: Presenta a vista de pájaro la arquitectura del software y describe en forma breve la asignación de funcionalidad de cada subsistema.
- Descomposición en subsistemas: Describe la descomposición y las responsabilidades de cada uno de ellos. Es el producto principal.
- Correspondencia entre hardware y software: Describe la asignación de subsistemas al hardware así como la posible reutilización del software.
- Administración de datos persistentes: Describe los datos persistentes guardados por el sistema y la infraestructura de administración de datos que se requiere para ello. Incluye la descripción de esquema de datos, selección de una base de datos (si se precisa) y el encapsulado de la base de datos.
- Control de acceso y seguridad: Describe la manera en que se implementa el control de software global, es decir la sincronización de los subsistemas y concurrencia.
- Condiciones de frontera: Comportamiento del sistema en el arranque, apagado y errores.