



UNIVERSIDAD NACIONAL DE  
EDUCACIÓN A DISTANCIA

INGENIERÍA TÉCNICA en INFORMÁTICA de SISTEMAS y de GESTIÓN  
Plan de estudios en extinción  
CÓDIGO CARRERA: 40=SISTEMAS y 41=GESTIÓN  
CÓDIGO DE ASIGNATURA: 210=SISTEMAS y 208=GESTIÓN  
NACIONAL 1ª SEMANA



Departamento de Lenguajes  
y Sistemas Informáticos

ASIGNATURA: INGENIERÍA DEL SOFTWARE (2º CURSO)  
FECHA: 28 de mayo de 2003 Hora: 11:30 Duración: 2 horas MATERIAL: NINGUNO

---

**Todas las preguntas de este ejercicio son eliminatorias en el sentido de que debe obtener una nota mínima en cada una de ellas. En la primera parte (las preguntas teóricas que se valoran con 2'5 puntos cada una) la nota mínima es 1 punto; en la segunda parte (ejercicio de teoría aplicada que se valora con 5 puntos) la nota mínima que debe obtener es de 2 puntos.**

**¡ATENCIÓN! PONGA SUS DATOS EN LA HOJA DE LECTURA ÓPTICA QUE DEBERÁ ENTREGAR JUNTO CON EL RESTO DE LAS RESPUESTAS.**

Conteste a las preguntas teóricas, en cualquier orden, en hojas diferentes a las que utilice para la contestación de la segunda parte. En cada parte, la cantidad MÁXIMA de papel (de examen, timbrado) que puede emplear ESTÁ LIMITADA al equivalente a DOS (2) HOJAS de tamaño A4 (210 x 297 mm)

---

PRIMERA PARTE. PREGUNTAS TEÓRICAS (2'5 PUNTOS CADA UNA)

1. Deduzca y justifique en qué casos es mejor aplicar un ciclo de vida en cascada respecto a uno en espiral y viceversa.
2. Dé una breve definición de la fase de diseño. Describa cuáles son los objetivos últimos que se pretende alcanzar con la descomposición modular del diseño.

SEGUNDA PARTE. PREGUNTA DE APLICACIÓN (5 PUNTOS)

3. Con el fin de promocionar el uso del transporte público y el ocio al aire libre, RENFE ha decidido encargar la construcción de un sistema informático que asesore a sus clientes acerca de 'rutas verdes' para hacer a pie a partir de sus estaciones de tren.

El sistema recibirá periódicamente la siguiente información:

- Un informe meteorológico del Instituto Nacional de Meteorología que contendrá las previsiones climáticas para los próximos días.
- Datos referentes a las estaciones de tren, horarios y precios de billetes. Esta información será suministrada por RENFE.
- Se ha encargado a la empresa "Viajes Najarra" la elaboración e introducción en el sistema de las rutas verdes. Para ello, la empresa podrá solicitar del sistema un informe de las estaciones de RENFE existentes.

Los clientes introducirán en el sistema sus preferencias. A partir de estas y los datos antes descritos, se construirá un informe con las rutas aconsejadas.

*Analice el sistema mediante DFDs (Diagramas de Flujo de Datos), desarrollando exclusivamente los DFDs de nivel 0 y 1.*



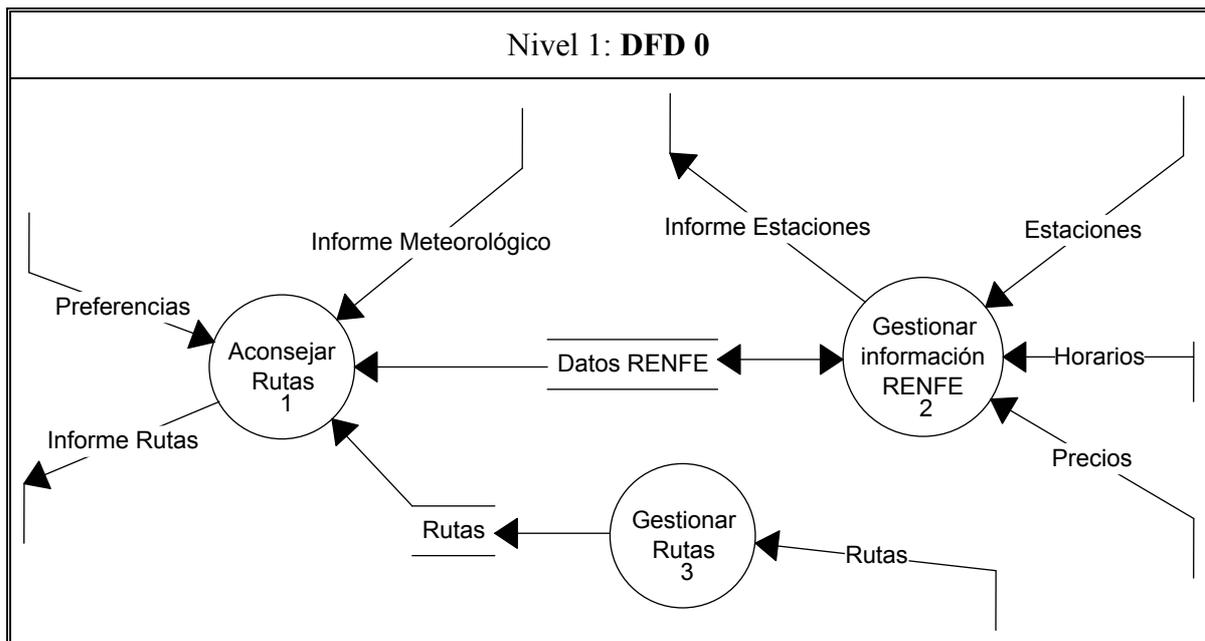
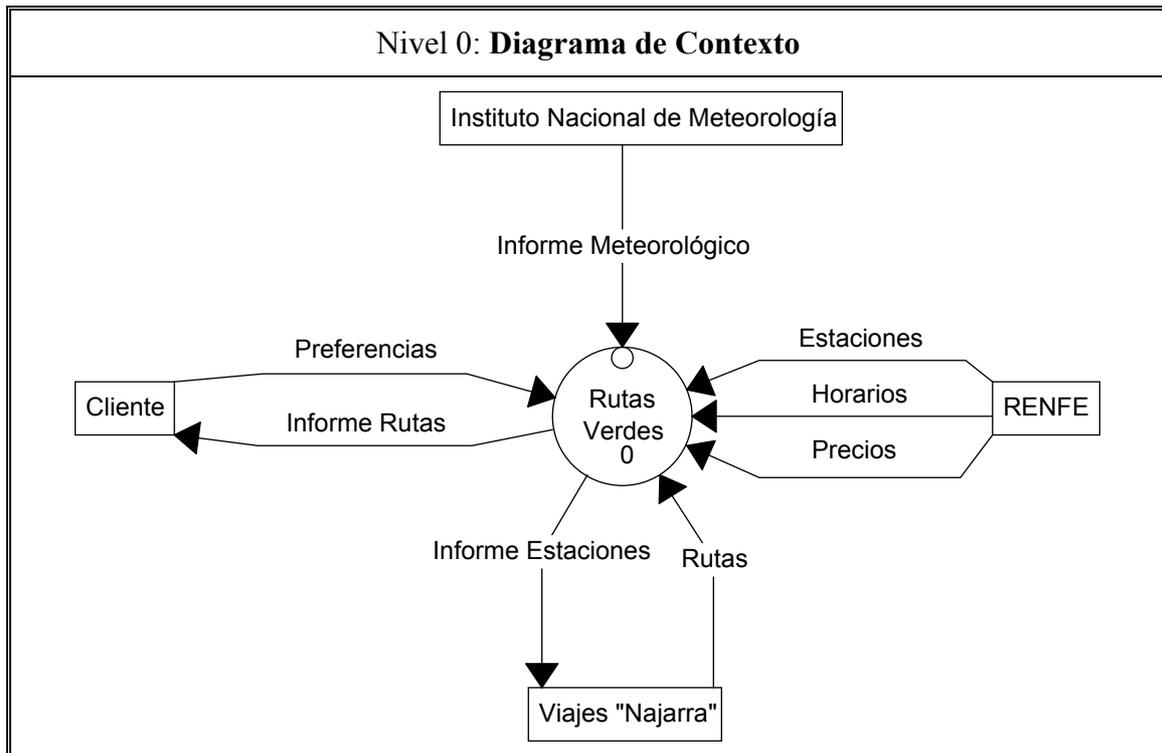
Universidad Nacional  
de Educación a  
Distancia

## Solución propuesta por el equipo docente

Autores: José Félix Estívariz López, Rubén Heradio Gil, Juan Antonio Mascarell Estruch



Departamento de  
Lenguajes y Sistemas  
Informáticos





ASIGNATURA: INGENIERÍA DEL SOFTWARE (2º CURSO)

FECHA: 11 de junio de 2003

Hora: 11:30

Duración: 2 horas

MATERIAL: NINGUNO

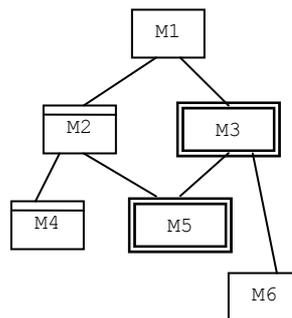
**Todas las preguntas de este ejercicio son eliminatorias en el sentido de que debe obtener una nota mínima en cada una de ellas. En la primera parte (las preguntas teóricas que se valoran con 2'5 puntos cada una) la nota mínima es 1 punto; en la segunda parte (ejercicio de teoría aplicada que se valora con 5 puntos) la nota mínima que debe obtener es de 2 puntos.**

**¡ATENCIÓN! PONGA SUS DATOS EN LA HOJA DE LECTURA ÓPTICA QUE DEBERÁ ENTREGAR JUNTO CON EL RESTO DE LAS RESPUESTAS.**

Conteste a las preguntas teóricas, en cualquier orden, en hojas diferentes a las que utilice para la contestación de la segunda parte. En cada parte, la cantidad MÁXIMA de papel (de examen, timbrado) que puede emplear ESTÁ LIMITADA al equivalente a DOS (2) HOJAS de tamaño A4 (210 x 297 mm)

PRIMERA PARTE. PREGUNTAS TEÓRICAS (2'5 PUNTOS CADA UNA)

1. Enumere, defina y explique brevemente las fases que componen el ciclo de vida en cascada.
2. Dado el siguiente Diagrama de Abstracciones:



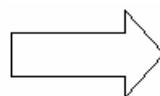
Establezca una comparativa entre las estrategias de integración ascendente y descendente, indicando el orden en el que se construirían los módulos del diagrama para cada una de ellas, así como sus ventajas e inconvenientes.

SEGUNDA PARTE. PREGUNTA DE APLICACIÓN (5 PUNTOS)

3. Se desea construir un sistema que reciba como entrada una lista ordenada de líneas, cada una de las cuales está formada por una lista ordenada de palabras, que a su vez será una lista ordenada de caracteres. Sobre cada línea se pueden realizar rotaciones, que consisten en eliminar la primera palabra y concatenarla al final de la línea. El sistema devolverá como resultado una lista con las posibles rotaciones de todas las líneas ordenadas alfabéticamente (incluyendo las rotaciones nulas).

Por ejemplo,

voy a cenar  
al restaurante



a cenar voy  
al restaurante  
cenar voy a  
restaurante al  
voy a cenar

Diseñe el sistema utilizando diagramas de abstracciones.



Universidad Nacional  
de Educación a  
Distancia

## Soluciones propuestas por el equipo docente

Autores: José Félix Estívariz López, Rubén Heradio Gil, Juan Antonio Mascarell Estruch

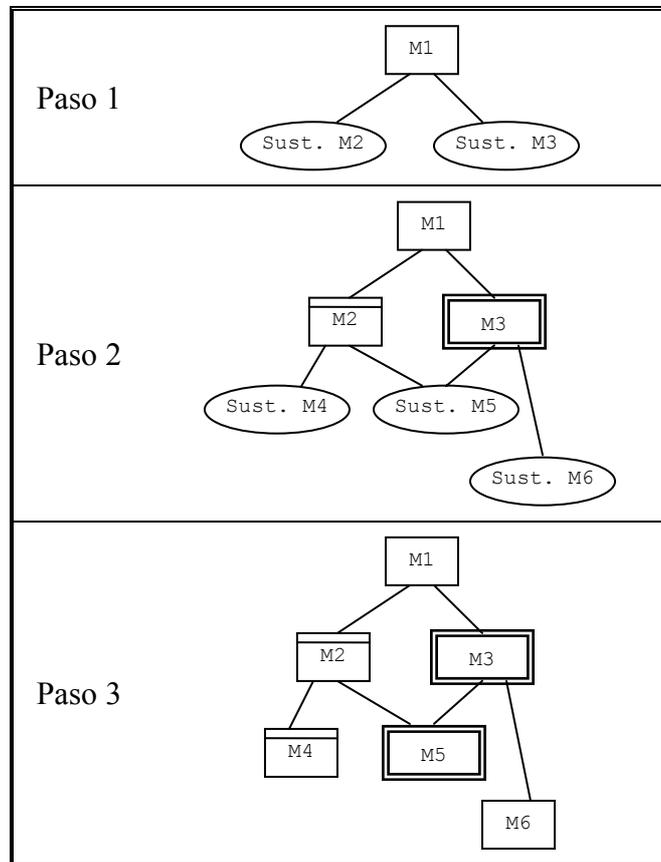


Departamento de  
Lenguajes y  
Sistemas  
Informáticos

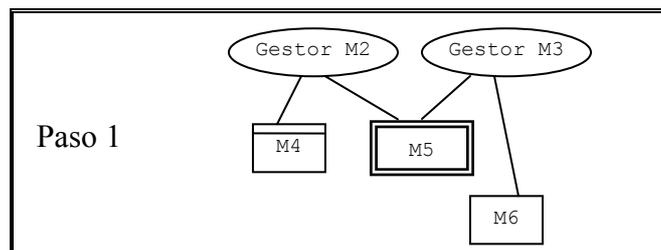
### Solución a la pregunta teórica 2

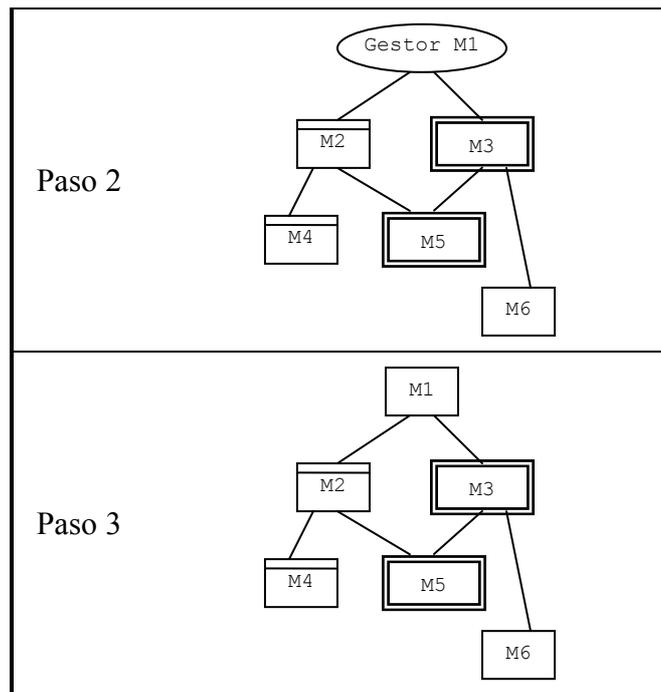
Examinemos el orden de realización de los módulos para cada estrategia de integración:

#### Integración Descendente



#### Integración Ascendente





La siguiente tabla resume las ventajas e inconvenientes derivados del uso de las integraciones descendente y ascendente en el ejemplo que nos ocupa:

	<b>I. Descendente</b>	<b>I. Ascendente</b>
<b>¿Facilita una visión general de la aplicación desde el principio?</b>	Sí	No
<b>Número de elementos de código desechables a construir</b>	5 sustitutos o stubs	3 gestores o drivers
<b>¿Facilita el ensayo de situaciones especiales para los módulos?</b>	No	Sí
<b>¿Facilita el trabajo en paralelo?*</b>	Sí	Sí

(\*): Aunque generalmente la integración ascendente propicia en mayor grado el trabajo en paralelo que la integración descendente, en este ejemplo, las dos estrategias de integración facilitan el trabajo en paralelo en el mismo grado.

### **Solución a la pregunta de aplicación**

En una primera aproximación, aplicaremos el método de Abbott para determinar las abstracciones que componen el diseño. Para ello, marcaremos en rojo los sustantivos (candidatos a ser tipos de datos) y en azul los verbos (candidatos a ser operaciones).

*Se desea construir un sistema que reciba como entrada una **lista ordenada de líneas**, cada una de las cuales está formada por una **lista ordenada de palabras**, que a su vez será una **lista ordenada de caracteres**. Sobre cada línea se pueden **realizar rotaciones**, que consisten en **eliminar la primera palabra** y **concatenarla** al final de la línea. El sistema devolverá como resultado una **lista con las posibles rotaciones de todas las líneas ordenadas alfabéticamente** (incluyendo las rotaciones nulas).*

A partir de este marcado elaboramos una doble lista con los elementos correspondientes a datos y a operaciones.

DATOS	OPERACIONES
<ul style="list-style-type: none"> <li>• lista ordenada de líneas</li> <li>• lista ordenada de palabras</li> <li>• lista ordenada de caracteres</li> <li>• lista con las posibles rotaciones de todas las líneas ordenadas alfabéticamente</li> </ul>	<ul style="list-style-type: none"> <li>• realizar rotaciones</li> <li>• eliminar la primera palabra</li> <li>• concatenar la primera palabra al final de la línea</li> </ul>



### ¿Qué son los Tipos Abstractos de Datos?

*Un Tipo Abstracto de Datos (TAD) es una entidad que agrupa la estructura de un tipo de datos con las operaciones necesarias para su manejo.*

*Tradicionalmente, en la programación imperativa se optaba por separar los programas en dos partes: la de proceso y la de datos (ejemplos de lenguajes imperativos son FORTRAN, COBOL, PASCAL, BASIC...). La parte de proceso accedía y operaba directamente sobre los datos.*

*Esta separación produce una independencia funcional muy baja. Un ejemplo que ilustra esto es el “efecto 2000”, donde la simple adición de dígitos al formato de las fechas supuso realizar muchas modificaciones en la parte de proceso.*

*Para solventar estos problemas surgió el concepto de Tipo Abstracto de Datos, que junta la representación de los datos con la parte de proceso que los manipula. Los TADs ocultan la representación de los datos, que sólo es accesible desde las operaciones. De esta forma, un cambio en la representación de los datos de un TAD no se propaga a todo el programa, sino solamente a las operaciones del TAD.*

*Antiguamente, las estructuras de datos se venían definiendo por su representación. Sin embargo, el paso clave hacia la abstracción de los datos es invertir este punto de vista: olvidar por el momento la representación y considerar que las operaciones en sí mismas definen la estructura de datos.*

*NOTA: un error grave cometido por muchos alumnos consiste en diseñar una operación fuera del TAD que le corresponde. Por ejemplo, en este examen algunos alumnos proponían utilizar un TAD llamado “Rotaciones” y una abstracción funcional separada llamada “ProducirRotaciones”.*

Tras analizar la doble lista sobre datos y operaciones se proponen tres datos encapsulados:

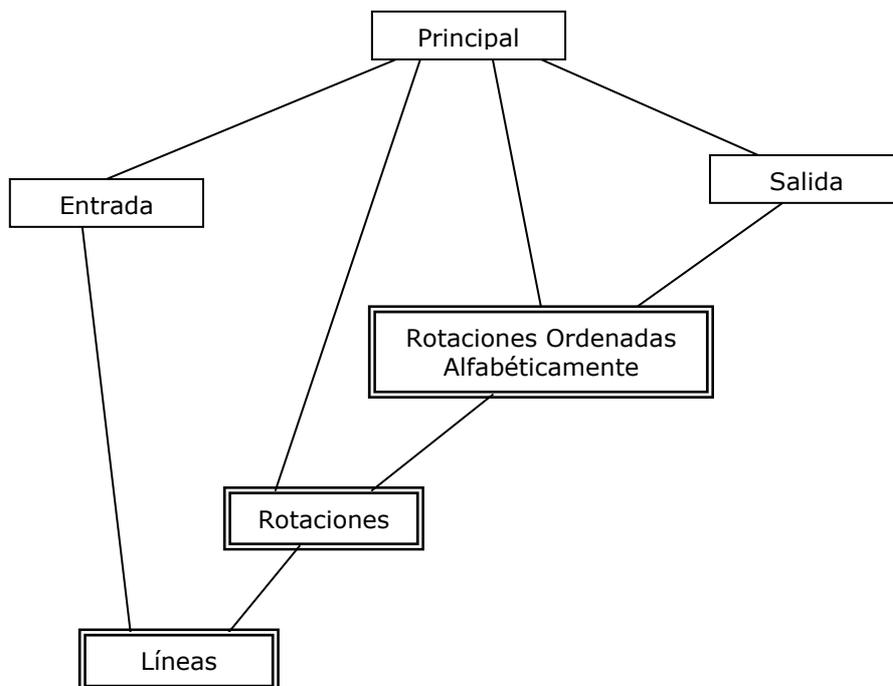


*Los TAD propuestos son datos encapsulados por que sólo existirá una variable o elemento de cada tipo en el sistema.*

<p><b>DATO: Líneas</b> Operaciones: Introducir Obtener</p>	<p><i>Se ha tomado la decisión de aglutinar los datos “lista ordenada de líneas”, “lista ordenada de palabras” y “lista ordenada de caracteres” en un sólo dato encapsulado, ya que el</i></p>
--	--

EstaVacía	<p><i>crecimiento innecesario del número de abstracciones aumentaría la complejidad del sistema. Ésta abstracción consta de tres operaciones:</i></p> <ul style="list-style-type: none"> <li>• <i>“Introducir”</i>: se encargará de recibir una línea y almacenarla en el dato encapsulado.</li> <li>• <i>“Obtener”</i>: proporcionará una línea.</li> <li>• <i>“EstaVacía”</i>: indicará si hay alguna línea almacenada dentro del dato encapsulado.</li> </ul>
<b>DATO: Rotaciones</b> Operaciones: ProducirRotaciones Obtener EstaVacía	 <i>“Rotaciones” es un dato encapsulado que producirá y almacenará todas las posibles rotaciones realizadas sobre las líneas que el sistema reciba como entrada. La operación “ProducirRotaciones” engloba a las operaciones “realizar rotaciones”, “eliminar la primera palabra” y “concatenar la primera palabra al final de la línea” que aparecen en la doble lista de datos y operaciones.</i>
<b>DATO: RotacionesOrdenadasAlfabéticamente</b> Operaciones: Ordenar Obtener EstaVacía	 <i>Éste es un dato encapsulado que producirá y almacenará las rotaciones realizadas sobre las líneas ordenadas alfabéticamente.</i>

El diagrama de abstracciones propuesto para construir el sistema solicitado es el siguiente:



Como puede verse, el diseño consta de la abstracción funcional “Principal” que se encargará de coordinar al resto de las abstracciones. Además, el diseño posee las siguientes abstracciones funcionales:

- “Entrada”: se encargará de obtener los datos del exterior (en el enunciado no se especifica la forma en que el usuario introducirá los datos) y de almacenarlos en “Lineas”. Para ello, empleará la operación “Introducir” proporcionada por dicho dato encapsulado.
- “Salida”: mostrará el resultado final de procesar los datos de entrada. Para este fin, utilizará la operación “Obtener” del dato encapsulado “RotacionesOrdenadasAlfabéticamente”.

 Uno de los objetivos de la fase de diseño es facilitar el mantenimiento. El diseño propuesto consigue este objetivo en gran medida. Piense el lector en posibles modificaciones que pueden producirse sobre las especificaciones del sistema y verá que la parte del diseño afectada es mínima y está claramente localizada. A modo de ejemplo, la siguiente tabla propone algunas modificaciones y muestra las partes del sistema que se verían afectadas:

<i>Modificación</i>	<i>Partes afectadas</i>
<i>Presentar el resultado de la ordenación en un nuevo formato gráfico</i>	<i>“Salida”</i>
<i>La ordenación de las rotaciones deja de ser alfabética y pasa a ser de otro tipo</i>	<i>“RotacionesOrdenadasAlfabéticamente”</i>
<i>Las rotaciones se efectuarán de una nueva manera</i>	<i>“Rotaciones”</i>



UNIVERSIDAD NACIONAL DE  
EDUCACIÓN A DISTANCIA

INGENIERÍA TÉCNICA en INFORMÁTICA de SISTEMAS y de GESTIÓN  
Plan de estudios en extinción  
CÓDIGO CARRERA: 40=SISTEMAS y 41=GESTIÓN  
CÓDIGO DE ASIGNATURA: 210=SISTEMAS y 208=GESTIÓN  
EXTRANJERO ORIGINAL



Departamento de Lenguajes  
y Sistemas Informáticos

ASIGNATURA: INGENIERÍA DEL SOFTWARE (2º CURSO)

FECHA: 11 de junio de 2003

Hora: 11:30

Duración: 2 horas

MATERIAL: NINGUNO

**Todas las preguntas de este ejercicio son eliminatorias en el sentido de que debe obtener una nota mínima en cada una de ellas. En la primera parte (las preguntas teóricas que se valoran con 2'5 puntos cada una) la nota mínima es 1 punto; en la segunda parte (ejercicio de teoría aplicada que se valora con 5 puntos) la nota mínima que debe obtener es de 2 puntos.**

**¡ATENCIÓN! PONGA SUS DATOS EN LA HOJA DE LECTURA ÓPTICA QUE DEBERÁ ENTREGAR JUNTO CON EL RESTO DE LAS RESPUESTAS.**

Conteste a las preguntas teóricas, en cualquier orden, en hojas diferentes a las que utilice para la contestación de la segunda parte. En cada parte, la cantidad MÁXIMA de papel (de examen, timbrado) que puede emplear ESTÁ LIMITADA al equivalente a DOS (2) HOJAS de tamaño A4 (210 x 297 mm)

PRIMERA PARTE. PREGUNTAS TEÓRICAS (2'5 PUNTOS CADA UNA)

1. Deduzca y justifique en qué casos es mejor aplicar un ciclo de vida en cascada respecto a uno en espiral y viceversa.
2. Explique la relevancia en el diseño de la cohesión y del grado de acoplamiento entre módulos.

SEGUNDA PARTE. PREGUNTA DE APLICACIÓN (5 PUNTOS)

3. Se desea construir un sistema que reciba como entrada una lista ordenada de líneas, cada una de las cuales está formada por una lista ordenada de palabras, que a su vez será una lista ordenada de caracteres. Sobre cada línea se pueden realizar rotaciones, que consisten en eliminar la primera palabra y concatenarla al final de la línea. El sistema devolverá como resultado una lista con las posibles rotaciones de todas las líneas ordenadas alfabéticamente (incluyendo las rotaciones nulas).

Por ejemplo,

voy a cenar  
al restaurante



a cenar voy  
al restaurante  
cenar voy a  
restaurante al  
voy a cenar

*Diseñe el sistema utilizando diagramas de abstracciones.*



Universidad Nacional  
de Educación a  
Distancia

## Solución propuesta por el equipo docente

Autores: José Félix Estívariz López, Rubén Heradio Gil, Juan Antonio Mascarell Estruch



Departamento de  
Lenguajes y  
Sistemas  
Informáticos

En una primera aproximación, aplicaremos el método de Abbott para determinar las abstracciones que componen el diseño. Para ello, marcaremos en rojo los sustantivos (candidatos a ser tipos de datos) y en azul los verbos (candidatos a ser operaciones).

*Se desea construir un sistema que reciba como entrada una **lista ordenada de líneas**, cada una de las cuales está formada por una **lista ordenada de palabras**, que a su vez será una **lista ordenada de caracteres**. Sobre cada línea se pueden **realizar rotaciones**, que consisten en **eliminar la primera palabra** y **concatenarla** al final de la línea. El sistema devolverá como resultado una **lista con las posibles rotaciones de todas las líneas ordenadas alfabéticamente** (incluyendo las rotaciones nulas).*

A partir de este marcado elaboramos una doble lista con los elementos correspondientes a datos y a operaciones.

DATOS	OPERACIONES
<ul style="list-style-type: none"><li>• lista ordenada de líneas</li><li>• lista ordenada de palabras</li><li>• lista ordenada de caracteres</li><li>• lista con las posibles rotaciones de todas las líneas ordenadas alfabéticamente</li></ul>	<ul style="list-style-type: none"><li>• realizar rotaciones</li><li>• eliminar la primera palabra</li><li>• concatenar la primera palabra al final de la línea</li></ul>



### ¿Qué son los Tipos Abstractos de Datos?

*Un Tipo Abstracto de Datos (TAD) es una entidad que agrupa la estructura de un tipo de datos con las operaciones necesarias para su manejo.*

*Tradicionalmente, en la programación imperativa se optaba por separar los programas en dos partes: la de proceso y la de datos (ejemplos de lenguajes imperativos son FORTRAN, COBOL, PASCAL, BASIC...). La parte de proceso accedía y operaba directamente sobre los datos.*

*Esta separación produce una independencia funcional muy baja. Un ejemplo que ilustra esto es el “efecto 2000”, donde la simple adición de dígitos al formato de las fechas supuso realizar muchas modificaciones en la parte de proceso.*

*Para solventar estos problemas surgió el concepto de Tipo Abstracto de Datos, que junta la representación de los datos con la parte de proceso que los manipula. Los TADs ocultan la representación de los datos, que sólo es accesible desde las operaciones. De esta forma, un cambio en la representación de los datos de un TAD no se propaga a todo el programa, sino solamente a las operaciones del TAD.*

*Antiguamente, las estructuras de datos se venían definiendo por su representación. Sin embargo, el paso clave hacia la abstracción de los datos es invertir este punto de vista: olvidar por el momento la representación y considerar que las operaciones en sí*

*mismas definen la estructura de datos.*

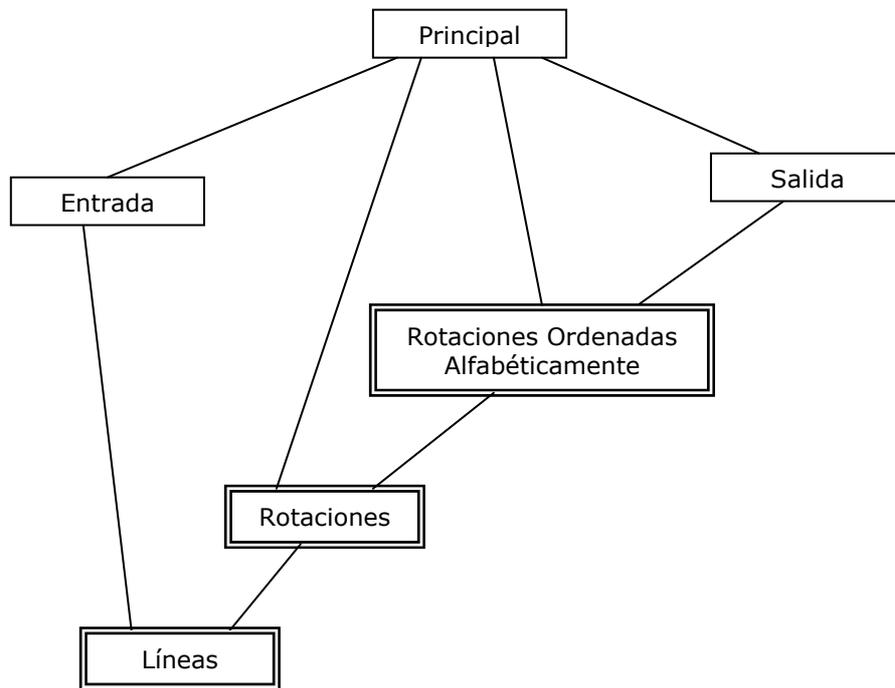
*NOTA: un error grave cometido por muchos alumnos consiste en diseñar una operación fuera del TAD que le corresponde. Por ejemplo, en este examen algunos alumnos proponían utilizar un TAD llamado “Rotaciones” y una abstracción funcional separada llamada “ProducirRotaciones”.*

Tras analizar la doble lista sobre datos y operaciones se proponen tres datos encapsulados:

 *Los TAD propuestos son datos encapsulados por que sólo existirá una variable o elemento de cada tipo en el sistema.*

<p><b>DATO: Líneas</b> Operaciones: Introducir Obtener EstaVacía</p>	<p> <i>Se ha tomado la decisión de aglutinar los datos “lista ordenada de líneas”, “lista ordenada de palabras” y “lista ordenada de caracteres” en un sólo dato encapsulado, ya que el crecimiento innecesario del número de abstracciones aumentaría la complejidad del sistema. Ésta abstracción consta de tres operaciones:</i></p> <ul style="list-style-type: none"><li><i>• “Introducir”: se encargará de recibir una línea y almacenarla en el dato encapsulado.</i></li><li><i>• “Obtener”: proporcionará una línea.</i></li><li><i>• “EstaVacía”: indicará si hay alguna línea almacenada dentro del dato encapsulado.</i></li></ul>
<p><b>DATO: Rotaciones</b> Operaciones: ProducirRotaciones Obtener EstaVacía</p>	<p> <i>“Rotaciones” es un dato encapsulado que producirá y almacenará todas las posibles rotaciones realizadas sobre las líneas que el sistema reciba como entrada. La operación “ProducirRotaciones” engloba a las operaciones “realizar rotaciones”, “eliminar la primera palabra” y “concatenar la primera palabra al final de la línea” que aparecen en la doble lista de datos y operaciones.</i></p>
<p><b>DATO: RotacionesOrdenadasAlfabéticamente</b> Operaciones: Ordenar Obtener EstaVacía</p>	<p> <i>Éste es un dato encapsulado que producirá y almacenará las rotaciones realizadas sobre las líneas ordenadas alfabéticamente.</i></p>

El diagrama de abstracciones propuesto para construir el sistema solicitado es el siguiente:



Como puede verse, el diseño consta de la abstracción funcional “Principal” que se encargará de coordinar al resto de las abstracciones. Además, el diseño posee las siguientes abstracciones funcionales:

- “Entrada”: se encargará de obtener los datos del exterior (en el enunciado no se especifica la forma en que el usuario introducirá los datos) y de almacenarlos en “Líneas”. Para ello, empleará la operación “Introducir” proporcionada por dicho dato encapsulado.
- “Salida”: mostrará el resultado final de procesar los datos de entrada. Para este fin, utilizará la operación “Obtener” del dato encapsulado “RotacionesOrdenadasAlfabéticamente”.



Uno de los objetivos de la fase de diseño es facilitar el mantenimiento. El diseño propuesto consigue este objetivo en gran medida. Piense el lector en posibles modificaciones que pueden producirse sobre las especificaciones del sistema y verá que la parte del diseño afectada es mínima y está claramente localizada. A modo de ejemplo, la siguiente tabla propone algunas modificaciones y muestra las partes del sistema que se verían afectadas:

<i>Modificación</i>	<i>Partes afectadas</i>
<i>Presentar el resultado de la ordenación en un nuevo formato gráfico</i>	<i>“Salida”</i>
<i>La ordenación de las rotaciones deja de ser alfabética y pasa a ser de otro tipo</i>	<i>“RotacionesOrdenadasAlfabéticamente”</i>
<i>Las rotaciones se efectuarán de una nueva manera</i>	<i>“Rotaciones”</i>



UNIVERSIDAD NACIONAL DE  
EDUCACIÓN A DISTANCIA

INGENIERÍA TÉCNICA en INFORMÁTICA de SISTEMAS y de GESTIÓN  
Plan de estudios en extinción  
CÓDIGO CARRERA: 40=SISTEMAS y 41=GESTIÓN  
CÓDIGO DE ASIGNATURA: 210=SISTEMAS y 208=GESTIÓN  
EXTRANJERO RESERVA



Departamento de Lenguajes  
y Sistemas Informáticos

ASIGNATURA: INGENIERÍA DEL SOFTWARE (2º CURSO)

FECHA: 14 de junio de 2003

Hora:

Duración: 2 horas

MATERIAL: NINGUNO

**Todas las preguntas de este ejercicio son eliminatorias en el sentido de que debe obtener una nota mínima en cada una de ellas. En la primera parte (las preguntas teóricas que se valoran con 2'5 puntos cada una) la nota mínima es 1 punto; en la segunda parte (ejercicio de teoría aplicada que se valora con 5 puntos) la nota mínima que debe obtener es de 2 puntos.**

**¡ATENCIÓN! PONGA SUS DATOS EN LA HOJA DE LECTURA ÓPTICA QUE DEBERÁ ENTREGAR JUNTO CON EL RESTO DE LAS RESPUESTAS.**

Conteste a las preguntas teóricas, en cualquier orden, en hojas diferentes a las que utilice para la contestación de la segunda parte. En cada parte, la cantidad MÁXIMA de papel (de examen, timbrado) que puede emplear ESTÁ LIMITADA al equivalente a DOS (2) HOJAS de tamaño A4 (210 x 297 mm)

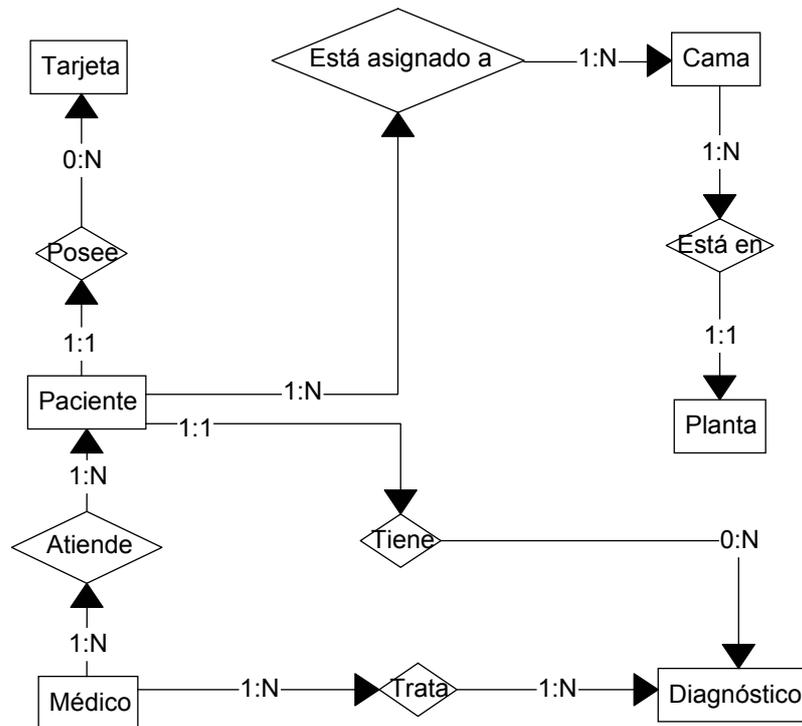
PRIMERA PARTE. PREGUNTAS TEÓRICAS (2'5 PUNTOS CADA UNA)

1. Explique la diferencia entre las pruebas de **caja negra** y las pruebas de **caja transparente**.  
¿Alguna de ellas garantiza la ausencia de fallos?
2. Dé una breve definición de Defecto, Fallo y Error. Explique la diferencia entre la estrategia de prevención de errores frente a la estrategia de recuperación de errores.

SEGUNDA PARTE. PREGUNTA DE APLICACIÓN (5 PUNTOS)

3. En un centro hospitalario se desea informatizar parte de la gestión relativa a sus pacientes. El sistema a construir deberá contemplar las siguientes cuestiones:
  - Un paciente estará asignado a una cama determinada de una planta del hospital, pudiendo estar a lo largo del tiempo de ingreso en diferentes camas y plantas.
  - Para cada paciente se entregarán hasta un máximo de 4 tarjetas de visita. Estas tarjetas servirán para que familiares y amigos del paciente le visiten durante su convalecencia.
  - A un paciente le pueden atender diferentes médicos.
  - Un paciente puede tener distintos diagnósticos de enfermedad.
  - Un médico puede tratar diferentes diagnósticos y viceversa.

*Analice el sistema mediante la notación DER (Diagrama Entidad Relación).*



**Observación:** El anterior diagrama se ha elaborado con la herramienta DOME. La siguiente tabla resume la equivalencia entre la notación de dicha herramienta y la propuesta en el libro de la asignatura.

Notación libro	Notación DOME
⊆	0:1
	1:1
⊆	0:N
⊆	1:N



UNIVERSIDAD NACIONAL DE  
EDUCACIÓN A DISTANCIA

INGENIERÍA TÉCNICA en INFORMÁTICA de SISTEMAS y de GESTIÓN  
Plan de estudios en extinción  
CÓDIGO CARRERA: 40=SISTEMAS y 41=GESTIÓN  
CÓDIGO DE ASIGNATURA: 210=SISTEMAS y 208=GESTIÓN  
ORIGINAL NACIONAL



Departamento de Lenguajes  
y Sistemas Informáticos

ASIGNATURA: INGENIERÍA DEL SOFTWARE (2º CURSO)

FECHA: 4 de septiembre de 2003 Hora: 11:30 Duración: 2 horas MATERIAL: NINGUNO

**Todas las preguntas de este ejercicio son eliminatorias en el sentido de que debe obtener una nota mínima en cada una de ellas. En la primera parte (las preguntas teóricas que se valoran con 2'5 puntos cada una) la nota mínima es 1 punto; en la segunda parte (ejercicio de teoría aplicada que se valora con 5 puntos) la nota mínima que debe obtener es de 2 puntos.**

**¡ATENCIÓN! PONGA SUS DATOS EN LA HOJA DE LECTURA ÓPTICA QUE DEBERÁ ENTREGAR JUNTO CON EL RESTO DE LAS RESPUESTAS.**

Conteste a las preguntas teóricas, en cualquier orden, en hojas diferentes a las que utilice para la contestación de la segunda parte. En cada parte, la cantidad MÁXIMA de papel (de examen, timbrado) que puede emplear ESTÁ LIMITADA al equivalente a DOS (2) HOJAS de tamaño A4 (210 x 297 mm)

PRIMERA PARTE. PREGUNTAS TEÓRICAS (2'5 PUNTOS CADA UNA)

1. Explique los inconvenientes que tiene el utilizar un ciclo de vida clásico para desarrollar determinado producto en el seno de una organización que carece de experiencia en el desarrollo de este tipo de productos y en las tecnologías implicadas.
2. Supóngase que, para describir el funcionamiento de una aplicación en el diseño, se utiliza un sistema de representación de la realidad con dos categorías: entidades y relaciones. Sitúe, en la categoría que le corresponda, cada uno de los elementos que se manejan en el diseño con abstracciones y en el diseño con objetos.

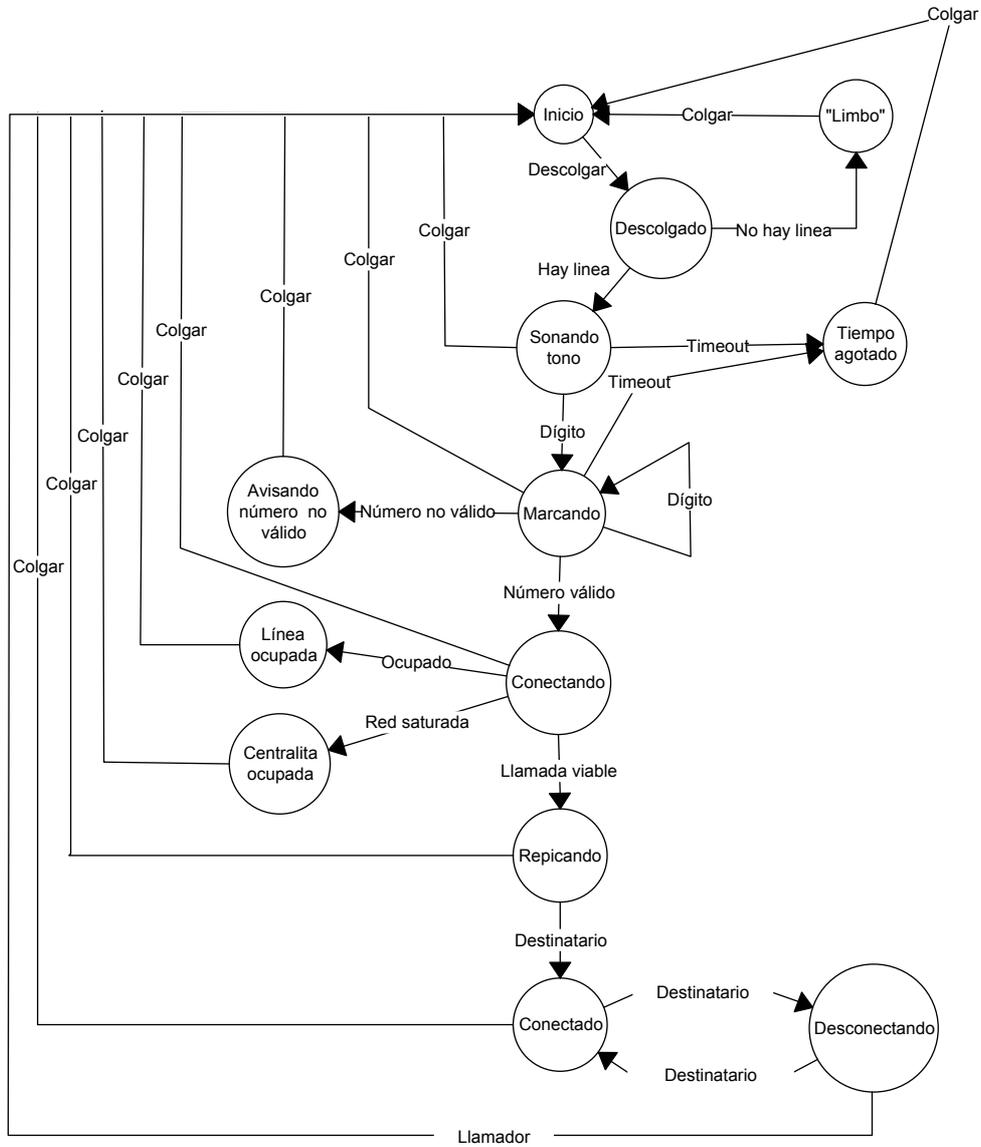
SEGUNDA PARTE. PREGUNTA DE TEORÍA APLICADA (MÁXIMO 5 PUNTOS)

3. Modele el establecimiento y la finalización de una llamada telefónica mediante un Diagrama de Transición de Estados. A continuación, se resume el funcionamiento que debe modelar:  
*Para establecer una llamada, en primer lugar, se debe descolgar el teléfono. En caso de que no se oiga una señal o tono, se deberá colgar el teléfono y repetir el intento. En caso contrario, se procede a marcar el número de teléfono del destinatario de la llamada. Si el número no es válido, habrá que colgar el teléfono y repetir el proceso desde el principio. Lo mismo ocurre si pasa demasiado tiempo hasta que se marca el número. Si no están ocupadas ni la línea ni la centralita, se establecerá la conexión en cuanto el destinatario descuelgue su teléfono. Por último, la llamada finalizará exclusivamente cuando la persona que inició la llamada cuelgue su teléfono.*



# Solución propuesta por el equipo docente

Autores: José Félix Estívariz López, Rubén Heradio Gil, Juan Antonio Mascarell Estruch





UNIVERSIDAD NACIONAL DE  
EDUCACIÓN A DISTANCIA

INGENIERÍA TÉCNICA en INFORMÁTICA de SISTEMAS y de GESTIÓN  
Plan de estudios en extinción  
CÓDIGO CARRERA: 40=SISTEMAS y 41=GESTIÓN  
CÓDIGO DE ASIGNATURA: 210=SISTEMAS y 208=GESTIÓN  
RESERVA NACIONAL



Departamento de Lenguajes  
y Sistemas Informáticos

ASIGNATURA: INGENIERÍA DEL SOFTWARE (2º CURSO)

FECHA: 8 de septiembre de 2003 Hora: 16:00 Duración: 2 horas MATERIAL: NINGUNO

**Todas las preguntas de este ejercicio son eliminatorias en el sentido de que debe obtener una nota mínima en cada una de ellas. En la primera parte (las preguntas teóricas que se valoran con 2'5 puntos cada una) la nota mínima es 1 punto; en la segunda parte (ejercicio de teoría aplicada que se valora con 5 puntos) la nota mínima que debe obtener es de 2 puntos.**

**¡ATENCIÓN! PONGA SUS DATOS EN LA HOJA DE LECTURA ÓPTICA QUE DEBERÁ ENTREGAR JUNTO CON EL RESTO DE LAS RESPUESTAS.**

Conteste a las preguntas teóricas, en cualquier orden, en hojas diferentes a las que utilice para la contestación de la segunda parte. En cada parte, la cantidad MÁXIMA de papel (de examen, timbrado) que puede emplear ESTÁ LIMITADA al equivalente a DOS (2) HOJAS de tamaño A4 (210 x 297 mm)

PRIMERA PARTE. PREGUNTAS TEÓRICAS (2'5 PUNTOS CADA UNA)

1. Explique la diferencia entre especificaciones funcionales y no funcionales en el análisis.
2. El objetivo del diseño es describir el funcionamiento de la aplicación de forma que: la codificación sea fácil y con economía de recursos; el producto funcione según especificaciones y, sobre todo, el mantenimiento sea sencillo. En este sentido, ¿qué inconvenientes acarrea el hacer un diseño de tipo monolítico?

SEGUNDA PARTE. PREGUNTA DE TEORÍA APLICADA (MÁXIMO 5 PUNTOS)

3. Se va a construir una estación meteorológica automática junto a un río. Esta estación medirá datos atmosféricos así como niveles de contaminación del río y los transmitirá, vía satélite, a la central de datos. Las especificaciones de funcionamiento son estas:  
La temperatura se mide a través de un termopar, estas medidas se realizan cada minuto. Cada 10 minutos se hace la media de las temperaturas leídas y se almacena su valor. Los datos de la presión se leen cada cuarto de hora y se calcula y guarda su media cada hora. También cada hora, se analizan 3 parámetros de nivel de contaminación de las aguas y se registran sus valores. Si algún parámetro pasa cierto umbral de peligro se genera una señal de alarma y se envía automáticamente a la central. Así mismo se mide el caudal del río cada 2 horas. Si se produce una crecida de forma brusca se envía una señal de alarma.  
Cada 2 horas la estación automática recopila sus datos los transmite a la central vía satélite. Para ello, previamente tiene que codificar dichos datos en un formato estándar de control de errores para realizar transmisiones tolerantes a fallos.

SE PIDE realizar un Diagrama de Flujos de Datos que modele el sistema anterior.

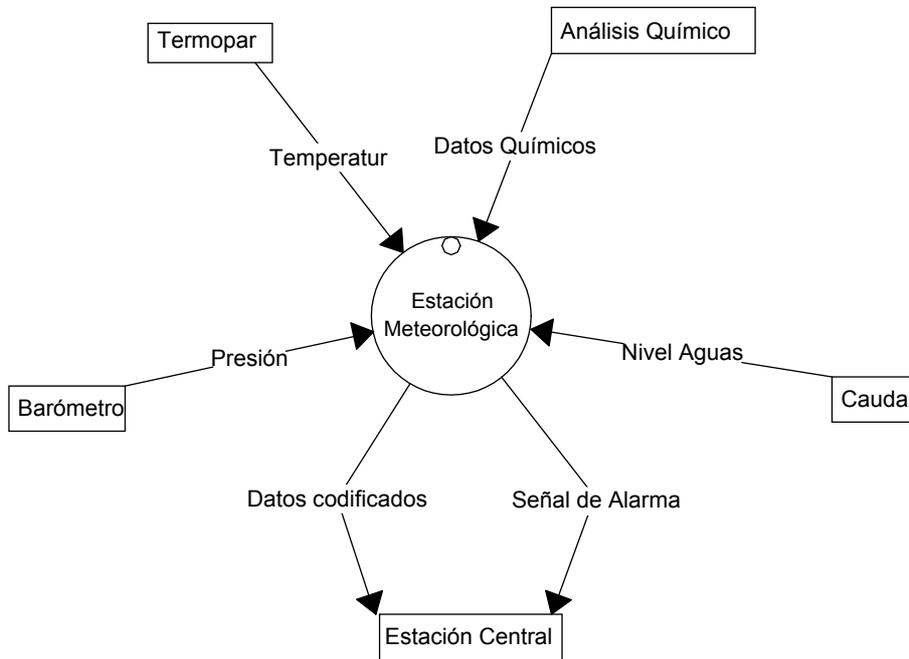


# Solución propuesta por el equipo docente

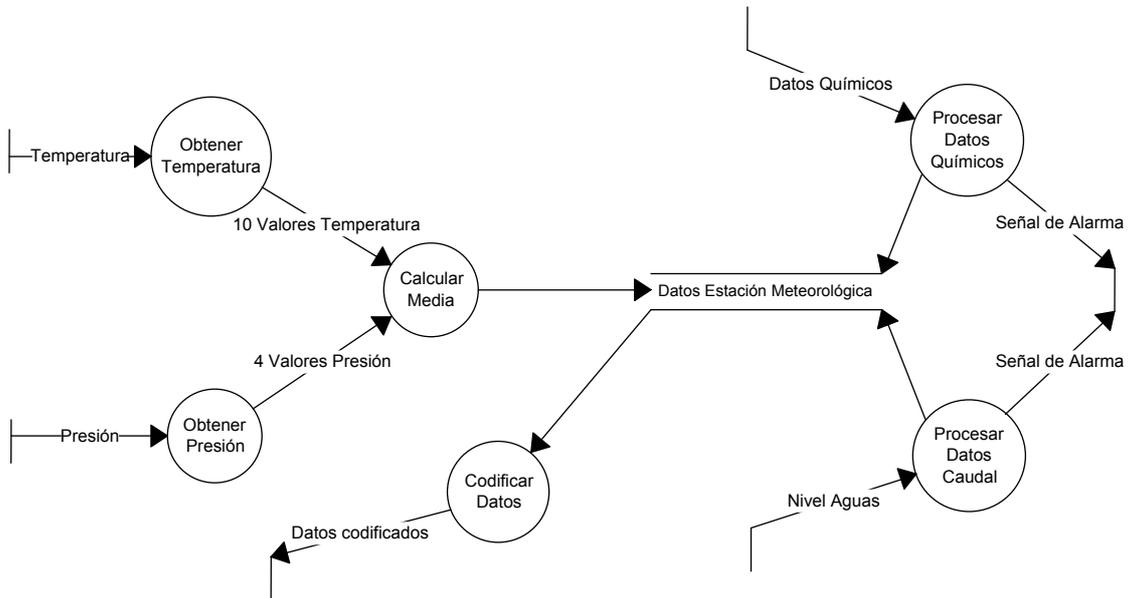
Autores: José Félix Estívariz López, Rubén Heradio Gil, Juan Antonio Mascarell Estruch



## Diagrama de Contexto:



## DFD 0:



*Observación: dada la simplicidad del DFD 0, se ha optado por no explotarlo en más niveles.*



UNIVERSIDAD NACIONAL DE  
EDUCACIÓN A DISTANCIA

INGENIERÍA TÉCNICA en INFORMÁTICA de SISTEMAS y de GESTIÓN  
Plan de estudios en extinción  
CÓDIGO CARRERA: 40=SISTEMAS y 41=GESTIÓN  
CÓDIGO DE ASIGNATURA: 210=SISTEMAS y 208=GESTIÓN  
EXTRANJERO ORIGINAL



Departamento de Lenguajes  
y Sistemas Informáticos

ASIGNATURA: INGENIERÍA DEL SOFTWARE (2º CURSO)  
FECHA: septiembre 2003 Hora: Duración: 2 horas MATERIAL: NINGUNO

---

**Todas las preguntas de este ejercicio son eliminatorias en el sentido de que debe obtener una nota mínima en cada una de ellas. En la primera parte (las preguntas teóricas que se valoran con 2'5 puntos cada una) la nota mínima es 1 punto; en la segunda parte (ejercicio de teoría aplicada que se valora con 5 puntos) la nota mínima que debe obtener es de 2 puntos.**

**¡ATENCIÓN! PONGA SUS DATOS EN LA HOJA DE LECTURA ÓPTICA QUE DEBERÁ ENTREGAR JUNTO CON EL RESTO DE LAS RESPUESTAS.**

Conteste a las preguntas teóricas, en cualquier orden, en hojas diferentes a las que utilice para la contestación de la segunda parte. En cada parte, la cantidad MÁXIMA de papel (de examen, timbrado) que puede emplear ESTÁ LIMITADA al equivalente a DOS (2) HOJAS de tamaño A4 (210 x 297 mm)

---

#### PRIMERA PARTE. PREGUNTAS TEÓRICAS (2'5 PUNTOS CADA UNA)

1. Suponga que se quiere modelar un sistema en el que lo más importante es representar las tareas y transformaciones que se hacen con la información y los demás aspectos son prácticamente irrelevantes. ¿Sería imprescindible el uso de Diagramas Entidad-Relación? ¿Qué aspecto del modelo prioriza esta notación?
2. ¿El diseño estructurado garantiza, por sí sólo, la independencia funcional? Indique cómo se mide la independencia funcional y explique en qué consisten estas métricas.

#### SEGUNDA PARTE. PREGUNTA DE TEORÍA APLICADA (MÁXIMO 5 PUNTOS)

3. Se nos pide que realicemos la aplicación informática de un cajero automático de un videoclub. El cliente nos da las siguientes especificaciones:  
*Para entrar al sistema el usuario necesita introducir su tarjeta personal. Lo primero que hace el sistema es comprobar el saldo de la tarjeta. Si tiene saldo cero o negativo solamente permite la acción de recargar tarjeta.*  
*A continuación aparece un menú con las tres únicas opciones: devolver una película, alquilar hasta un máximo de tres películas o recargar la tarjeta.*  
*Para alquilar una película la tarjeta tiene que estar actualizada y con saldo. En caso contrario, no permite alquilar. Si la tarjeta tiene saldo, el usuario puede seleccionar hasta un máximo de tres películas siempre y cuando estén disponibles. Para retirarlas debe proceder a confirmar los títulos elegidos.*  
*Al devolver la película el sistema calcula el importe y actualiza la tarjeta.*  
*Para recargar la tarjeta el usuario marca la cantidad deseada e introduce el dinero.*

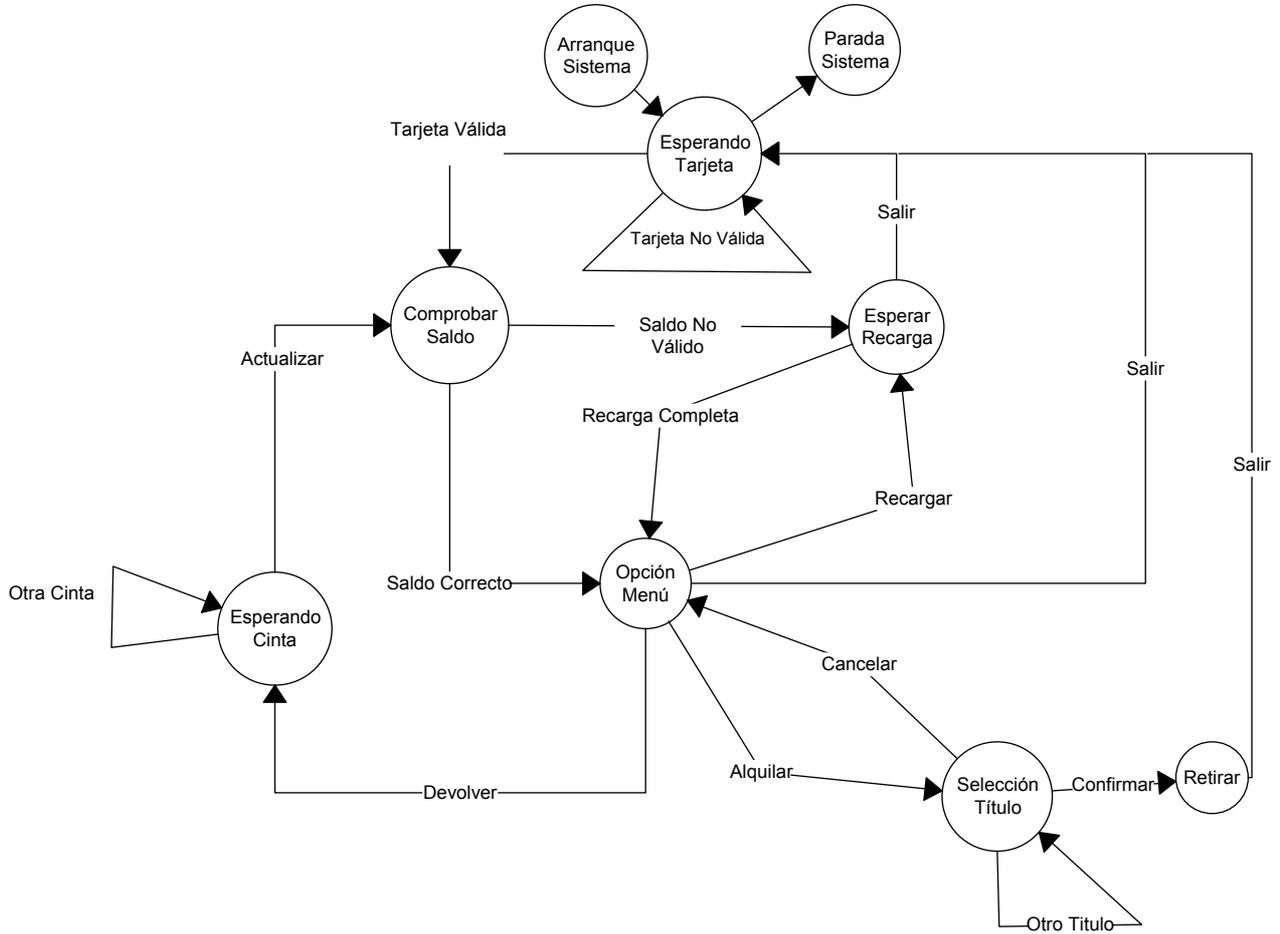
Realice un Diagrama de Transición de Estados con el cual se pueda comprobar, junto con el cliente, que hemos comprendido el funcionamiento de la aplicación.

Indique qué tipos de pruebas utilizaría para asegurar que el sistema funciona correctamente.



# Solución propuesta por el equipo docente

Autores: José Félix Estívariz López, Rubén Heradio Gil, Juan Antonio Mascarell Estruch





UNIVERSIDAD NACIONAL DE  
EDUCACIÓN A DISTANCIA

INGENIERÍA TÉCNICA en INFORMÁTICA de SISTEMAS y de GESTIÓN  
Plan de estudios en extinción  
CÓDIGO CARRERA: 40=SISTEMAS y 41=GESTIÓN  
CÓDIGO DE ASIGNATURA: 210=SISTEMAS y 208=GESTIÓN  
EXTRANJERO RESERVA



Departamento de Lenguajes  
y Sistemas Informáticos

ASIGNATURA: INGENIERÍA DEL SOFTWARE (2º CURSO)  
FECHA: septiembre 2003 Hora: Duración: 2 horas MATERIAL: NINGUNO

---

**Todas las preguntas de este ejercicio son eliminatorias en el sentido de que debe obtener una nota mínima en cada una de ellas. En la primera parte (las preguntas teóricas que se valoran con 2'5 puntos cada una) la nota mínima es 1 punto; en la segunda parte (ejercicio de teoría aplicada que se valora con 5 puntos) la nota mínima que debe obtener es de 2 puntos.**

**¡ATENCIÓN! PONGA SUS DATOS EN LA HOJA DE LECTURA ÓPTICA QUE DEBERÁ ENTREGAR JUNTO CON EL RESTO DE LAS RESPUESTAS.**

Conteste a las preguntas teóricas, en cualquier orden, en hojas diferentes a las que utilice para la contestación de la segunda parte. En cada parte, la cantidad MÁXIMA de papel (de examen, timbrado) que puede emplear ESTÁ LIMITADA al equivalente a DOS (2) HOJAS de tamaño A4 (210 x 297 mm)

---

PRIMERA PARTE. PREGUNTAS TEÓRICAS (2'5 PUNTOS CADA UNA)

1. Reflexione sobre la figura del analista. ¿Se limita a recoger las necesidades del cliente/usuario, investigar soluciones análogas y deducir unas conclusiones que traduce a un lenguaje comprensible por los informáticos y que representa mediante un modelo o tiene que tomar, en ocasiones, algún tipo de decisión? ¿Se le ocurre de qué tipo serían estas decisiones?
2. Explique brevemente en qué consisten y qué tipos hay de pruebas de caja transparente.

SEGUNDA PARTE. PREGUNTA DE TEORÍA APLICADA (MÁXIMO 5 PUNTOS)

3. Se desea construir un sistema informático que automatice la gestión de los empleados, departamentos y proyectos que se realizan en una empresa.

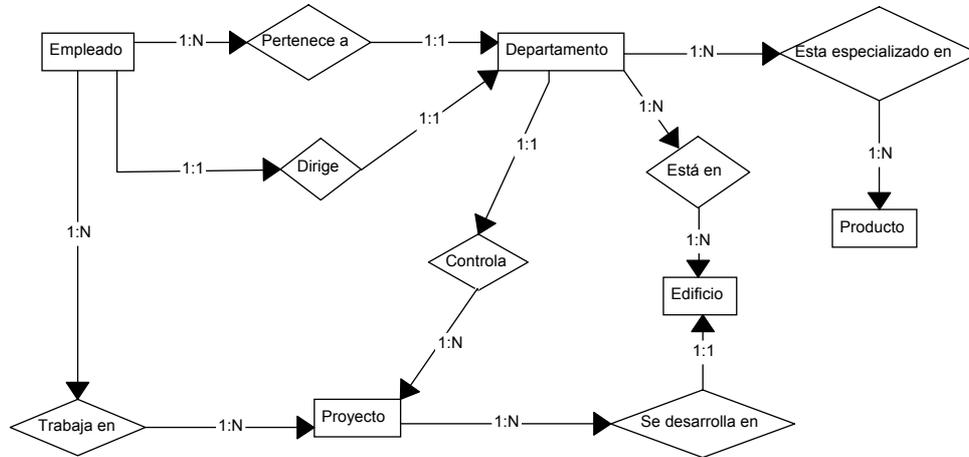
La empresa en cuestión, se organiza en departamentos. Cada departamento dispone de cierto número de empleados y de un director. Los departamentos se especializan en uno o varios productos, aunque puede darse la situación de que más de un departamento esté cualificado para construir un determinado producto. Por otro lado, cada departamento controla cierto número de proyectos. Un empleado está asignado a un sólo departamento, aunque puede trabajar en varios proyectos controlados por otros departamentos. Por último, la empresa dispone de varias sedes. Los departamentos pueden estar repartidos en distintos edificios. Sin embargo, los proyectos se desarrollan exclusivamente en una sede.

Analice el sistema propuesto utilizando un diagrama Entidad-Relación.



# Solución propuesta por el equipo docente

Autores: José Félix Estívariz López, Rubén Heradio Gil, Juan Antonio Mascarell Estruch



**Observación:** El anterior diagrama se ha elaborado con la herramienta DOME. La siguiente tabla resume la equivalencia entre la notación de dicha herramienta y la propuesta en el libro de la asignatura.

Notación libro	Notación DOME
d	0:1
	1:1
o	0:N
K	1:N



UNIVERSIDAD NACIONAL DE  
EDUCACIÓN A DISTANCIA

INGENIERÍA TÉCNICA en INFORMÁTICA de SISTEMAS y de GESTIÓN

Plan de estudios en extinción

CÓDIGO CARRERA: 40=SISTEMAS y 41=GESTIÓN

CÓDIGO DE ASIGNATURA: 210=SISTEMAS y 208=GESTIÓN

**EXAMEN EXTRAORDINARIO DE DICIEMBRE**



Departamento de Lenguajes  
y Sistemas Informáticos

ASIGNATURA: INGENIERÍA DEL SOFTWARE (2º CURSO)

FECHA: 10 de diciembre de 2003 Hora: 11:30 Duración: 2 horas MATERIAL: NINGUNO

---

**Todas las preguntas de este ejercicio son eliminatorias en el sentido de que debe obtener una nota mínima en cada una de ellas. En la primera parte (las preguntas teóricas que se valoran con 2'5 puntos cada una) la nota mínima es 1 punto; en la segunda parte (ejercicio de teoría aplicada que se valora con 5 puntos) la nota mínima que debe obtener es de 2 puntos.**

**¡ATENCIÓN! PONGA SUS DATOS EN LA HOJA DE LECTURA ÓPTICA QUE DEBERÁ ENTREGAR JUNTO CON EL RESTO DE LAS RESPUESTAS.**

Conteste a las preguntas teóricas, en cualquier orden, en hojas diferentes a las que utilice para la contestación de la segunda parte. En cada parte, la cantidad MÁXIMA de papel (de examen, timbrado) que puede emplear ESTÁ LIMITADA al equivalente a DOS (2) HOJAS de tamaño A4 (210 x 297 mm)

---

#### **PRIMERA PARTE. PREGUNTAS TEÓRICAS (2'5 PUNTOS CADA UNA)**

1. Defina los siguientes tipos de prueba de software: caja negra, caja transparente, alfa y beta. ¿Las pruebas alfa y beta son de caja negra o transparente?
2. Explique en que consisten las fases de análisis y diseño de un producto software. ¿Cuáles serían las diferencias entre un Diagrama de Transición de Estados utilizado en el análisis respecto a otro usado en el diseño?

#### **SEGUNDA PARTE. PREGUNTA DE APLICACIÓN (5 PUNTOS)**

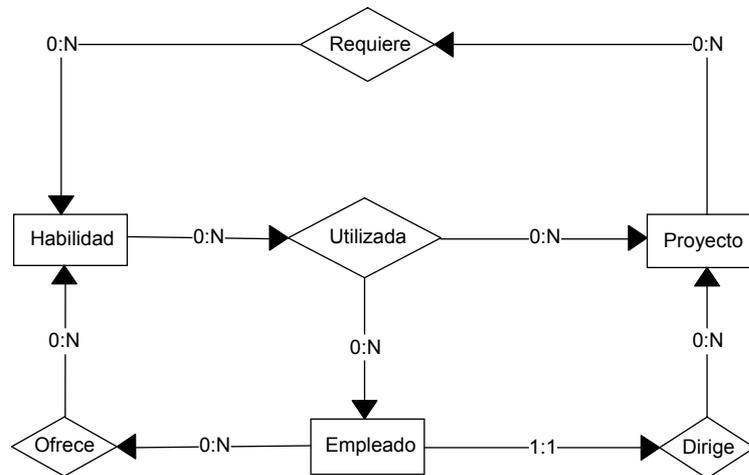
3. Modele mediante un Diagrama Entidad-Relación (DER) la información que se enuncia a continuación sobre los proyectos que se desarrollan en una empresa:

*Un empleado puede dirigir varios proyectos, aunque cada proyecto ha de ser dirigido por un solo empleado. La asignación entre proyectos y directores se realiza según las habilidades necesarias para el desarrollo de cada proyecto y las que ofrece cada empleado. El DER deberá reflejar las habilidades que un empleado utiliza en un proyecto concreto.*



## Solución propuesta por el equipo docente

Autores: José Félix Estívariz López, Rubén Heradio Gil, Juan Antonio Mascarell Estruch



**Observación:** El anterior diagrama se ha elaborado con la herramienta DOME. La siguiente tabla resume la equivalencia entre la notación de dicha herramienta y la propuesta en el libro de la asignatura.

Notación libro	Notación DOME
0:1	0:1
1:1	1:1
0:N	0:N
1:N	1:N