

UT01. DESARROLLO DEL SOFTWARE

Entornos de Desarrollo

1 DAW – C.I.F.P. Carlos III - Cartagena

4. Fases del desarrollo de una aplicación

1. Análisis

- **Previo:** seleccionar el modelo de ciclo de vida a seguir
- **Objetivo:** especificación de requisitos → entender, comprender y dar solución al problema.
- **Dificultades:**
 - El cliente puede no tenerlos claros
 - Pueden surgir nuevos requisitos
 - Pueden cambiar los ya especificados
 - Puede existir malos entendidos por falta de conocimiento del equipo de desarrollo sobre el problema
 - El cliente puede no expresarse bien por falta de conocimientos informáticos
- Buena comunicación cliente-desarrolladores

1. Análisis

- Técnicas de recogida de información
 - **Entrevistas.** Conocimientos en psicología
 - **Desarrollo Conjunto de Aplicaciones-JAD** (*Joint Application Development*): dinámica de grupos, entrevista muy estructurada aplicable a grupos de personas (usuarios, administradores, analistas, desarrolladores, ...)
 - **Planificación conjunta de requisitos-JRP** (Joint Requirements Planning): subconjunto de JAD dirigidas a la alta dirección. Requisitos de alto nivel o estratégicos
 - **Brainstorming:** grupales, generación de ideas para la resolución de problemas. Uso al principio del proyecto
 - **Prototipos:** versión inicial del sistema, para clarificar algunos puntos.
 - **Casos de uso:** en UML (Unified Modeling Language). Escenarios de uso del software en una situación

1. Análisis

- Tipos de requisitos
 - **Funcionales.** Describen la función del sistema, cómo reacciona, cómo se comporta, ...
 - **No funcionales:** características del sistema (fiabilidad, mantenibilidad, sistema operativo, restricciones,






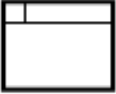






Requisitos funcionales	Requisitos no funcionales
El usuario puede agregar un nuevo contacto	La aplicación debe funcionar en sistemas operativos Linux y Windows
El usuario puede ver una lista con todos los contactos	El tiempo de respuesta a consultas, altas, bajas y modificaciones ha de ser inferior a 5 segundos
A partir de la lista de contactos el usuario puede acceder a un contacto	Utilizar un sistema gestor de base de datos para almacenar los datos
El usuario puede eliminar un contacto o varios de la lista	Utilizar un lenguaje multiplataforma para el desarrollo de la aplicación
El usuario puede modificar los datos de un contacto seleccionado de la lista	La interfaz de usuario es a través de ventanas, debe ser intuitiva y fácil de manejar
El usuario puede seleccionar determinados contactos	El manejo de la aplicación se realizará con el teclado y el ratón
El usuario puede imprimir la lista de contactos	Espacio libre en disco, mínimo: 1GB. Mínima cantidad de memoria 2GB

1. Análisis

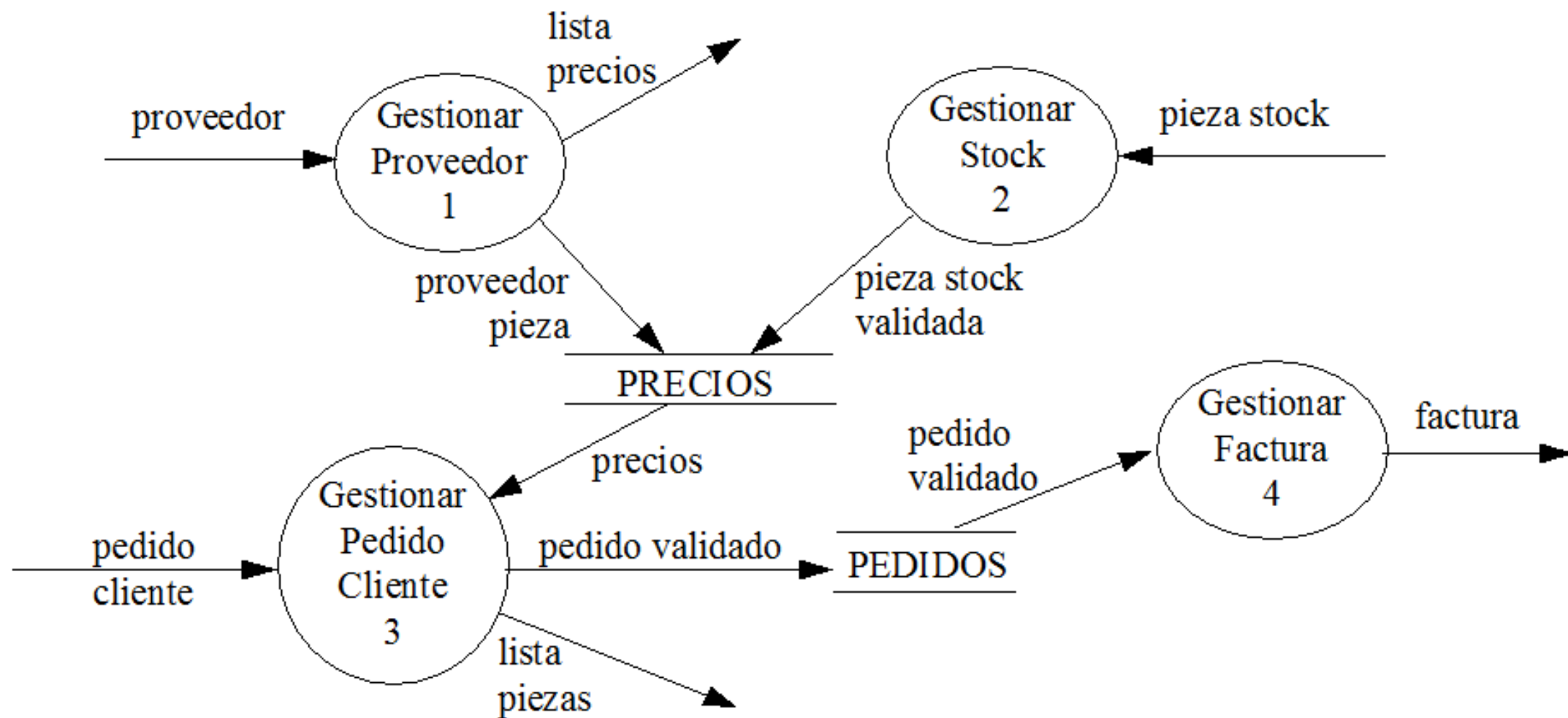
- Técnicas de representación

- **DFD**-Diagramas de Flujo de Datos

- Representa el cómo viajan los datos entre los procesos, entidades externas y almacenes en diferentes niveles de abstracción
 - **Procesos**=funciones del sistema
 - **Entidades externas**=componentes que no forman parte del sistema, pero proporcionan o reciben datos del sistema
 - **Almacén**=lugar dónde se almacenan los datos

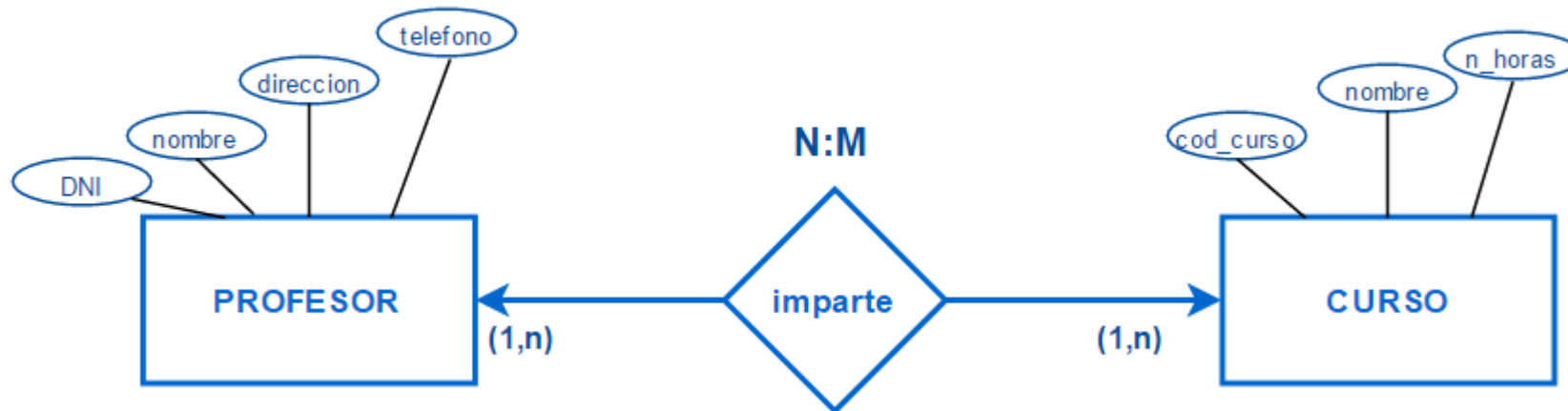
	YORDON/ DEMARCO	GALE & SARSON	SSADM METRICA
Flujo de datos			
Proceso			
Almacén de datos			
Entidad externa			

Ejemplo DFD



1. Análisis

- **Diagramas de control de flujos:** similar a DFD que muestras el flujo de control, en lugar de datos
- **Diagramas de transición de estados:** cómo se comporta el sistema como consecuencia de sucesos externos
- **Diagrama Entidad/Relación:** para representar los datos y las relaciones entre ellos.



1. Análisis

- ERS-Especificación de Requisitos del Software
 - No ambigüedades
 - Completo
 - Consistente
 - Fácil de verificar
 - Fácil de modificar
 - Fácil de usar en Explotación y mantenimiento
 - Fácil de identificar el origen y las consecuencias de los requisitos
- IEEE, 830

1. Introducción.

- 1.1 Propósito.
- 1.2 Ámbito del Sistema.
- 1.3 Definiciones, Acrónimos y Abreviaturas.
- 1.4 Referencias.
- 1.5 Visión general del documento.

2. Descripción General.

- 2.1 Perspectiva del Producto.
- 2.2 Funciones del Producto.
- 2.3 Características de los usuarios.
- 2.4 Restricciones.
- 2.5 Suposiciones y Dependencias.
- 2.6 Requisitos Futuros.

3. Requisitos Específicos.

- 3.1 Interfaces Externas.
- 3.2 Funciones.
- 3.3 Requisitos de Rendimiento.
- 3.4 Restricciones de Diseño.
- 3.5 Atributos del Sistema.
- 3.6 Otros Requisitos.

4 Apéndices.

<http://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>

2. Diseño

- Forma de solucionar el problema una vez identificados los requisitos
- Diseño **estructurado** y diseño **orientado a objetos**
- Diseño **estructurado**
 - **Diseño de datos:** transformación en basado en los datos y sus relaciones y su descripción detallada
 - **Diseño arquitectónico:** representación de la estructura de los componentes del software, propiedades e interacciones. Módulo, base de datos.
 - **Diseño de la interfaz:** cómo se comunica el software. Formatos de pantalla
 - **Diseño a nivel de componentes** (diseño procedimental). Resultado de esta tarea es el diseño de cada componente software con el suficiente con el suficiente nivel de detalle para que pueda servir de guía en la generación de código fuente en un lenguaje de programación.

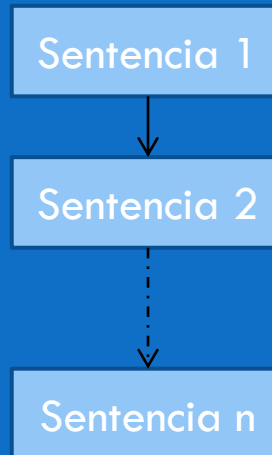
2. Diseño

- Programación estructurada.
 - Secuenciales.
 - Alternativas.
 - Cíclicas o repetitivas.
- Notaciones
 - Diagramas de flujo: ordinogramas-organigramas
 - Pseudocódigo
 - Diagramas de cajas
 - Tablas de decisión

2. Diseño

- Secuenciales

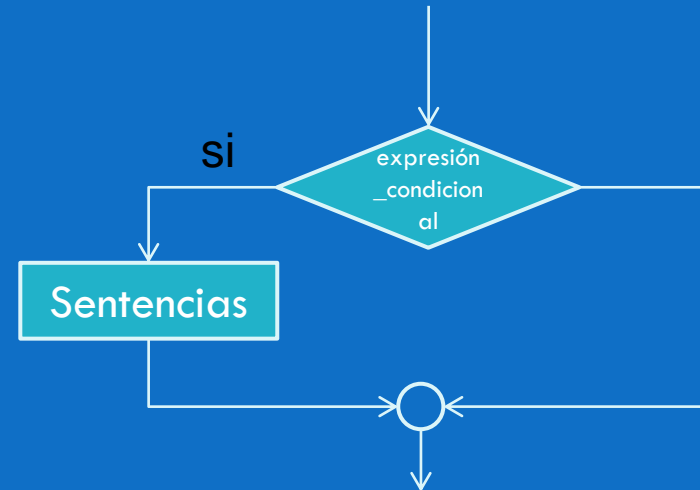
Sentencia 1;
Sentencia 2;
.....;
Sentencia n ;



2. Diseño

- Alternativa
 - Simple

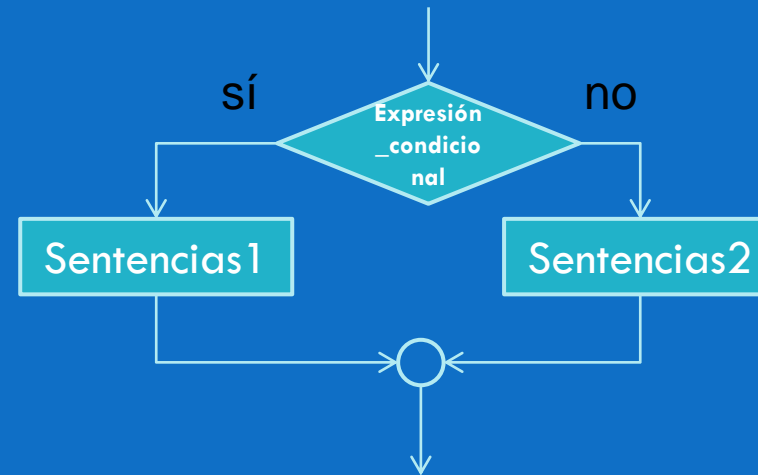
```
si <expresión_condicional> entonces  
    sentencias;  
finSi;
```



2. Diseño

- Alterantiva
 - Doble

```
si <expresión_condicional> entonces  
    sentencias1;  
si no  
    sentencias2;  
finSi;
```



2. Diseño

- Alternativa
 - Múltiple

```
Según <variable_numérica> hacer  
  <opción1>:  
    <setencias1>;  
  <opción2>:  
    <setencias2>;  
  .....  
  de otro modo:  
    <setenciasn>;  
finSegun
```



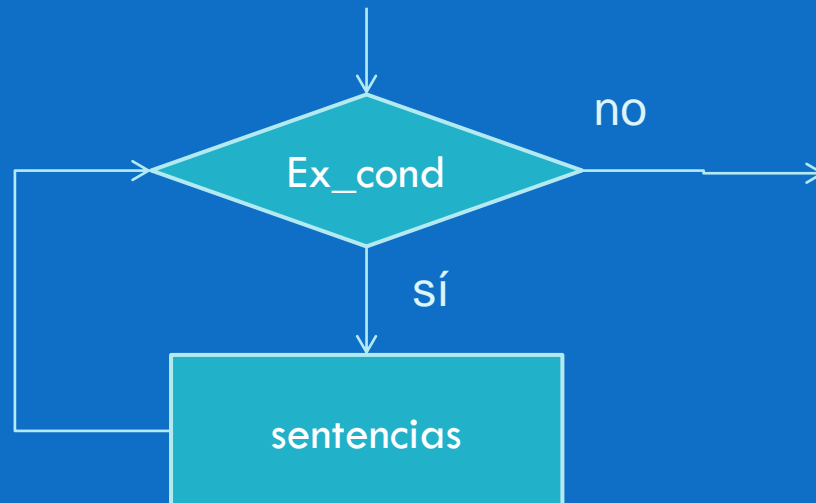
2. Diseño

- Alternativa.
 - Anidamiento de estructuras alternativas.
 - Cada elemento “si no” en una estructura alternativa se corresponde con la estructura “si” inmediatamente superior.
 - Supone tener completamente una estructura dentro de otra.

2. Diseño

- Repetitivas o cíclicas.
 - Mientras

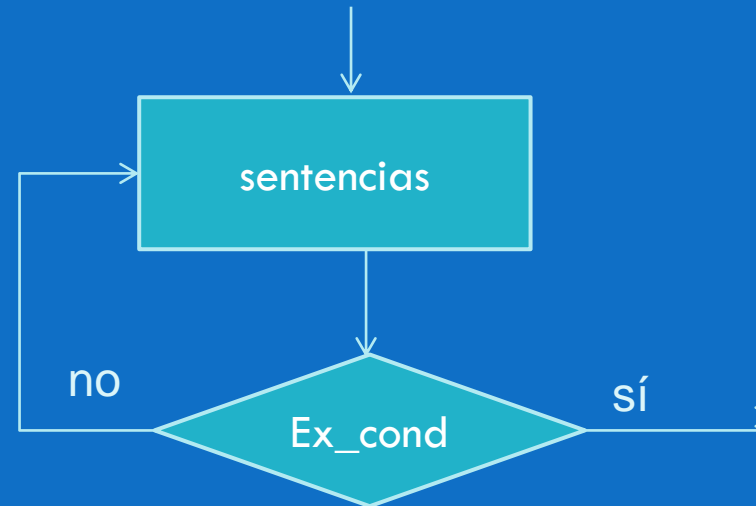
Mientras <expresión-condicional> hacer
 <sentencias>
FinMientras;



2.Diseño

- Repetitivas o cíclicas.
 - Hasta.

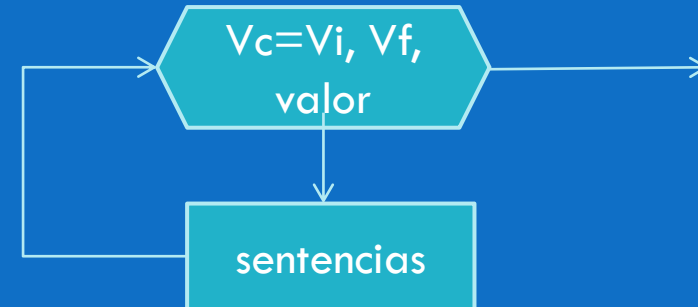
Repetir
 <sentencias>
Hasta <expresión-condicional>;



2. Diseño

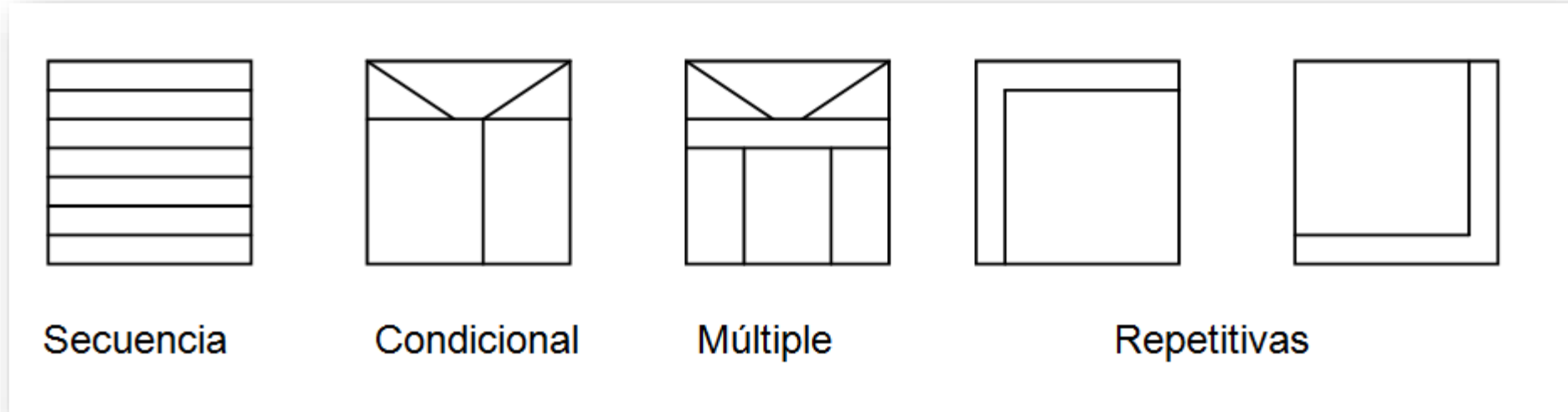
- Repetitivas o cíclicas
 - Para

Para $\langle V_c \rangle = \langle V_i \rangle$ hasta $\langle V_f \rangle$ incremento $\langle \text{valor} \rangle$
 $\langle \text{sentencias} \rangle$
FinPara;

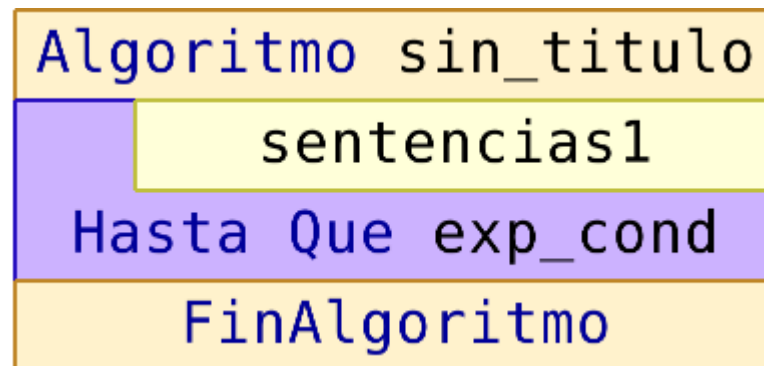
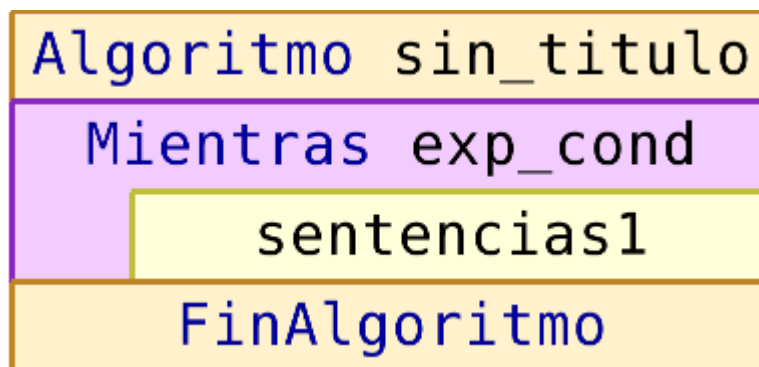
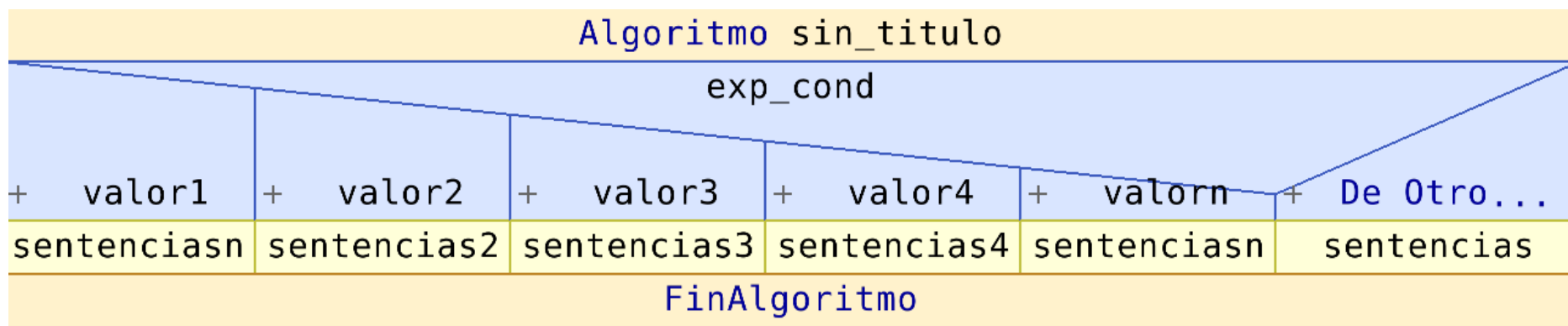
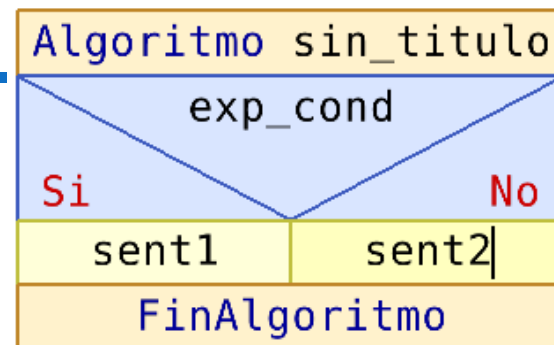
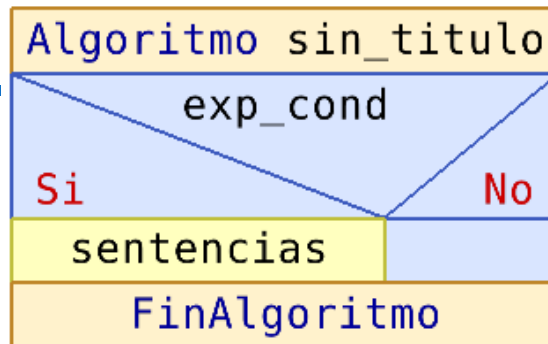
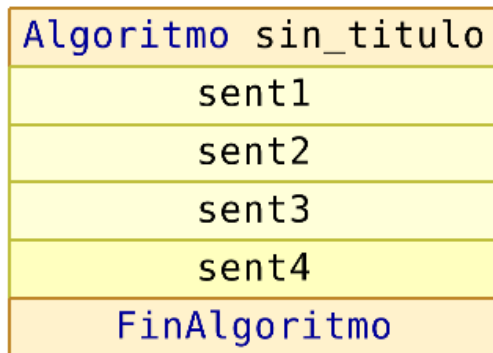


2. Diseño

- Diagramas de cajas, Nassi-Schneiderman o de Chapin



- Ventajas: la representación gráfica
- Inconvenientes: dificultad a la hora de modificar el diagrama



2. Diseño

- Tablas de decisión
 - **Uso:** evaluación de una compleja combinación de condiciones y selección de las acciones a realizar.

Condiciones que intervienen en el problema	Condiciones	1	2	3	4				n
	Condición 1	SI	SI	SI	SI				NO
	Condición 2	SI	SI	NO	NO				NO
	Condición 3	SI	NO	SI	NO				NO
Acciones a realizar en un determinado estado de las condiciones	Acciones								
	Acción 1	X			X				
	Acción 2	X	X						
	Acción 3		X	X	X				
	Acción 4			X	X				

Matriz de tantas filas como condiciones y columnas como situaciones distintas tenga

Mismas columnas y número de filas = acciones

Situación

Tratamiento

Regla de
decisión

2. Diseño. Tabla de decisión

- Situaciones.
 - **Simple.** Todas las condiciones han de ser evaluadas
 - **Compuestas.** Una o más condiciones no necesitan ser evaluada. Se tiene una situación de indiferencia (-)

Situación Simple					Situación Compuesta		
Condición 1	SI	SI	SI	NO	---	NO	NO
Condición 2	SI	NO	NO	SI	SI	NO	NO
Condición 3	SI	SI	NO	SI	NO	SI	NO
Acción 1	X		X	X			
Acción 2			X		X	X	X
Acción 3		X		X			X

Equivale a:

SI	NO
SI	SI
NO	NO

2. Diseño

- Clasificación de las tablas de decisión
 - TD binarias
 - TD múltiples
 - TD mixtas

2. Diseño

- **TD Binarias:** condiciones dos valores (si/no, blanco/negro, ...)
- **Ejemplo.** Aplicación para representar gráficos con un pc por pantalla o impresora, dependiendo de las características del equipo (monitor monocromo (M) o color (C), tarjeta gráfica adecuada (S/N) e impresora adecuada (S/N))

Tipo de monitor	M	M	M	M	C	C	C	C
Tarjeta gráfica	S	S	N	N	S	S	N	N
Impresora	S	N	S	N	S	N	S	N
Por pantalla	X	X			X	X		
Impresora	X		X		X			
Equipo insuficiente				X				
Contradicción							X	X

- TD múltiples. Condiciones con más de dos valores.

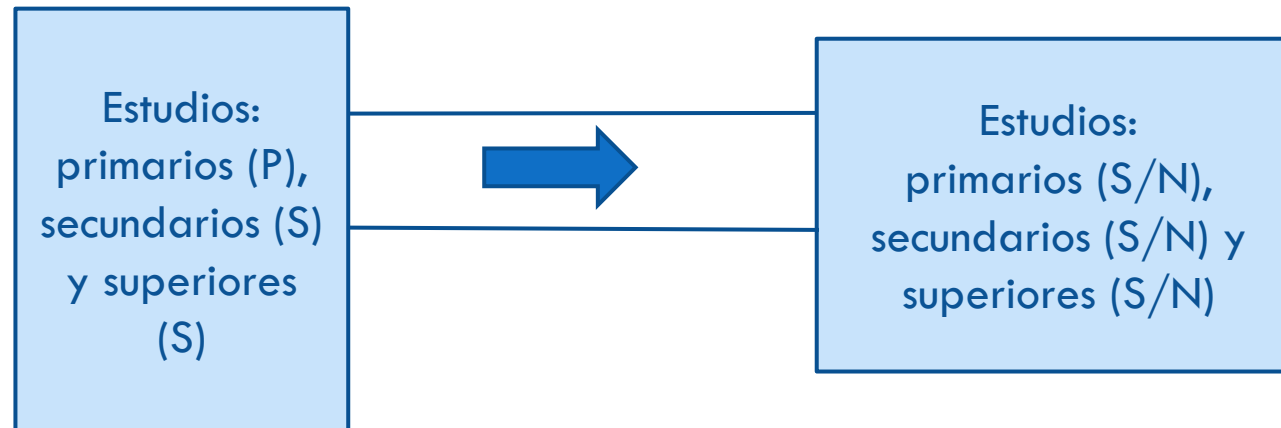
- **Ejemplo:** una empresa, en la confección de la nómina, incluye un plus sobre el sueldo de sus empleados, dependiendo del número de hijos (sin hijos (S), con hijos (C), numerosa general (NG) y numerosa especial (NE)) y de su nivel de estudios (primarios (P), medios (M) y superiores (S)).

[illegible]

[illegible]

2. Diseño


- Resolución de una tabla de decisión
 - Se trabaja con binarias y reglas AND.
 - **Primer paso** → transformar reglas OR en AND (---, indiferencias) y múltiples en binarias



2. Diseño

- Segundo paso → eliminar redundancias.
- Tercer paso → completitud. Comprobar que tiene todas las situaciones posibles.
- Cuarto paso → simplificación. Mantener el mayor número de indiferencias posible.
 - Agrupar dos reglas con el mismo tratamiento y mismas condiciones excepto una y sustituirlas por una nueva regla con el mismo tratamiento y mismas condiciones con una indiferencia en la condición diferente.

C1	...	SI	SI	...
C2	...	NO	NO	...
C3	...	NO	SI	...
A1				
A2	...	X	X	...
A3				



C1	...	SI	...
C2	...	NO	...
C3	...	---	...
A1			
A2	...	X	...
A3			

2. Diseño

C1	...	SI	NO	...
C2	...	---	---	...
C3	...	SI	SI	...
A1		X	X	
A2
A3		X	X	



C1	...	---	...
C2	...	---	...
C3	...	SI	...
A1		X	
A2
A3		X	

- **Regla del paraguas.** Establecimiento de ciclos cerrados uniendo pares de reglas que cumplen las condiciones de la simplificación.
 - 2 simples = 1 compuesta con 1 indiferencia
 - 4 simples = 1 compuesta con 2 indiferencias
 - 8 simples = 1 compuesta con 3 indiferencias
 - 2^n simples = 1 compuesta con n indiferencias

2. Diseño

C1	S	S	S	S	S	S	S	S	N	N	N	N	N	N	N	N
C2	S	S	S	S	N	N	N	N	S	S	S	S	N	N	N	N
C3	S	S	N	N	S	S	N	N	S	S	N	N	S	S	N	N
C4	S	N	S	N	S	N	S	N	S	N	S	N	S	N	S	N
A1			X						X	X	X	X	X			
A2	X	X		X	X	X	X	X					X	X		
A3				X					X	X	X	X		X	X	X



2. Diseño

C1	S	S	S	S	S	N	N	N	N
C2	---	S	S	S	N	S	N	N	N
C3	S	S	N	N	N	---	S	S	N
C4	---	N	S	N	---	---	S	N	---
A1			X			X	X		
A2	X	X		X	X		X	X	
A3				X		X		X	X

- **Ordenación por importancias.** Ordenación de las reglas más importantes hacia la izquierda y las condiciones más importantes .
 - **Importancia de una regla.** Número de reglas a las que equivale.
 - **Importancia de una condición.** Suma de las importancias de las reglas en las que la condición es evaluada.

2. Diseño

Importancia de reglas								
	4	1	1	2	4	1	1	2
C1	---	S	N	---	---	S	N	---
C2	---	S	S	N	S	N	N	N
C3	S	S	S	S	N	N	N	N
C4	S	N	N	N	---	S	S	N
A1		X			X	X		
A2	X		X	X		X	X	
A3	X			X	X			X

4
12
16
12

Importancia de condiciones

12
4
0
4

D.C.

D.C. Dash Count de una condición. Es la suma de las importancias de las reglas en las que la condición **no** es evaluada (indiferencia)

2. Diseño

	4	4	2	2	1	1	1	1
C1	---	---	---	---	S	N	S	N
C2	---	S	N	N	S	S	N	N
C3	S	N	S	N	S	S	N	N
C4	S	---	N	N	N	N	S	N
A1		X			X		X	
A2	X		X			X	X	X
A3	X	X	X	X				

Ordenación por reglas

Ordenación por condiciones. Si hay dos valores iguales → se calcula un segundo coeficiente δ , como el valor absoluto de la diferencia entre la cantidad de valores S y valores N

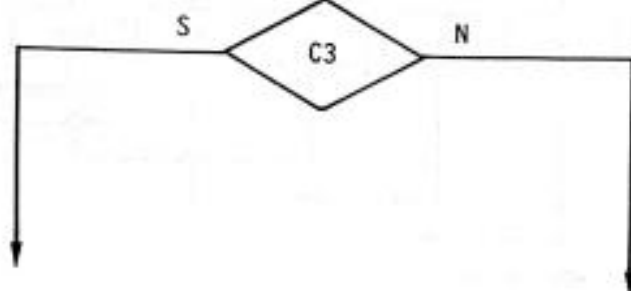
									I.C.	δ
C3	S	N	S	N	S	S	N	N	16	0
C2	---	S	N	N	S	S	N	N	12	1
C4	S	---	N	N	N	N	S	N	12	3
C1	---	---	---	---	S	N	S	N	4	0
A1		X			X		X			
A2	X		X			X	X	X		
A3	X	X	X	X						

2. Diseño

- Paso a ordinograma
 - Se inicia con la primera condición
 - Se divide la tabla en dos, con los S de la condición evaluada y los N.
 - Con cada tablas resultante se aplica el mismo proceso, recalculando las importancias.

I.R.	4	4	2	2	1	1	1	1	I.C.
C3	S	N	S	N	S	S	N	N	16
C2	-	S	N	N	S	S	N	N	12
C4	S	-	N	N	N	N	S	S	12
C1	-	-	-	-	S	N	S	N	4

A1		X			X		X	
A2	X		X			X	X	X
A3	X	X	X	X				



I.R.	4	2	1	1	I.C.
C2	-	N	S	S	4
C4	S	N	N	N	8
C1	-	-	S	N	2

A1			X	
A2	X	X		X
A3	X	X		

1

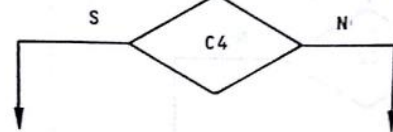
I.R.	4	2	1	1	I.C.
C2	S	N	N	N	8
C4	-	N	S	S	4
C1	-	-	S	N	2

A1	X		X	
A2	X		X	X
A3	X	X		

2

1

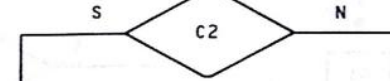
I.R.	4	2	1	1	I.C.
C4	S	N	N	N	8
C2	-	N	S	S	4
C1	-	-	S	N	2
A1			X		
A2	X	X		X	
A3	X	X			



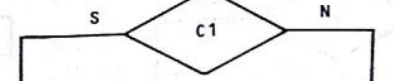
C2	-
C1	-
A1	X
A2	X
A3	X

A2
A3

I.R.	2	1	1	I.C.
C2	N	S	S	4
C1	-	S	N	2
A1		X		
A2	X		X	
A3	X			



C1	S	N
A1	X	
A2		X
A3		



A1

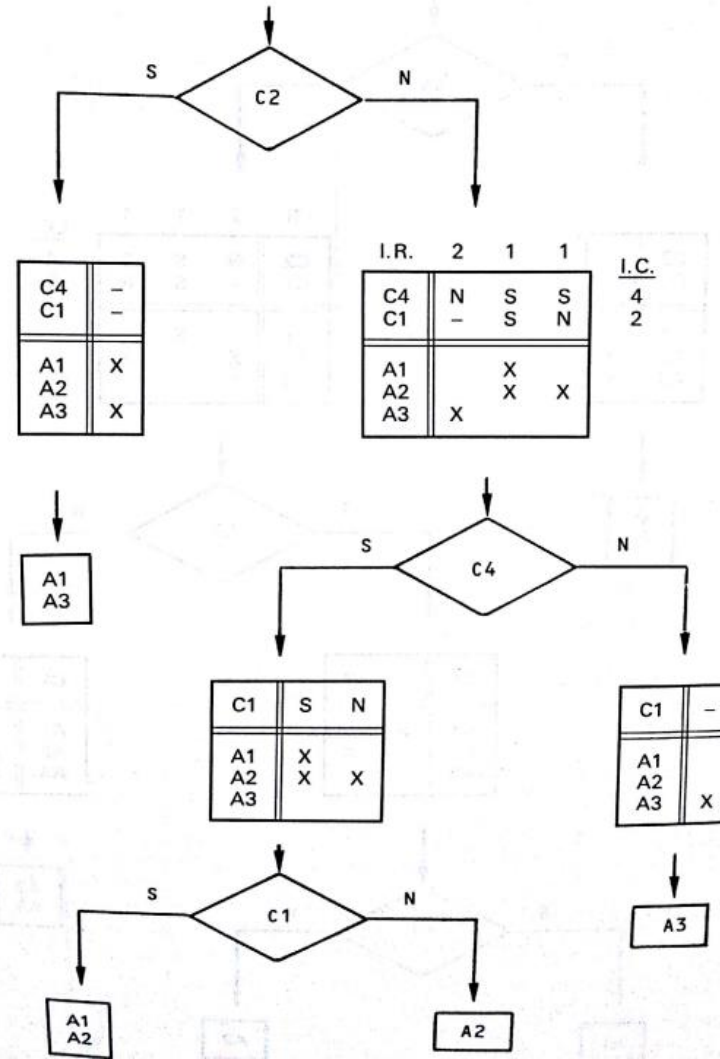
A2

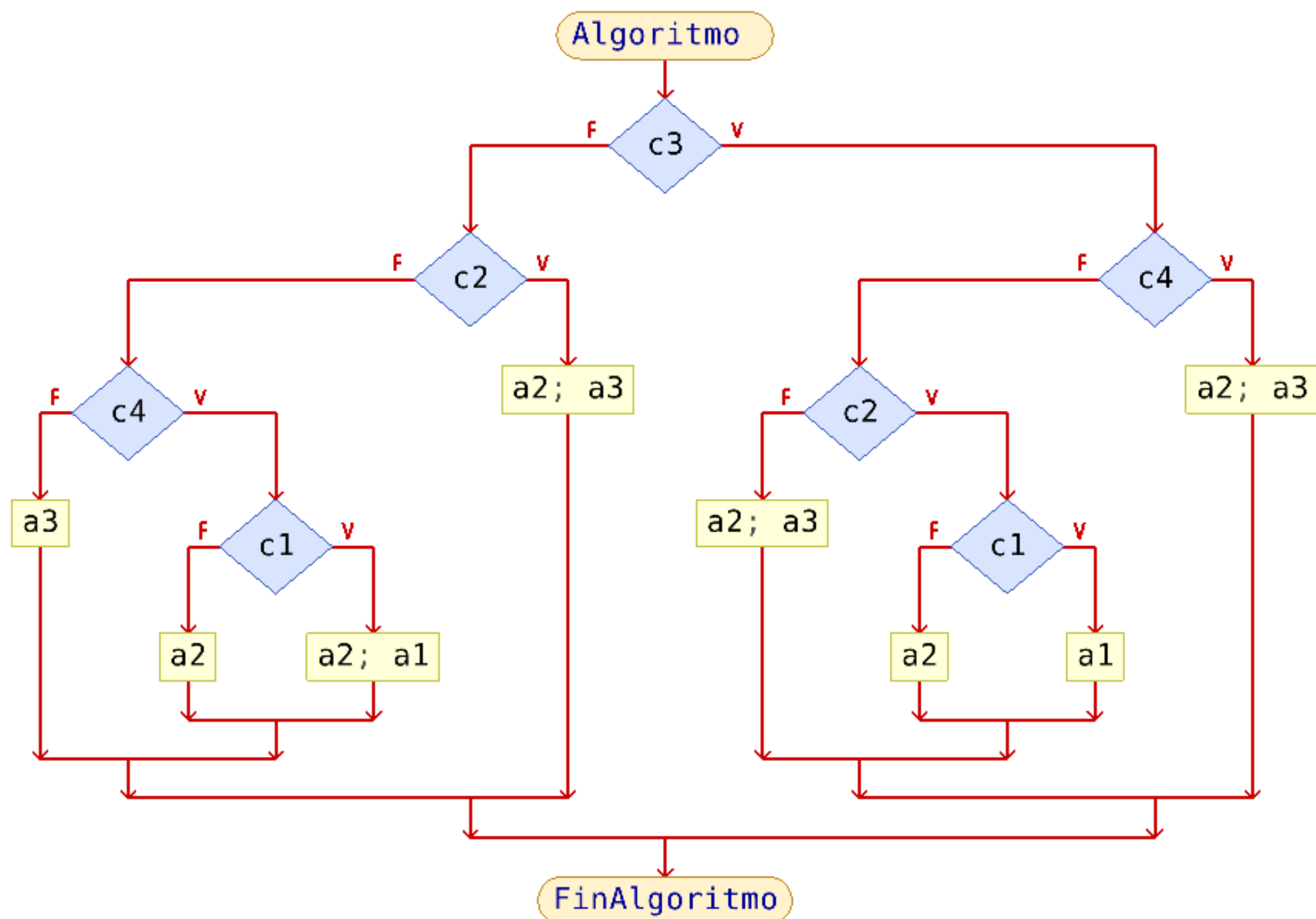
C1	-
A1	X
A2	X
A3	X

A2
A3

2

I.R.	4	2	1	1	I.C.
C2	S	N	N	N	8
C4	-	N	S	S	4
C1	-	-	S	N	2
A1	X		X		
A2			X	X	
A3	X	X			

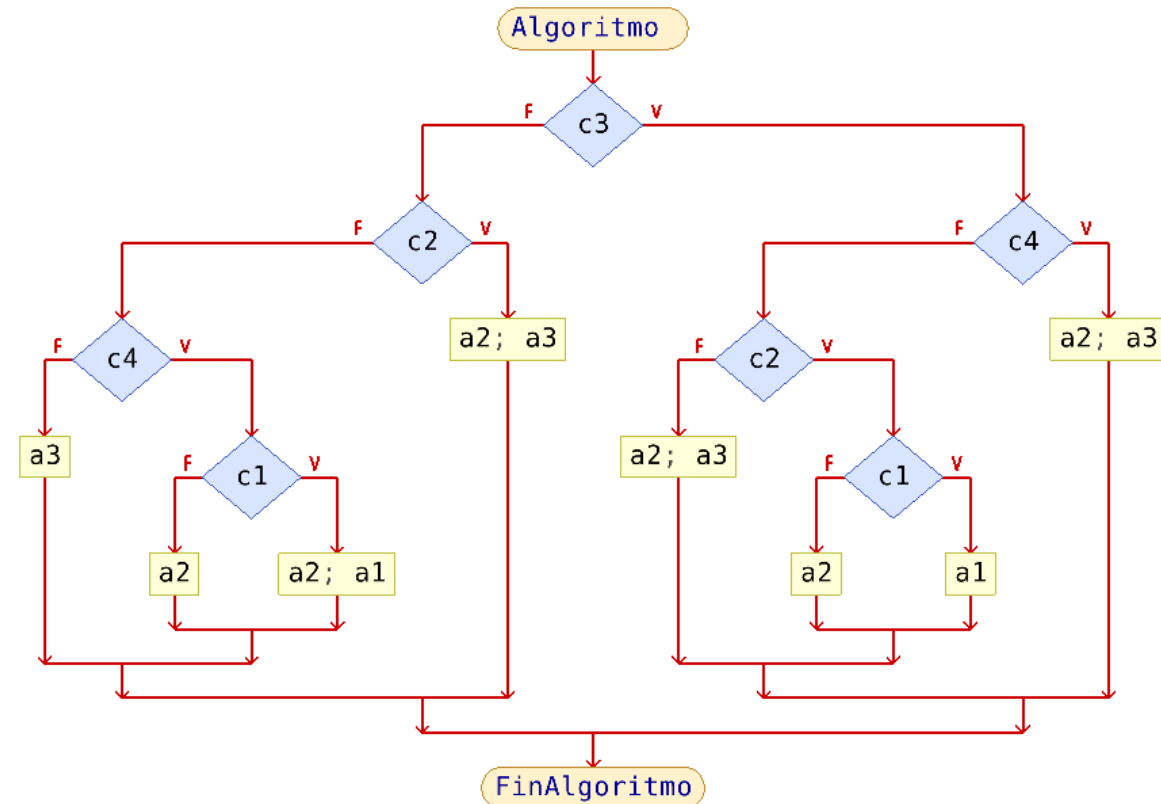





```

1  Algoritmo sin titulo
2      Si c3 Entonces
3          Si c4 Entonces
4 +             a2; a3
5          Sino
6              Si c2 Entonces
7                  Si c1 Entonces
8 +                     a1
9                  Sino
10 +                     a2
11                 FinSi
12             Sino
13 +                 a2; a3
14             FinSi
15         FinSi
16     Sino
17         Si c2 Entonces
18 +             a2; a3
19         Sino
20             Si c4 Entonces
21                 Si c1 Entonces
22 +                     a2; a1
23                 Sino
24 +                     a2
25                 FinSi
26             Sino
27 +                 a3
28             FinSi
29         FinSi
30     FinSi
31 FinAlgoritmo
32
33

```



Ejercicios

- Resolver y pasar a ordinograma la siguiente TD. Si hay redundancias incongruentes, eliminar la regla que tenga un tratamiento de mayor número. Si falta alguna situación, poner como tratamiento A4

C1	N	S	S	N	---	S	N
C2	---	S	---	S	S	N	N
C3	S	S	N	---	N	S	---
C4	S	---	S	---	N	---	---
A1	X			X			X
A2		X				X	
A3			X				
A4					X		

Ejercicios

- Escribir la TD que refleje estas circunstancias:
 - Un número natural del 0 al 15 puede ser representado en base 2 utilizando cuatro dígitos binarios (b_3, b_2, b_1, b_0). Indicar cuál es primo.
 - Una persona desea efectuar un crucero por el Mediterráneo, pero solo puede realizarlo si obtiene algún ingreso extra. Por ello, va a jugar a la lotería, quinielas y primitiva.
 - Un estudiante desea ir de vacaciones y ordena sus preferencias a la hora de comprar el billete del viaje respecto a los medios de transporte: primero avión si el billete cuesta menos de 300 €, segundo tren y tercero bus.
 - Funcionamiento de un semáforo que contemple las siguientes situaciones:
 - Amarillo intermitente → paSar
 - Verde o verde con amarillo fijo → paSar
 - Rojo o rojo con amarillo fijo → paRar

Ejercicios

Querido amigo:

*Al poco de comprar una mansión, he comprobado que está hechizada con dos sonidos que la hacen inhabitable: una risa sardónica (**R**) y un canto pionero (**C**).*

*La experiencia me demuestra que su comportamiento obedece a ciertas leyes, de modo que se puede saber un minuto después qué va a ocurrir con la risa (**R'**) y con el canto (**C'**):*

*El canto conserva un minuto después su estado presente o ausente, salvo si en el minuto actual no se oye la risa y se toca el órgano (**O**), en cuyo caso, el canto toma el estado opuesto.*

*La risa, si no se quema incienso (**I**), se oirá o no según el canto esté presente o ausente, de forma que la risa imita al canto con un minuto de retraso. Si se quema incienso, la risa hará justamente lo contrario de lo que hacía el canto.*