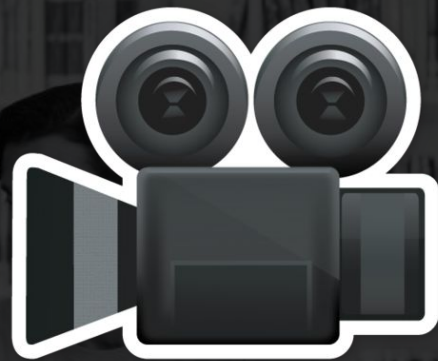


Programador COBOL

**MAINFRAME**

**Recuerda  
poner a grabar  
la clase.**





# **MÓDULO 1 – PROGRAMACIÓN ESTRUCTURADA**



## ... Mapa Conceptual

01

Tipos de  
estructuras

02

Métodos



## ... Agenda

1 Tipos de estructuras

2 Método Warnier

3 Método Jackson

4 Método Yourdon

5 Método Hipo





## ... Programación estructurada

### ¿POR QUÉ?

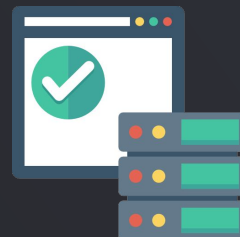
Porque un programa que está basado en una estructura lógica, con un tronco principal en secuencia que invoca secciones con lógicas aplicativas , que una vez finalizada cada una de ellas, vuelven nuevamente al tronco principal, hasta la finalización del mismo.



## ... Programación estructurada

### BENEFICIOS:

1. Fácil de seguir y mantener.
2. Permite identificar fácilmente la lógica aplicada.
3. Los errores en la ejecución del programa sean bajos.

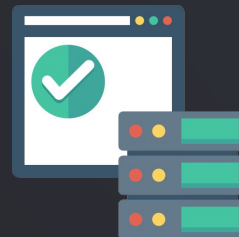




## ... Programación estructurada

### BENEFICIOS:

1. Facilita que los programas sean autodocumentados.
2. Facilita el uso de una nomenclatura en los LABELS del código fuente





## ... Programación estructurada



¿ Qué ayuda ?

El uso de sentencias de loop controlado

Que los nombres de las rutinas, tengan una identificación del nivel y lógica

Documentar que realiza cada rutina

Documentar los cambios por mantenimiento, con fecha, causas y un código de mantenimiento



## ... Programación estructurada

¿ Qué ayuda ?

Documentar los cambios por mantenimiento, con fecha, causas y un código de mantenimiento

Documentar en código fuente las sentencias de cambios por mantenimiento

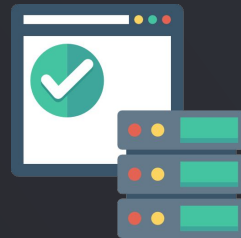
Que el programador tenga un pensamiento lógico del negocio a tratar en el programa



## ... Programación estructurada

¿ Qué evitar?

- El uso de sentencias que rompen estructuras y caminos lógicos
- El uso de sentencias de bifurcación anidadas y de lógicas complejas
- No documentar las rutinas

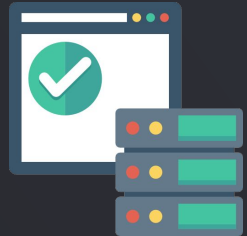




## ... Programación estructurada

¿ Qué evitar?

- No documentar los cambios por mantenimiento
- Que no exista un pensamiento lógico del negocio, en la confección del programa





## ... Programación estructurada

### TIPOS DE ESTRUCTURAS

El diseño del programa de lo general a lo particular es TOP DOWN

- Estructuras básicas:
  - Secuencial
  - Alternativa
  - Repetitiva

El programador con un pensamiento lógico del objetivo de negocio del programa, primero confecciona PSEUDOCÓDIGO



## ... Programación estructurada

### PSEUDOCÓDIGO

Es una herramienta utilizada para el diseño de programas, que permite al programador expresar sus pensamientos lógicos de una forma clara, utilizando su lenguaje materno y mostrando el orden de ejecución de las sentencias del programa sin ninguna ambigüedad.

El PSEUDOCÓDIGO, no está diseñado como lenguaje compilable.

El pasaje del programa escrito en PSEUDOCÓDIGO al programa escrito en lenguaje COBOL es simple

Para escribir un PSEUDOCÓDIGO se deben seguir reglas sintácticas

## ... Programación estructurada



### PSEUDOCÓDIGO

Según estructuras básicas

- **SECUENCIAL**, Los nombres de los campos, deben ser reales

Leer CANTIDAD, PRECIO

IMPORTE = CANTIDAD \* PRECIO

Imprimir IMPORTE

## ... Programación estructurada



### PSEUDOCÓDIGO

**ALTERNATIVA**, incluye palabras reservadas IF - THEN - ELSE

Se llama ALTERNATIVA, para indicar que según el resultado de una pregunta se toma una acción u otra.

Leer ARTICULO, CANTIDAD, PRECIO UNITARIO

$\text{IMPORTE} = \text{CANTIDAD} * \text{PRECIO UNITARIO}$





## ... Programación estructurada

### PSEUDOCÓDIGO

**ALTERNATIVA**, incluye palabras reservadas IF - THEN – ELSE (**cont.**)

IF CANTIDAD es mayor 0 AND es menor a 16 THEN

    IMPORTE = IMPORTE \* 0.9

ELSE

    IMPORTE = IMPORTE \* 0.85

ENDIF

Imprimir CANTIDAD, PRECIO UNITARIO, IMPORTE



## ... Programación estructurada

### PSEUDOCÓDIGO

**REPETITIVA**, incluyen palabras reservadas DO WHILE - ENDDO Se llama así, cuando se quiere que un conjunto de instrucciones se realice un número finito de veces.

SUMA = 0

NUM = 0

MEDIA = 0

Leer ALTURA



## ... Programación estructurada

### PSEUDOCÓDIGO

REPETITIVA, (cont)

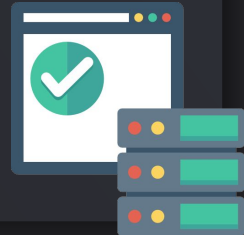
DO WHILE (ALTURA es diferente a 0)

SUMA = SUMA + ALTURA

NUM = NUM + 1

Leer ALTURA

ENDDO





## ... Programación estructurada

### PSEUDOCÓDIGO

#### REPETITIVA, (cont)

MEDIA = SUMA / NUM

Imprimir MEDIAE

## ... Programación estructurada

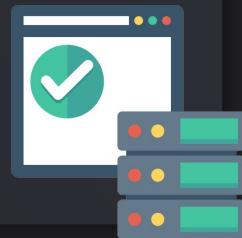


### PSEUDOCÓDIGO

### BIFURCACIÓN

Las BIFURCACIONES pueden ser:

- Bifurcación condicional
- Bifurcación incondicional o de invocación





## PSEUDOCÓDIGO

**Bifurcación condicional**, son las ESTRUCTURAS ALTERNATIVAS

IF condición THEN acción

IF condición THEN acción-1 ELSE acción-2

**Bifurcación incondicional** o de invocación, pasa el control a la sentencia etiquetada

GO TO acción-3



## ... Mapa Conceptual

01

Tipos de  
estructuras

02

Métodos





# Programación Estructurada





## ... Puntos a desarrollar

### 1. MÉTODOS DE DISEÑO ESTRUCTURADO

#### 2. Método WARNIER

#### 3. Método JACKSON

#### 4. Método YOURDON

#### 5. Método HIPO



## ... Programación estructurada

### FORTALEZAS DE CADA MÉTODO

Los 4 métodos tienen 2 características comunes:

Organización Jerárquica: Las funciones y subfunciones están organizadas en forma jerárquica

Modalidad de programación estructurada: Utilizan

- Secuencia
- Selección
- Iteración
- Invocación

para controlar las ejecuciones de las sub funciones



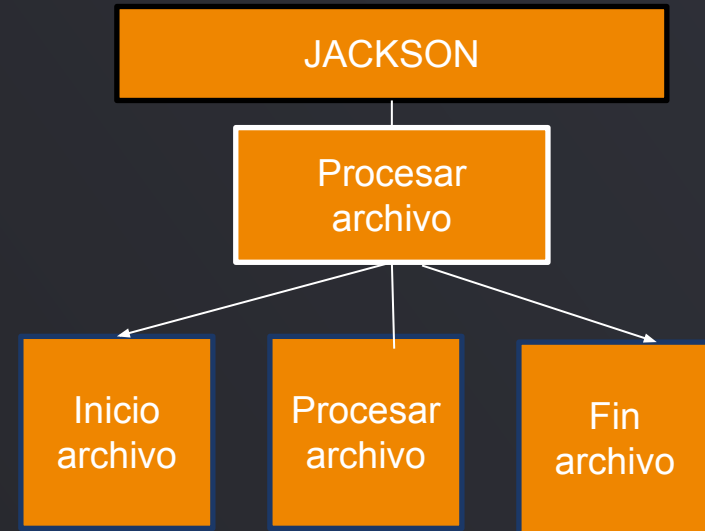
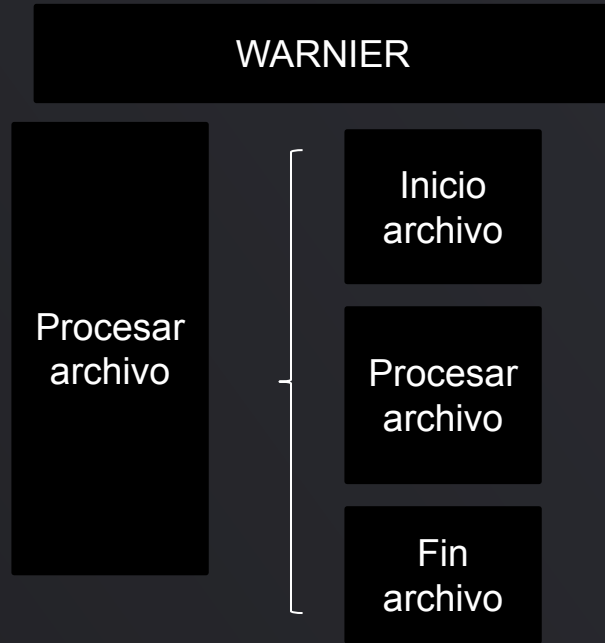
## ... Programación estructurada

### 1. MÉTODOS DE DISEÑO ESTRUCTURADO

Los métodos de WARNIER y JACKSON, las funciones de los programas son diseñados en base a las estructuras de datos.

Ambos métodos están estrechamente relacionados, la diferencia radica en la documentación.

## ... Programación estructurada





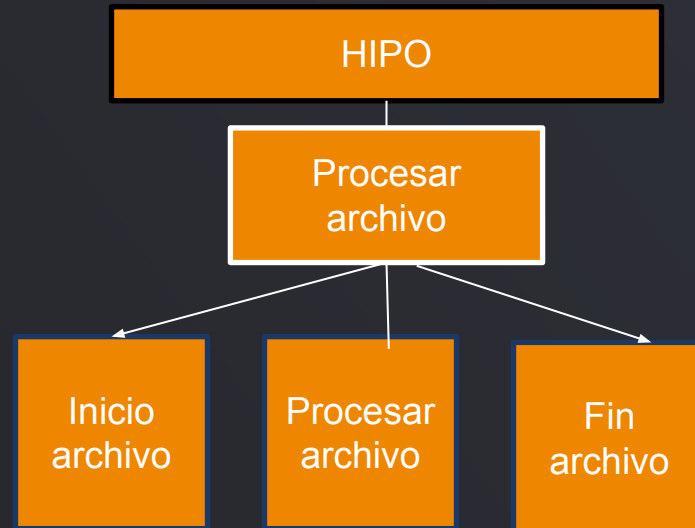
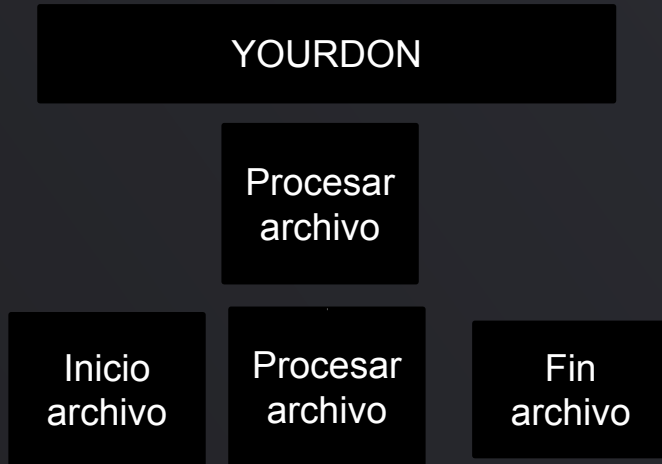
## 1. MÉTODOS DE DISEÑO ESTRUCTURADO

Los métodos de YOURDON Y HIPO, son también muy similares.

Ambos utilizan análisis del flujo de datos, para determinar las funciones que el programa necesita para transformar los datos de modo que produzcan información útil.

También los dos métodos difieren en su documentación.

## ... Programación estructurada





## ... Programación estructurada

**El método WARNIER** de desarrollo de funciones de programas se basa en la estructura de datos. Fue desarrollado por JEAN DOMINIQUE WARNIER mientras trabajaba en la empresa BULL en Francia.

WARNIER utiliza llaves para identificar niveles dentro de una jerarquía.

Dentro de la anotación de WARNIER cualquier función dentro de una llave es invocada por la función que la abraza.

Cada invocación de una función corresponde a una de las estructuras básicas de la programación estructurada: secuencia, iteración y selección.



## ... Programación estructurada

### 2. MÉTODO DE WARNIER - VENTAJAS

- Diseño basado en la estructura de datos, lo que simplificará la etapa inicial de los datos.
- Organización jerárquica a partir de la cual será fácil desarrollar la codificación estructurada.
- Es fácil aprenderla ya que su anotación es uniforme a todos los niveles.
- Es fácil de modificar.
- Uniformidad de diseño que facilita la comunicación entre los usuarios y los desarrolladores de sistemas.



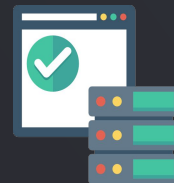


## ... Programación estructurada

### 2. MÉTODO DE WARNIER

#### DESVENTAJAS

- No prevé pautas para resolver los conflictos entre estructuras
- No brinda ilustración visual del flujo de datos





## ... Programación estructurada

### 2. MÉTODO DE WARNIER

#### Permite:

- Mostrar un diseño estructurado que sirva como herramienta de programación.
- Aportar un conocimiento operativo que permita una rápida asimilación y puesta en práctica a través de la resolución de casos prácticos.

Un programa ilustrado por un PSEUDOCÓDIGO debe contar con un FLUJO de CONTROL.



## ... Programación estructurada

### 2. MÉTODO DE WARNIER

- El FLUJO de CONTROL de contar con módulos INVOCANTES y módulos INVOCADOS.
- Un módulo INVOCANTE es un bloque de PSEUDOCÓDIGO, que da el control a un módulo INVOCADO.
- Un módulo INVOCADO es un bloque de PSEUDOCÓDIGO, que recibe el control de un módulo INVOCANTE.



## 2. MÉTODO DE WARNIER

Los módulos brindan cierto grado de organización, pero no garantizan que el control del programa sea entregado en forma ordenada.

Ya que los módulos no pueden tener más de una entrada/salida, lo que a su vez crea problemas complejos de control.

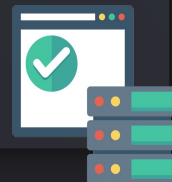
Una estructura de control programada ordenada, el control nunca pasa de un módulo a otro del mismo nivel o de niveles inferiores. Siempre devuelven el control al módulo invocante (que es de nivel superior, o sea pertenece al tronco principal de la estructura). Pero las funciones del PSEUDOCÓDIGO pueden estar dispersas en distintos módulos.

## ... Programación estructurada



### 2. MÉTODO DE WARNIER

Los diagramas estructurados son diseñados de izquierda a derecha y de arriba hacia abajo, llamados diagramas jerárquicos, normalmente se transforman en el flujograma del PSEUDOCÓDIGO.



## ... Programación estructurada



### 2. MÉTODO DE WARNIER

INICIO-PGMA.

READ una transacción

READ un registro maestro

IF NOT fin-maestro AND NOT fin-transacción

THEN DO actualización-items-maestros UNTIL  
fin-de-los-dos

FIN-PGMA.

STOP.

ACTUALIZACIÓN-ITEMS-MAESTROS.

IF clave mastro es menor

WRITE nuevo maestro FROM viejo maestro

READ un registro maestro



## ... Programación estructurada



### 2. MÉTODO DE WARNIER (cont.)

IF clave-trans es menor

WRITE nuevo maestro FROM trans

READ una transacción

IF clave-maestro EQUAL clave-trans DISPLAY  
'CLAVE YA EXISTE EN MAESTRO'

WRITE nuevo maestro FROM viejo  
maestro

READ una transacción

READ un registro maestro

FIN-ACTUALIZACIÓN-ITEMS-MAESTROS.

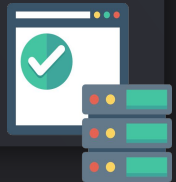


## ... Programación estructurada



### 2. MÉTODO DE WARNIER

Lo primero que se debe definir es el objetivo principal del programa. Por ejemplo 'FORMATEAR DIRECCIÓN POSTAL'. Es una estructura secuencial, tendríamos una ESTRUCTURA SECUENCIAL y un FLUJOGRAMA.





## ... Programación estructurada



### Estructura secuencial

Formatear  
dirección postal

- Inicio
- Formatear nombre
- Formatear calle
- Formatear ciudad / provincia
- Formatear código postal
- Fin

### Diagrama de Flujo

Inicio

Formatear  
nombre

Formatear calle

Formatear  
ciudad/provincia

Formatear código  
postal

Fin

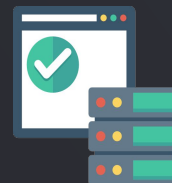
## ... Programación estructurada



### 2. MÉTODO DE WARNIER

Es una estructura de iteración:

Hay una ESTRUCTURA ITERACIÓN y un FLUJOGRAMA ESTRUCTURA DE ITERACIÓN





## 2. MÉTODO DE WARNIER

### Estructura de iteración

Procesar  
empleado (0,n  
veces. Hasta  
fin empleados

Calcular sueldo bruto  
Calcular impuestos  
Calcular sueldo neto =  
sueldo bruto - impuestos

### Diagrama de flujo





## 2. MÉTODO DE WARNIER

Las iteraciones o 'LOOP', se especifican mediante sentencias que tienen 3 partes:

- Nombre de LOOP, Procesar Empleado
- Factor de repetición (0, n )
- La condición del fin (hasta el fin de los empleados)



## 2. MÉTODO DE WARNIER

En el LOOP es importante el factor de repetición. Dado que podemos tener:

- Iteración Definida, el número de iteración es fija (ejemplo 10 veces)
- Iteración indefinida, varía el número de veces de una iteración, y el fin de la misma será por condición esperada. Si la misma falla, el LOOP nunca terminará



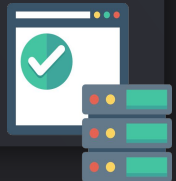
## 2. MÉTODO DE WARNIER

Ejemplos de Indefinidas de 0 a n veces

de 1 a n veces

de 0 a 10 veces, con tope máximo

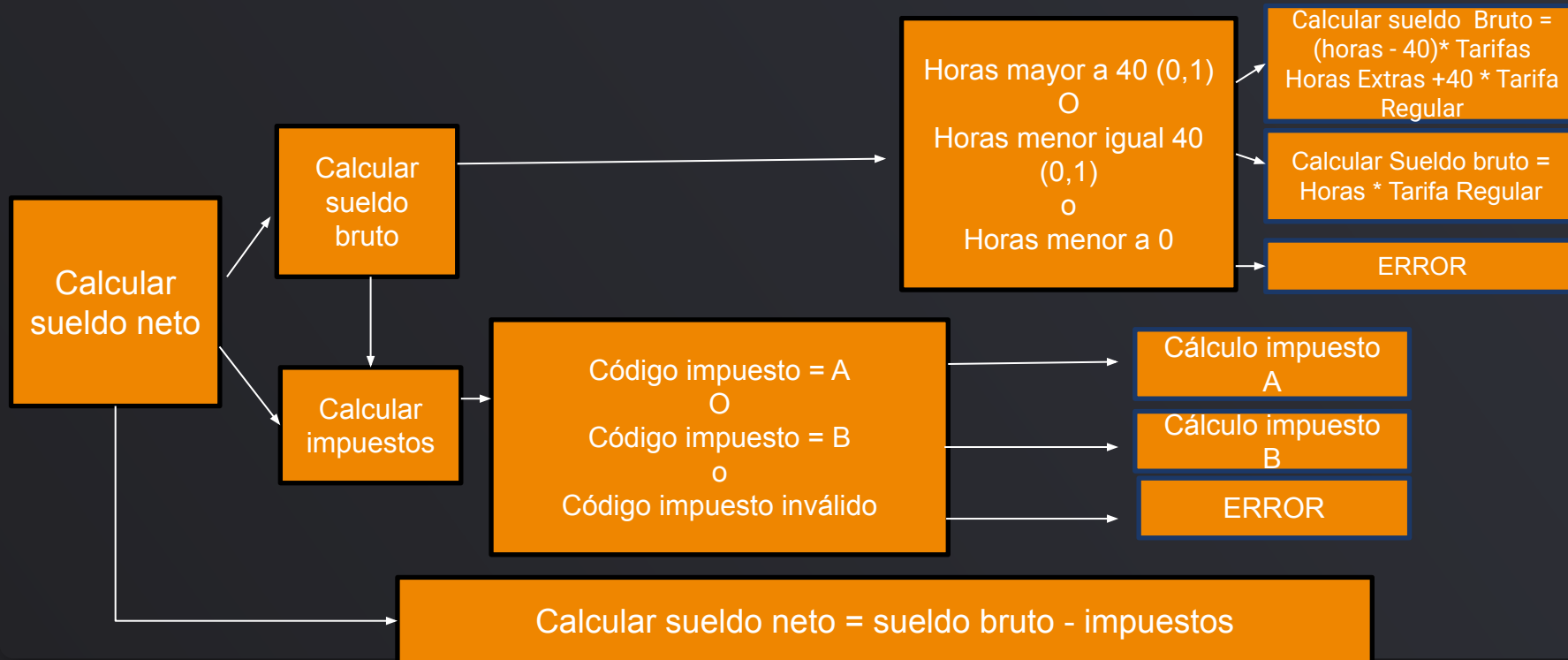
de 1 a 10 veces, con tope mínimo y máximo



## ... Programación estructurada



### 2. MÉTODO DE WARNIER



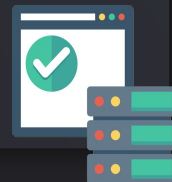


## 2. MÉTODO DE WARNIER

### DIAGRAMAS DE ESTRUCTURAS

Pasos para desarrollar Diagramas de Estructuras

- Desarrollar diagrama de estructura
- Dibujar llaves
- Listar los ítems del siguiente nivel
- Sub definir los ítems con llaves a menor nivel
- Describir la lógica
- Referencias documentación adicional







## 2. MÉTODO DE WARNIER

### DISEÑO ESTRUCTURADO DE PROGRAMAS CON PSEUDOCÓDIGO

Para ello debemos contar con:

- Definición de las funciones de negocio
- Definición de los datos para las funciones de soporte
- Diseño de las estructuras de datos de entrada y salida

### OBJETIVO

Diseñar una estructura de programas sobre la base de pautas lógicas de diseño, proporcionando documentación de Diseño Preliminar adecuada. Documentar dicho programa utilizando un diagrama de estructuras



## 2. MÉTODO DE WARNIER

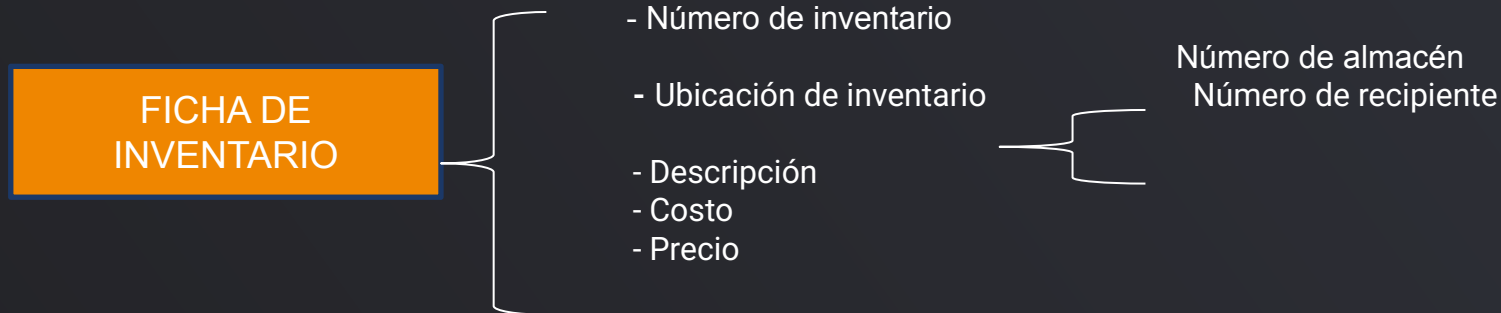
### DISEÑO ESTRUCTURADO DE PROGRAMAS CON PSEUDOCÓDIGO

Pasos del diseño estructurado:

1. Describir la estructura lógica de datos, tanto de entrada como de salida, a ser procesador por el programa
2. Formar una estructura lógica de programa en base a la estructura lógica de datos.
3. Extender la estructura lógica del programa dando consideración a los aspectos de estructura de datos y a los aspectos funcionales.
4. Empaquetar la estructura lógica del programa dentro de un juego de programa físicos.



## 2. MÉTODO DE WARNIER





## 2. MÉTODO DE WARNIER

### DISEÑO ESTRUCTURADO DE PROGRAMAS CON PSEUDOCÓDIGO

Una vez que tenemos descrito un archivo con datos de una misma naturaleza, debemos definir si el mismo es:

- Un Archivo Tradicional
- Una base de datos

Si es un archivo tradicional indexado o base de datos, debemos identificar qué dato va a ser clave primaria

La clave primaria debe identificar al registro del archivo por su naturaleza. Debemos identificar si la clave primaria es de un campo simple o es una clave estructurada por varios campos.

Luego al archivo, le debemos aplicar la primera forma normal, o sea identificar si contiene vectores, si es así sacar dicho vector para conformar otra identidad o archivo relacionándolo con el archivo primario. De esta forma evitamos archivos de longitud variable.



## 2. MÉTODO DE WARNIER

### DISEÑO ESTRUCTURADO DE PROGRAMAS CON PSEUDOCÓDIGO

Luego para los archivos indexados o base de datos, le debemos aplicar la segunda forma normal. O sea que todos los datos que conforman el archivo, dependen de la clave primaria y no de otro dato del archivo. Si fuera así se los deberá separar en otra entidad o archivo.

Si la clave es estructurada, todos los datos deben depender de toda la clave estructurada y no parcialmente.

Si el archivo es indexado o base de datos, identificar la necesidad o no de definir claves alternativas.



## 2. MÉTODO DE WARNIER

### DISEÑO ESTRUCTURADO DE PROGRAMAS CON PSEUDOCÓDIGO

Una vez definidas las estructuras lógicas de datos de entrada y salida. Podemos pasar al segundo paso en formar una estructura lógica de programas.

Si las estructuras lógicas de datos de entrada y salida son similares en base a:

- Las estructuras tienen el mismo número de niveles y/o
- Los mismos ítems están representados a los mismos niveles

Entonces, **la estructura del programa puede basarse en la estructura lógica de datos de salida.**

Si las **estructuras lógicas de datos de entrada y salida son diferentes**, entonces existe un conflicto de estructuras. Entonces el programa deberá tener una función que transforma la estructura de los datos de entrada en la estructura de datos de salida.

Para ello, **el programa se basará en la estructura de datos de entrada, para definir la estructura lógica del programa.**



## ... Programación estructurada

### 2. MÉTODO DE WARNIER

#### DISEÑO ESTRUCTURADO DE PROGRAMAS CON PSEUDOCÓDIGO

En la confección de una estructura lógica del programa, siempre vamos a contar con dos RUTINAS:

- INICIO
- FIN

**En INICIO, puede realizar distintas funciones.**

#### 1. Apertura de archivos de entrada/salida

- Se deberá analizar el código de retorno de dicha apertura.
- Si el código de retorno es diferente a cero puede ser que la longitud del archivo definido en el programa, difiere la longitud que tiene el archivo físico. Si es un proceso BATCH, puede ser que la DDNAME del JOB sea diferente a la asignada por el programa. También puede ser que difiera la organización definida en el programa.

#### 2. Validación de parámetros ingresados.



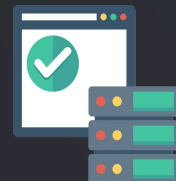
## 2. MÉTODO DE WARNIER

### DISEÑO ESTRUCTURADO DE PROGRAMAS CON PSEUDOCÓDIGO

En FIN, puede realizar distintas funciones.

- Cierre de archivos de entrada/salida
- Si es BATCH emisión de totales de control
- Si es BATCH emisión de código de retorno
- Fin de ejecución del programa

Entre ambas INICIO y FIN, va a existir la RUTINA de función de PROCESO







## 2. MÉTODO DE WARNIER

### DISEÑO ESTRUCTURADO DE PROGRAMAS CON PSEUDOCÓDIGO (cont.)

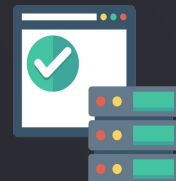
Todas las RUTINAS serán invocadas, con PERFORM, para que una vez que termina las ejecuciones de RUTINAS invocadas, vuelve a la RUTINA invocante (explicar a detalle)

Cuerpo o tronco principal del programa.

PERFORM INICIO THRU FIN-INICIO

PERFORM PROCESO THRU FIN-PROCESO

PERFORM FINAL THRU FIN-FINAL





## ... Programación estructurada

### 3. MÉTODO JACKSON

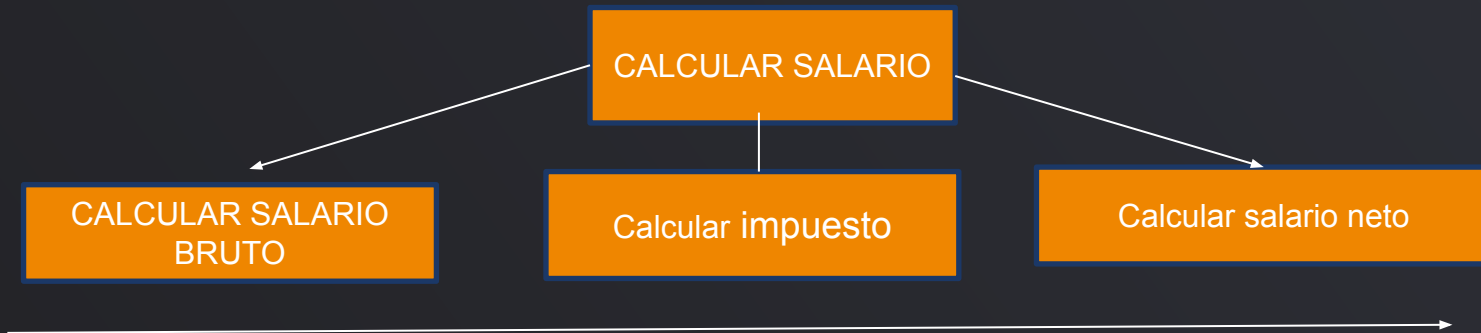
- El método de JACKSON fue desarrollado en la década del 70' por MICHAEL JACKSON en Inglaterra.
- El método de JACKSON es igual al de WARNIER y se basa en la estructura de datos.
- Se fundamenta en su diseño jerárquico de las funciones y subfunciones de los programas con estructuras de datos de entrada y salida. Para el método de JACKSON el programa es el medio por el cual los datos de entrada son transformados en datos de salida.
- Incorpora las tres estructuras básicas de secuencia, selección e iteración al invocar funciones y subfunciones.
- En la estructura del programa las sub funciones se procesarán de izquierda a derecha en todos los niveles.



### 3. MÉTODO JACKSON

Ejemplo de estructura de secuencia. Siempre su orden es de arriba hacia abajo y de izquierda a derecha.

En este ejemplo CALCULAR SALARIO, primero realizará CALCULAR SALARIO BRUTO, luego CALCULAR IMPUESTOS y por último CALCULAR SALARIO NETO.

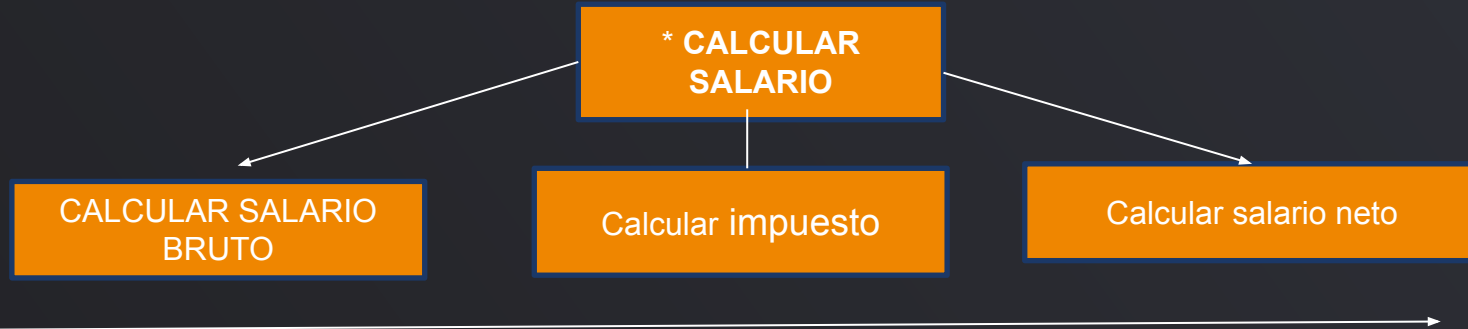




### 3. MÉTODO JACKSON

Ejemplo de estructura de iteración. La iteración está marcada con un (\*), representando n veces.

En este ejemplo CALCULAR SALARIO, primero realizará la **subfunción CALCULAR SALARIO**, que incluye CALCULAR SALARIO BRUTO, luego CALCULAR IMPUESTOS y por último CALCULAR SALARIO NETO.

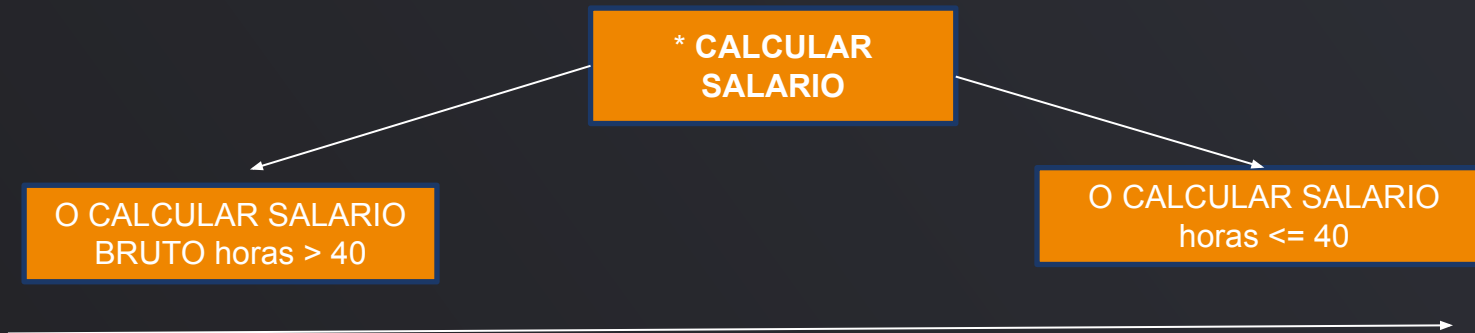




### 3. MÉTODO JACKSON

Ejemplo de estructura de selección. La selección está marcada con un 'O', que significa optativa.

En este ejemplo CALCULAR SALARIO, Si el empleado trabajó más de 40 horas realizará la sub función CALCULAR SALARIO BRUTO horas > 40; caso contrario realizará CALCULAR SALARIO horas <= 40





## ... Programación estructurada

### 4. MÉTODO YOURDON

Al igual que el método HIPO, se basa en el análisis del flujo de datos. Los datos son transformados en funciones y subfunciones pasando de los datos de entrada hasta los datos de salida.

Fue desarrollado por LARRY CONSTANTINE a mediados de la década de los 60'

El método de YOURDON, usa dos tipos distintos de documentación

- Diagramas de Flujo de Datos
- Diagramas de Estructuras

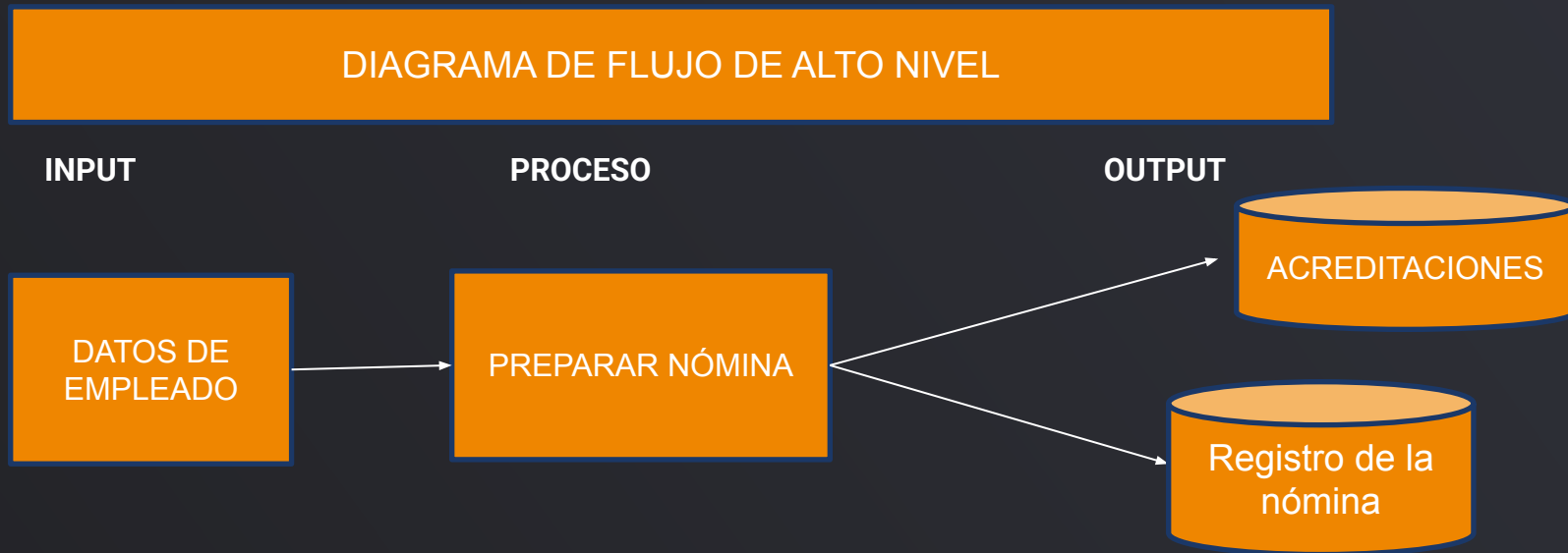
Del flujo de los datos, identifica las funciones que transforman los mismos, presentándola con un círculo.

Y el ingreso o salida de datos de una función son representados por una línea.

Los diagramas de flujo de datos se elaboran a un nivel muy alto, y luego se descomponen en diagramas de mayor detalle. Y posteriormente con las funciones representadas, se elabora diagrama de estructura



## 4. MÉTODO YOURDON





## ... Programación estructurada

### 4. MÉTODO YOURDON

El siguiente paso es descomponer el diagrama de flujo de datos de alto nivel en niveles más detallados.

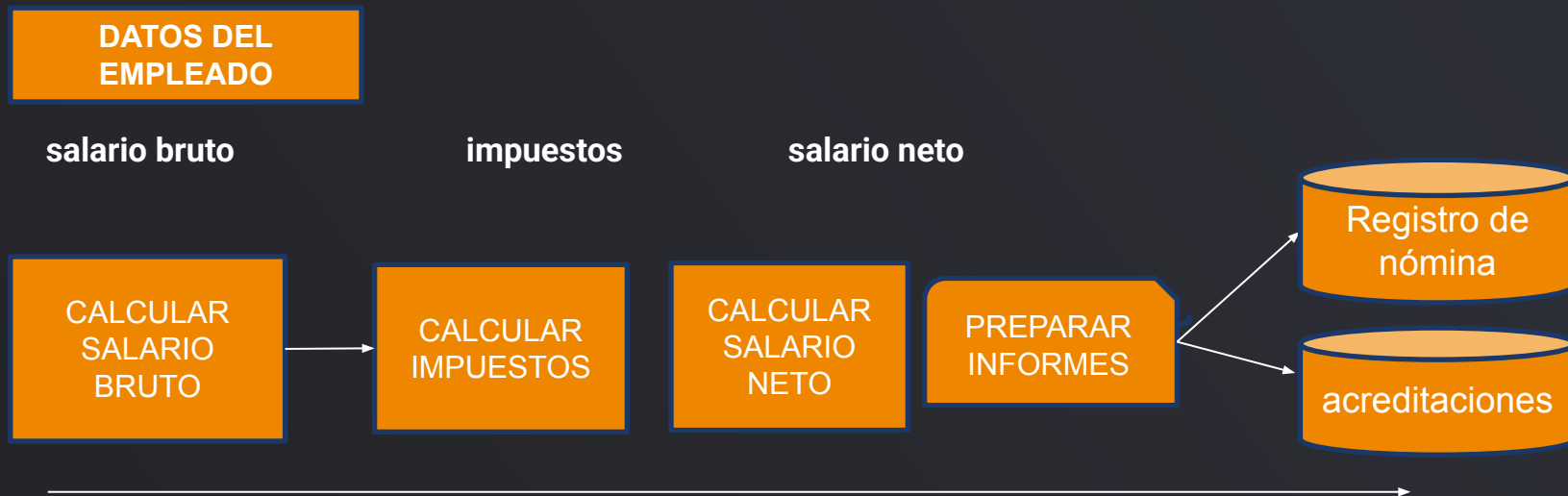
La función PREPARAR NÓMINA, puede ser descompuesta en las siguientes subfunciones:

- Calcular Salario Bruto
- Calcular Impuestos
- Calcular Salario Neto
- Preparar Informes





## 4. MÉTODO YOURDON



## ... Programación estructurada

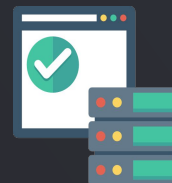


### 4. MÉTODO YOURDON

En la estructura de secuencia, el orden es de arriba hacia abajo y de izquierda a derecha.

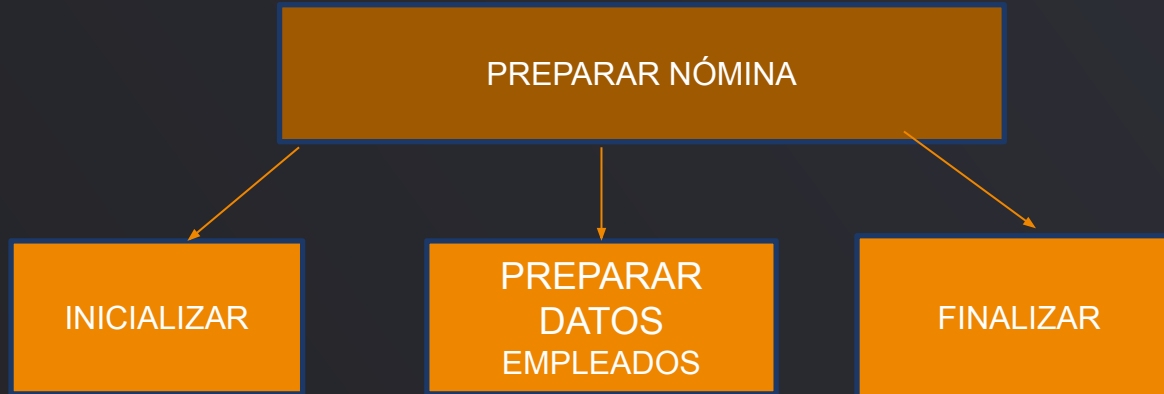
En el ejemplo PREPARAR NÓMINA, la secuencia es

- Inicializar
- Preparar Salarios Empleados
- Finalizar





## 4. MÉTODO YOURDON





## 4. MÉTODO YOURDON

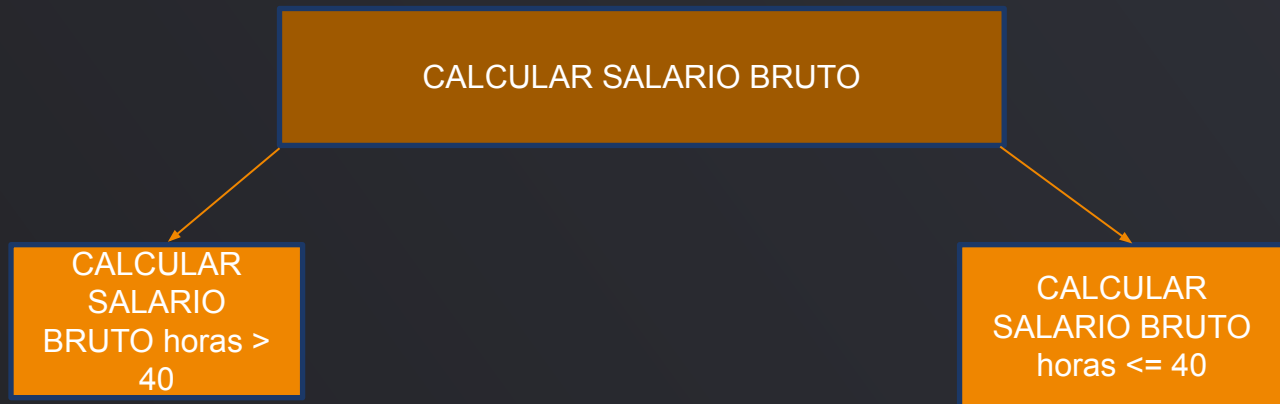
En la estructura de selección, es indicada con un rombo.

En el ejemplo PREPARAR NÓMINA, la selección, da que la función de decisión está contenida en la función Calcular Salario Bruto, y el rombo señala que se selecciona o bien cálculo con horas mayor a 40 o calcular con horas menor a igual a 40

- Calcular Salario Bruto
- Calcular Salario Bruto mayor a 40
- Calcular Salario Bruto menor igual a 40



## 4. MÉTODO YOURDON





## 4. MÉTODO YOURDON

### VENTAJAS

- El método conduce a programas estructurados; utiliza las estructuras de control básicas del diseño estructurado.
- Los criterios para evaluar el diseño brindan, a la vez un control de calidad.
- El método muestra el flujo de datos en el diagrama de estructuras que facilita la tarea de revisión.

### DESVENTAJAS

- El aprendizaje puede ser difícil, porque no es fácil convertir los diagramas de flujo de datos en diagramas de estructuras.
- Los distintos diseñadores que utilicen el método producirán diseños poco uniformes porque el enfoque es más intuitivo y depende más de la experiencia.
- Es más difícil mantener y usar la documentación, ya que los diagramas de estructuras pueden contener demasiado detalle



## 5. MÉTODO HIPO

El método de HIPO fue diseñado por IBM, y su sigla significa Hierarchy Input Proceso Output

El método de HIPO es igual al de YOURDON en las técnicas de diseño estructurado.

El método HIPO consiste de un juego de diagramas que describen las funciones del sistema desde el nivel más alto hasta un nivel de detalle. Las funciones son descompuestas sin consideración estrictas de las estructuras de datos.

El método HIPO tiene integrado tres tipos de diagramas que describen el sistema a niveles de distintos detalles

- Tabla visual de contenido
- Diagramas de Input-Proceso-Output
- Diagramas panorámicos
- Diagramas Detallados



## 5. MÉTODO HIPO

Este método está respaldado por 2 niveles de diagramas Input-Proceso-Output

- El nivel más alto se llama 'Diagrama HIPO Panorámico'
- El nivel más bajo se llama 'Diagrama HIPO Detallado'

Los módulos están enumerados. Dichos números llevan a su vez referencias cruzadas con los niveles inferiores de detalle en los diagramas Input-Proceso-Output.

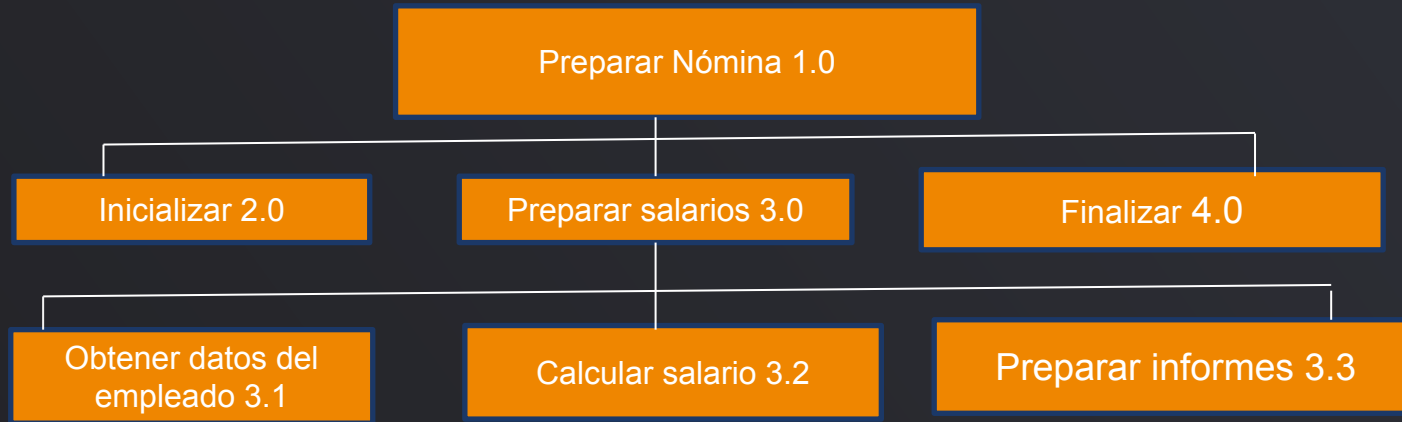
Los diagramas HIPO son estáticos y necesitan flechas para indicar los flujos de datos.





## 5. MÉTODO HIPO

### TABLA VISUAL DE CONTENIDOS





## 5. MÉTODO HIPO

Este método está respaldado por 2 niveles de diagramas Input-Proceso-Output

- El nivel más alto se llama 'Diagrama HIPO Panorámico'
- El nivel más bajo se llama 'Diagrama HIPO Detallado'

Los módulos están enumerados. Dichos números llevan a su vez referencias cruzadas con los niveles inferiores de detalle en los diagramas Input-Proceso-Output.

Los diagramas HIPO son estáticos y necesitan flechas para indicar los flujos de datos.



## 5. MÉTODO HIPO

### VENTAJAS

- Reúne muchas de las ventajas que brindan los demás métodos de diseño estructurado de programas, incluyendo la estructura jerárquica y funciones bien definidas.

### DESVENTAJAS

- Son similares a las descritas al tratar el método YOURDON. Y además la documentación HIPO es demasiado voluminosa y difícil de mantener.



**¿Consultas  
?**

**¡Muchas  
gracias!**

