Bueno, mientras escribo esta clase estamos llegando al gran número de 2000. Es impresionante decir que ya he superado la cantidad de alumnos de grandes instituciones. No esta bien compararlos pero me da felicidad jajajajaj:D.

Ah, y casi me olvido de comentarles que desde esta lectura en adelante, subo directamente los mismos cursos descargables en PDF para que cualquiera pueda leerlos. También lo estoy haciendo con las otras clases pero lleva tiempo y todavía no tengo todas hechas (aún no llegué a la 10).

De más está decir que me disculpen por la falta de dedicación al curso pero es que últimamente tengo demasiado trabajo encima y me está apretando el tiempo. ^^!

A ver de qué se trata esto del título que puse. **Todo ésto se trata de estructuras de control.**
La más simple es la estructura "**if**". Nosotros ya hemos visto esta estructura en las clases anteriores. Nuestra estructura decía:

si num_al = num_us, entonces
    "le has dado al numero"
    sino has esta otra cosa

Esta toma de decisión simple es el famoso "**if**". En inglés, equivale al "**si**" **condicional**.
Entonces:

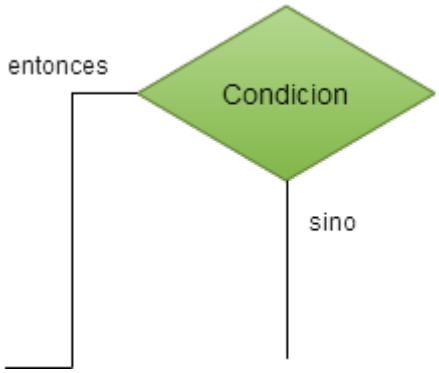
if num_al = num_us, entonces
    "le has dado al numero"
    sino has esta otra cosa

¿Entienden? Bueno, el tema está en que el "**if**" (y se los sigo repitiendo para que lo recuerden), se repite y **funciona** para **cualquier lenguaje** de programación. Aunque quizás cambien en la forma de llamarlo o de comandarlo, siempre estará. Por lo tanto es una herramienta que podremos usar cuando queramos.

Igualmente, la estructura entera es, en realidad, "**if else**". Es decir:

if condicion se cumple
    entonces haz ésto
    sino haz esto otro

El "**sino**" sería reemplazado por su hermano inglés "**else**", para que entiendan a donde voy.
Y como ya vimos en el diagrama de flujos sería:



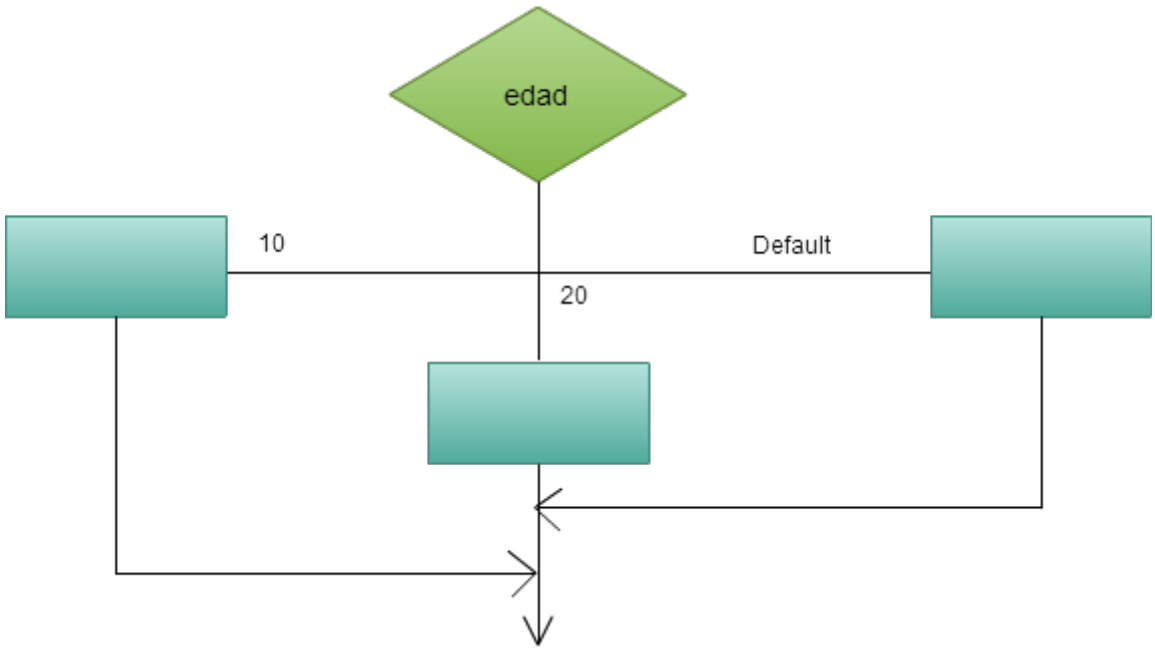
Pero esto no es todo. Aún tenemos mucho que ver. Por ejemplo, ¿Qué pasaría si en vez de cumplir o no, una sola condición, jugásemos con **más de una condición al mismo tiempo**? El anterior caso no podríamos usarlo. Para ésto, existe el "**switch case**". ¿Qué es? Digamos que la estructura habla por si misma:

switch edad

    case 10: "tenes 10 años"
    case 20: "tenes 20 años"
    case 30: "tenes 30 años"
    default: "no tenes ni 10, ni 20, ni 30 años"

No voy a seguir pero podría poner todas las condiciones que quisiera. En el **switch**, únicamente pongo cuál va a ser aquel que puede variar entre uno u otro valor. En los **case**, coloco los valores y su respectiva acción. El **default** está por si ninguna de las condiciones anteriores se cumplen.

El diagrama correspondería a una cosa así:



Y obviamente faltan más estructuras de control. Sigamos:D
Supongamos que queremos que una serie de comandos **se repitan hasta cierto momento**. Sería engorroso hacerlo con condicionales del tipo "**if else**" así que para ésto se creó el "**while**".
Algo pasa mientras se cumpla cierta condición. En términos de **pseudocódigo** veríamos algo así:

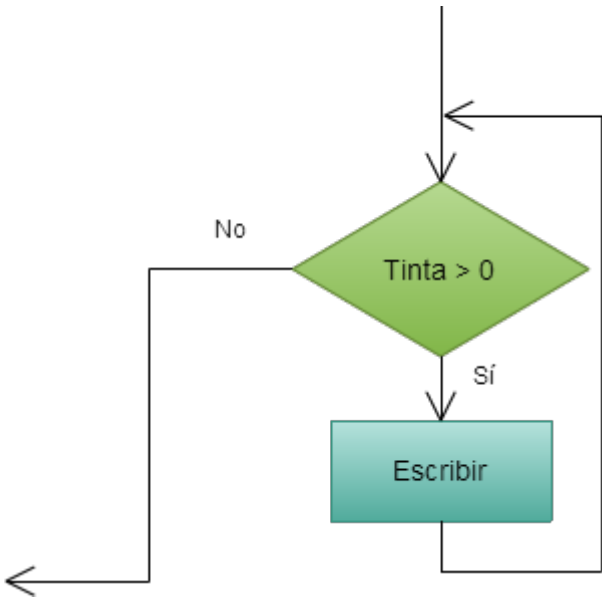
```
while condicion
        se hace esto
```

Intentemos de verlo en un **ejemplo real**. Digamos que queremos escribir hasta que siga habiendo tinta en el tintero:

```
while tinta > 0
        escribir
```

¿Ven qué **fácil**?:) Éstas son parte de nuestro kit de navajas suizas para programar, para idear nuestros programas.

En términos de **diagrama de flujos** veríamos ésto:

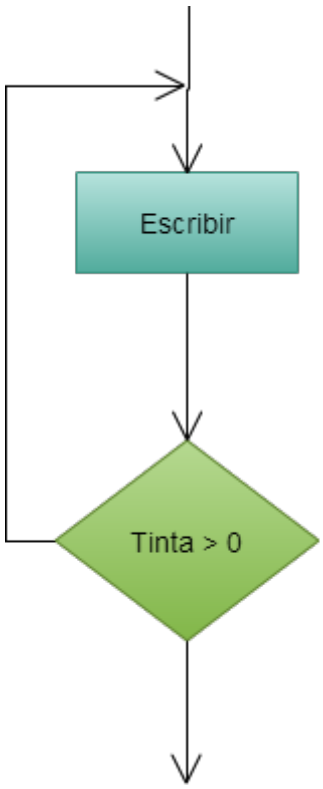


Y los hermanos sean unidos, tenemos al "**do while**" (creo que puse más comillas en esta lectura que en cualquiera de las otras sumadas). Éste es muy parecido al anterior, el tema sería que se **hace algo y se repite mientras tanto se cumpla cierta condición.** La estructura del **pseudocódigo**:

```
do
        esto
while condicion
```

¿Cuál es la diferencia de la anterior? Que en ésta, por lo menos se ejecuta **una vez.** Si en el anterior, no teníamos tinta desde un principio, no escribiremos nada. En cambio, en ésta primero se ejecuta una vez la instrucción (o las instrucciones), y **luego se comprueba el valor de la condición**.

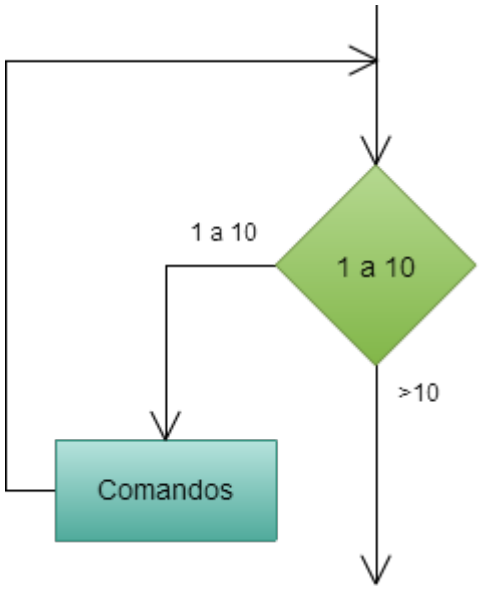
Volvamos a revisar el diagrama de flujos:



Pero aún podemos realizar más **bucles**. Digamos que queremos que una serie de comandos se repitan una cantidad de veces específicas. De manera elegante, podemos utillizar un "**for**". La **pseudosentencia** sería:

for 1 to 10 do
    comandos

Se **repitiría 10 veces** el comando ése. Simple, no es necesario especificar mucho su funcionamiento.
Veamos el **diagrama de flujos**:

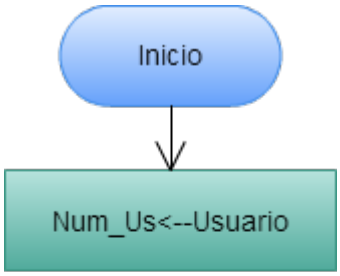


Ahora que terminamos de ver todos las estructuras de control que tenemos repartidos, necesitamos

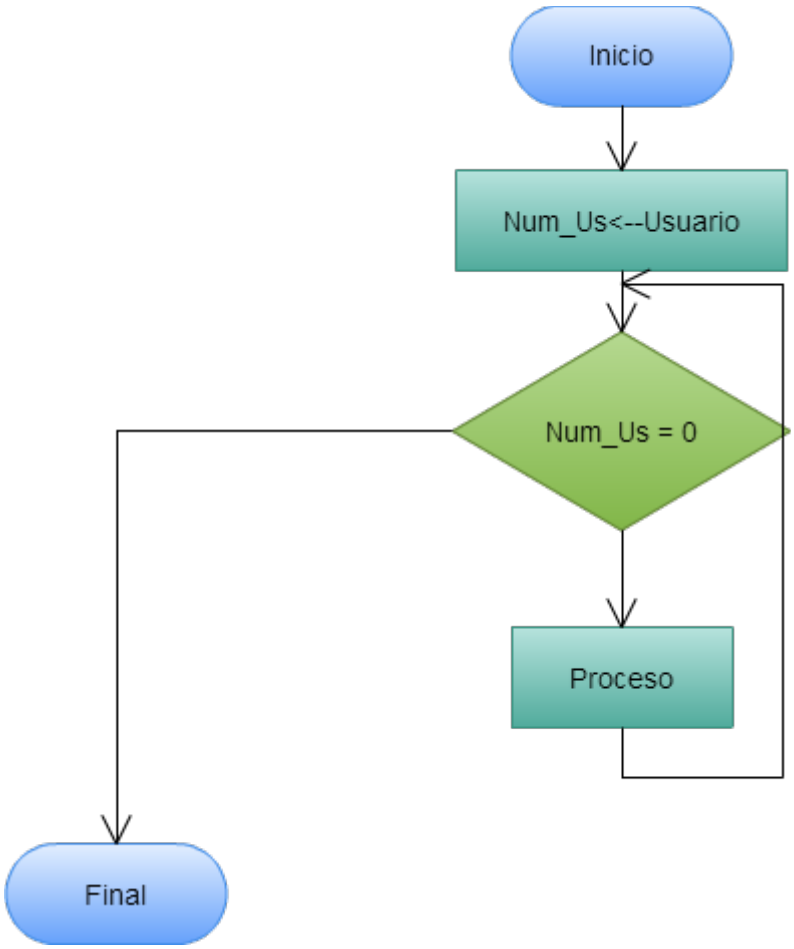
aplicarlos con un **gran ejemplo:D.**

EJEMPLO
Se debe automatizar una tarea que realice lo siguiente: el usuario ingresa la cantidad de números que quiera, y el programa deberá mostrar los 5 resultados, desde el número normal hasta 5 veces restado por un quinto de éste si es más alto de 100; deberá mostrar los 5 resultados, desde el número normal hasta 5 veces multiplicado por la mitad de éste, si es mayor a 50 y menor a 100; y deberá mostrar los 5 resultados, desde el número normal hasta 5 veces potenciado por 2, si el número es menor a 20. Si el número está entre 20 y 50, deberá haber un cartel que diga "Número incorrecto". El programa termina si el usuario ingresa 0.

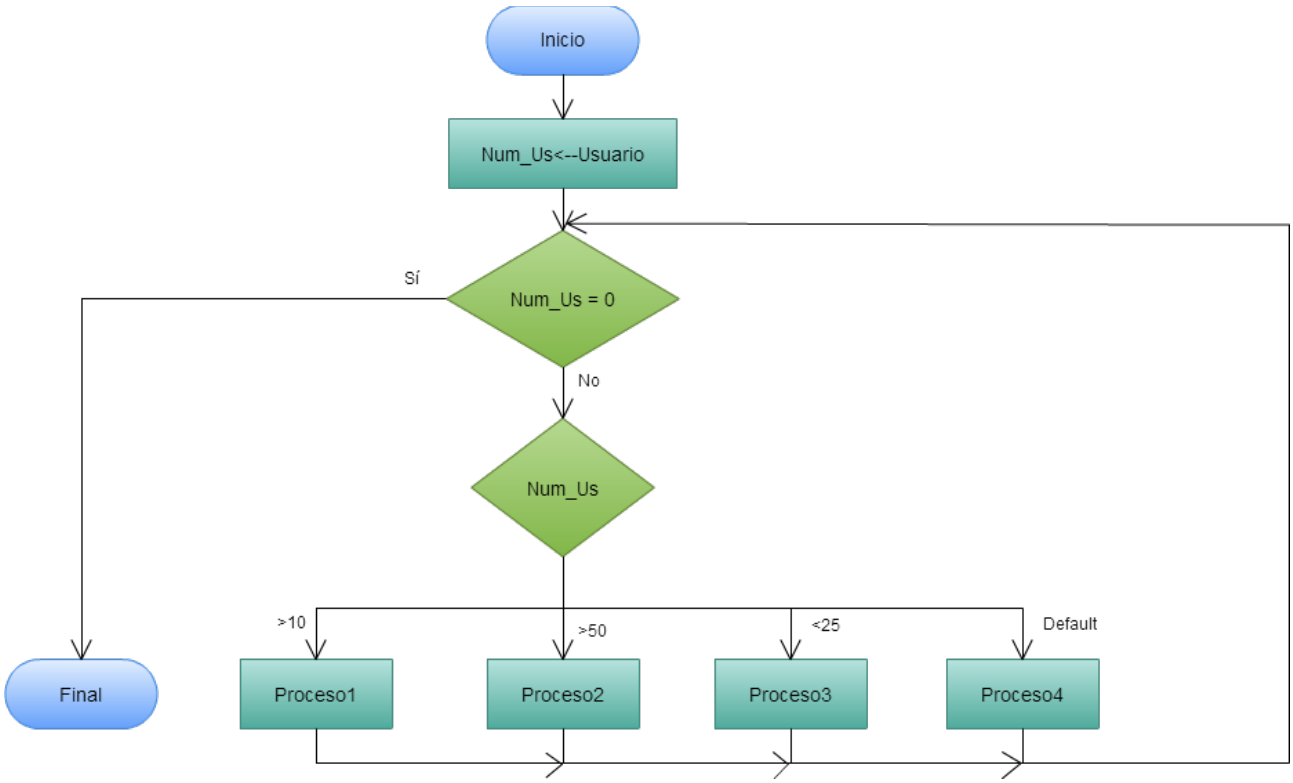
Bueno, primero vamos a hacer el diagrama de flujos. Lo primero que tenemos que hacer es crear el **inicio y el final**. Además, sabemos que lo primero que sucederá es que el **usuario** va a ingresar un número y deberemos guardarlo para realizar las tareas correspondientes.



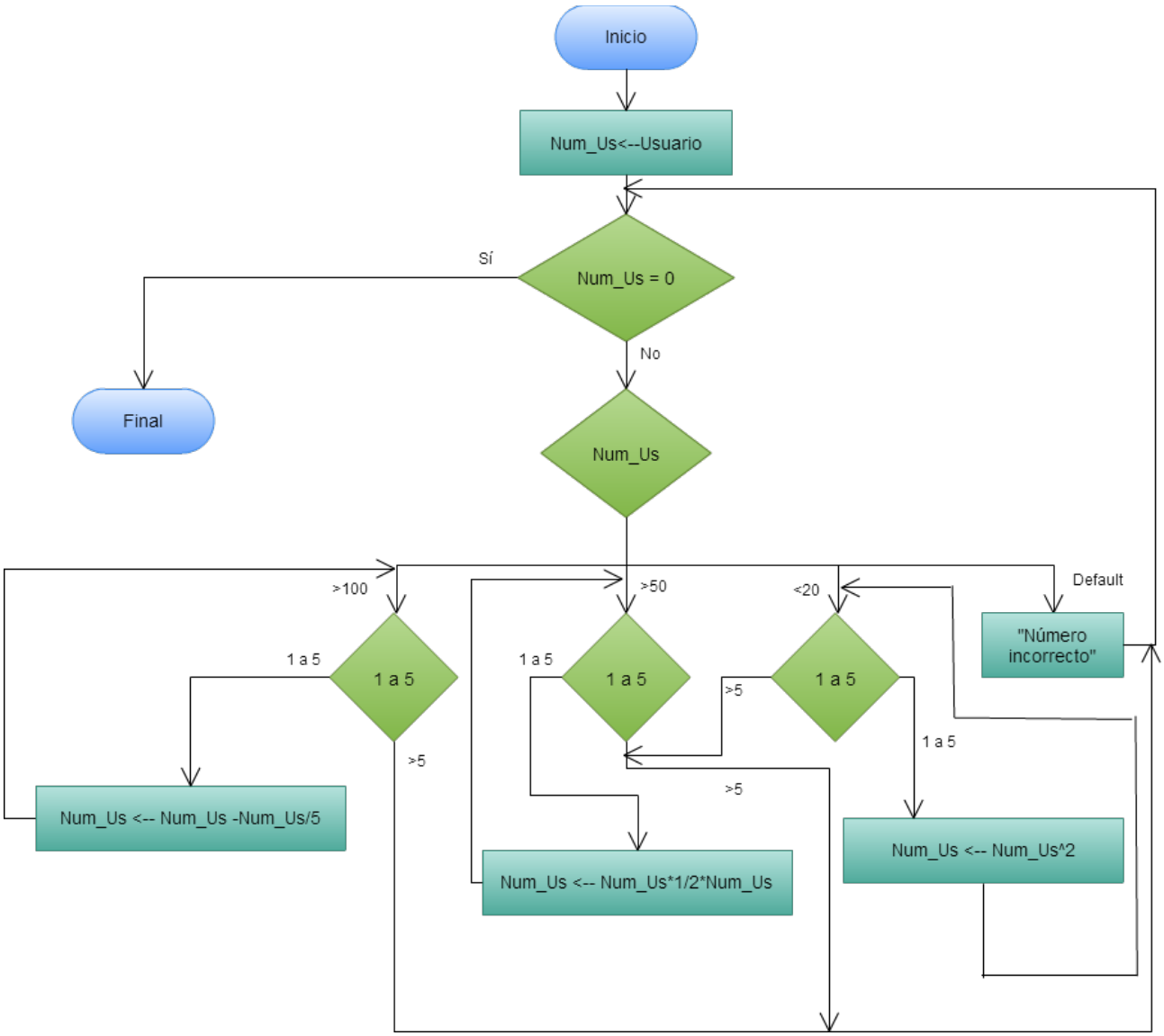
Verán que en vez de usar un círculo para el inicio, usé algo así como un cilindro con puntas redondeadas. Ésto se hace cuando tenemos una palabra y no una letra como indiqué al principio:) háganlo de la forma que más **cómoda** les quede.
Ahora, por lo que sabemos, si el número que ingresó es 0, el programa termina. Sino, sigue haciendo normalmente su tarea. Así que creo que aquí iría un "**while**".



Bien, tenemos lo suficiente. Ahora, debemos hacer un **switch** para identificar cuál es el valor guardado en “Num_Us”. Es indispensable poner primero (en este caso sería más a la izquierda), el “>100”, simplemente porque el switch también trabaja de forma secuencial, y si no es >100, podemos poner >50, sino deberíamos poner 2 condiciones en un mismo lugar del switch y es algo engorroso. Incluso hay algunos lenguajes que te limitan en ese sentido.



Luego ponemos un **for**, para realizar 5 veces la misma acción en los primeros tres. En el último, irá sólo un mensaje de error.



Buenísimo. **Terminamos** el diagrama de flujos del programa. Me quedo desorganizado pero es entendible. Podemos hacer también el pseudocódigo si quieren:). ¡Háganlo! Tomen la oportunidad y comiencen a pensar por ustedes mismos -y todas las reglas que debemos respetar-. Suerte alumnos:D

Ejercicio para que ustedes se rompan la cabeza:) :

Se debe leer la temperatura de un lugar cada una hora. El programa prende una calefacción si el ambiente es menor a 24ºC, y la refrigeración si es mayor a esta misma temperatura. Si es menor a 13ºC, el programa puede aumentar la temperatura de a 3ºC; si está entre 13ºC y 18ºC, el programa aumenta de a 2ºC; y si está entre 19ºC y 23ºC, aumenta de a 1.
En caso de que sea mayor, puede calefaccionar de a 2ºC o de a 1ºC.

Cualquier cosa pueden mandarme mail a: r0add@hotmail.com

Para donaciones, pueden hacerlo en bitcoin en la dirección siguiente:

1HqpPJbbWJ9H2hAZTmpXnVuoLKkP7RFSvw

Roadd.
