



*Logo de HDC :)*

*(Gracias a EdIL por el diseño)*

Bienvenidos a la trigésima quinta entrega de este curso:D. Espero que siga gustando como antes y que sigan tan entusiasmados como al principio. Yo por mi parte sigo aprendiendo muchísimas cosas pues me faltan tantas como a cualquiera y **me sigue apasionando**.

Hoy vamos a ver **diagrama de flujos**.

¿Qué es ésto? Es una forma **ilustrada**, con palabras dentro de **objetos conectados** entre sí, de ver los procesos que va a tener el software. Si entramos en otras áreas, el diagrama de flujos también sirve para otros procesos **distintos** al desarrollo de software. No es necesario encerrarlo en éso, pero nosotros lo estamos mirando desde esta **perspectiva**. Lo bueno e interesante, es que cualquiera que no esté en el campo, igualmente va a poder entender cómo queremos que pase.

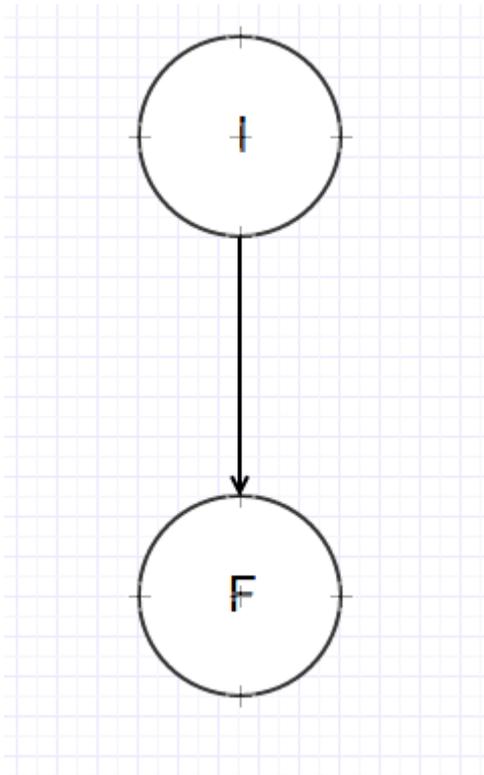
A diferencia del pseudocódigo, el diagrama de flujos es algo un poco más universal, donde ciertos **objetos** dentro de éste se **repiten**.



La “I” es de “Inicio” y la “F” es de “Final”.

Pero ahora ¿Cómo sé yo para dónde va? Porque más adelante tendremos bucles que irán a los costados o hacia arriba y mejor pensar ya que necesitamos darle una **dirección** a las conexiones. Para ésto, una **tercer regla: al final de las conexiones se indica una flecha con la dirección correspondiente.**

Miremos en el diagrama como quedaría ahora.



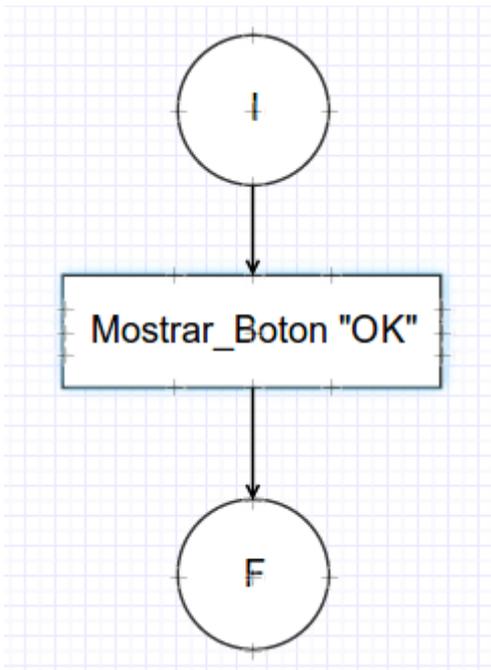
Bueno, pero nosotros queremos que además de iniciar y terminar, tengamos un proceso más extenso. Tengamos como ejemplo los pseudocódigos de la clase anterior. Cito:

**Empieza\_Programa**

**Mostrar\_botón “OK”**

**Termina\_Programa**

Cada comando, en este caso, llevaría su correspondiente símbolo. **Para los comandos simples o acciones de programa, utilizaremos un rectángulo.** Por lo que entonces sería algo así:



Muy bien, es casi un pseudocódigo. Pero ahora vamos a aumentar la cantidad de bloques. Para esto necesitamos un programa más grande aún. Intentemos con el que era más largo de la clase anterior. Cito:

**Empieza\_Programa**

**Num\_AI** ← **Generar\_Número\_Aleatorio**

**[PDR]**

**Num\_Us** ← **Usuario\_Ingresas\_Número**

**¿Num\_AI = Num\_Us?**

**Si: Mostrar\_Mensaje “Le has dado al número”**

**No: Empieza**

**¿Num\_AI > Num\_Us?**

**Si: Mostrar\_Mensaje “El número es más alto”**

**No: Mostrar\_Mensaje “El número es más bajo”**

**Volver [PDR]**

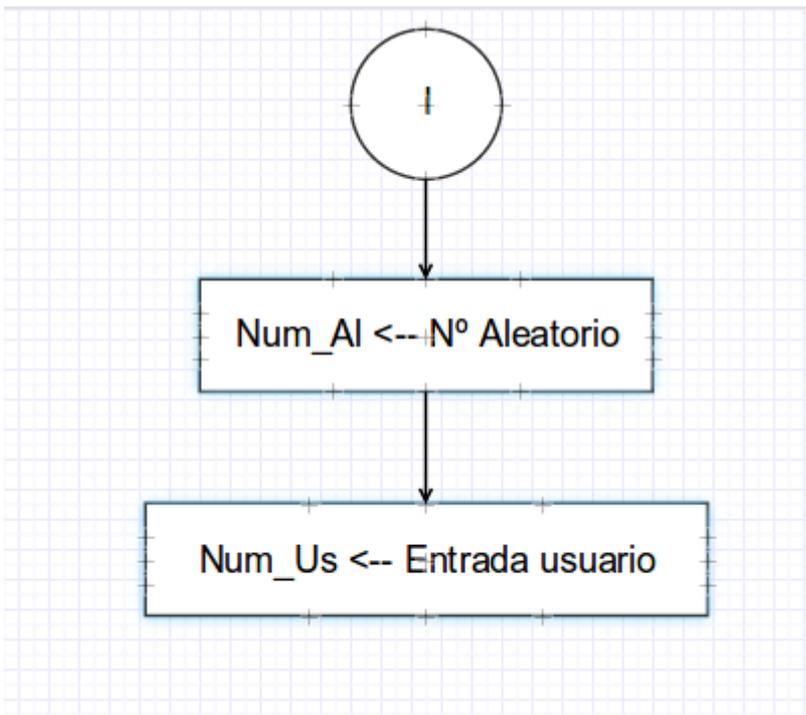
**Termina**

**Termina\_Programa**

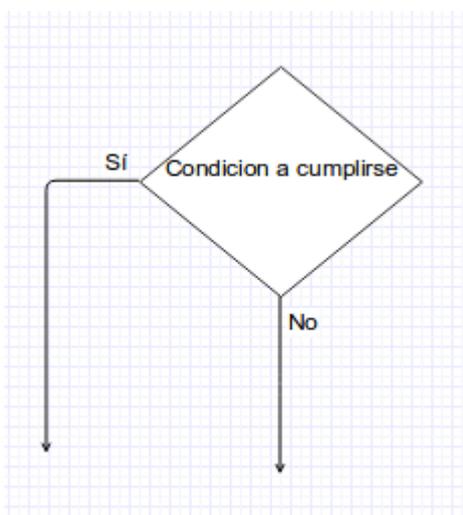
Veamos las cosas que necesitamos agregar. Primero, **guardamos unos valores con ciertos nombres**. Para esto, que **también es un comando o acción de programa, necesitamos colocar un cuadrado**.

Num\_AI <-- N° Aleatorio

La etiqueta no la necesitamos ahora, porque más adelante lo haremos de otra forma. Entonces **el diagrama empezaría así:**

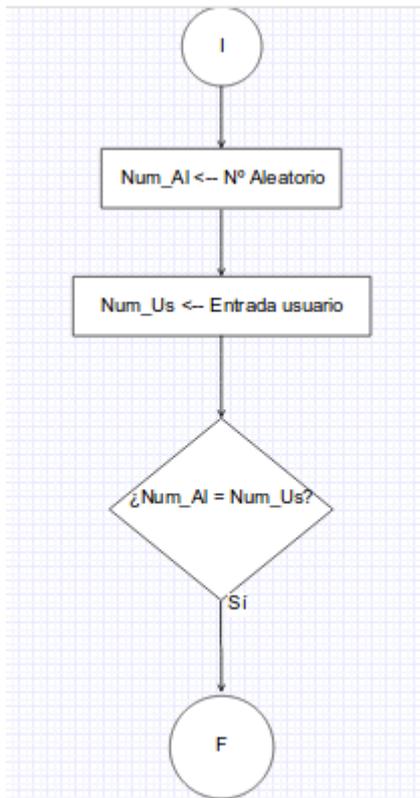


Muy bien. Ahora llega el momento de la pregunta y la **toma de una decisión**, dependiendo de la respuesta. Para esto haremos un **rombo donde dentro ponemos la pregunta y según la respuesta, tomará un camino u otro**. Cerca de cada línea (serán 2 y solo 2), se deberá colocar qué **condición necesita cumplirse para tomar esa trayectoria**. Veámoslo gráficamente:

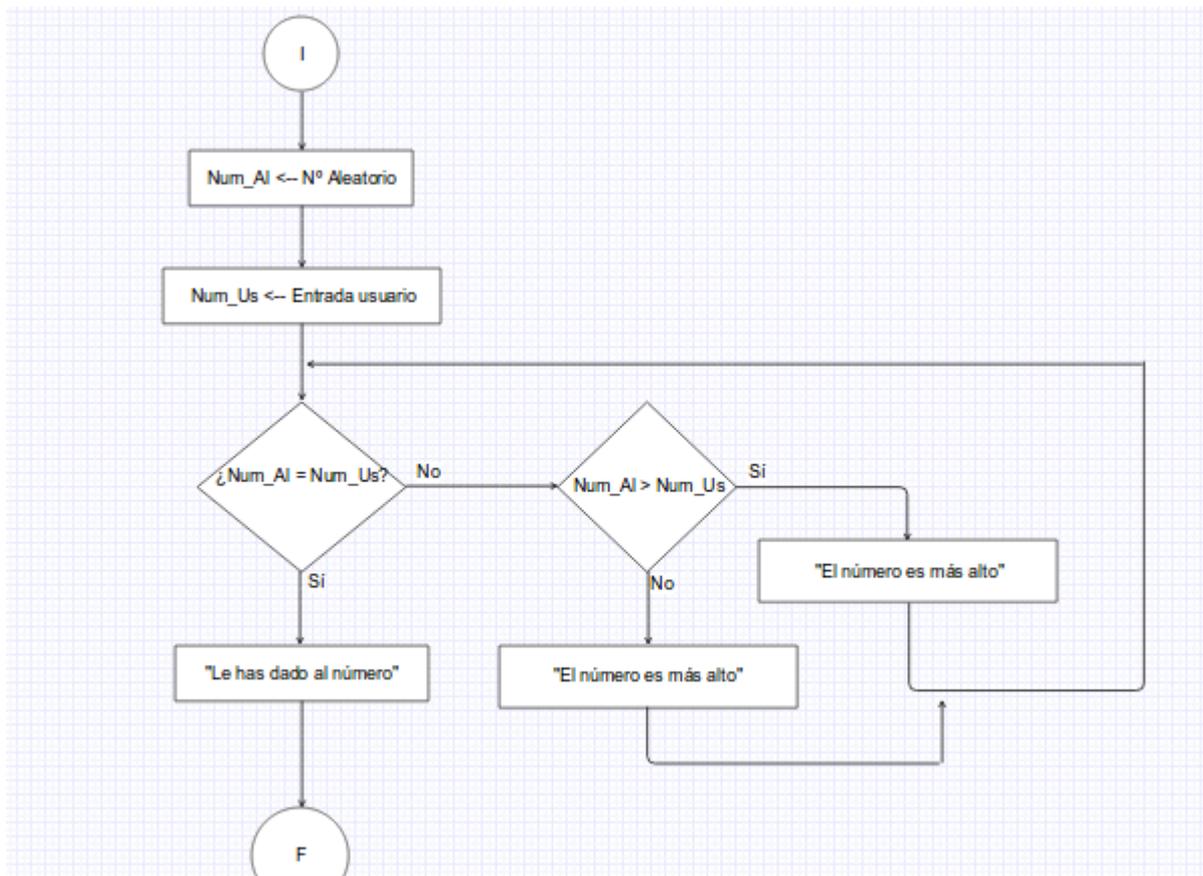


Si lo pegamos a lo que teníamos anteriormente, podemos ver que cuando llegamos al sí, ya **termina**. Hay que intentar de que **el final del programa esté lo más al sur de la hoja posible**. Esto

no es una norma ni nada, pero es lo más ordenado y así nuestra atención sobre el final del programa está siempre en el mismo lugar. Así también como si estamos buscando el principio, lo encontramos arriba de todo.



Y para terminar el diagrama, **sigamos hasta terminarlo todo**. Vemos que si va por el “volver” que teníamos en el pseudocódigo, aquí lo hacemos con una flecha que va hasta el lugar que queremos, pero siempre indicando que llega a otra conexión y no a un proceso. **Un tip: no es ordenado que de una flecha, salgan otras. Sólo está bien que entren más flechas.** Además está bien que a cada flecha sólo le llegue una más. Cuando llegue el momento verán como arreglamos los ejercicios para que podamos hacerlo. El diagrama completo:



**EDIT:** Hay un error en la imagen (en la segunda decisión "Num\_AI>Num\_Us"), cuando va al No, debería decir "El número es más bajo". Estos errores son por vago y usar demasiado el copy past xD.

Muy bien, ya les demostré como pensar y realizar un programa mediante las herramientas que disponemos. **Ahora, ejercicios donde deberán hacer tanto el pseudocódigo como el diagrama de flujos de los mismos.** Es vital saber que hay **infinitas** maneras distintas de realizar un programa y que ande de la misma manera. Por eso es que no existe una sola forma correcta.

**Usen el ingenio y su forma personalizada de resolución, porque acá todo es posible:)**

**Ejercicios:**

-Se necesita un programa que una vez se prenda, analiza la temperatura de una oficina. Si ésta es mayor a 25°, se prende un aire acondicionado. Si es menor, se apaga.

-Se quiere que una persona ingrese un número, y el programa indique, cuando sea par, si el número es mayor o menor a 200. Si el número es 0, el programa termina.

-Se quiere contar de 1 a 8 automáticamente, y en números pares se debe decir que son divisibles por 2.

-----  
Pueden seguirme en Twitter: @RoaddHDC

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

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

**1HqpPJbbWJ9H2hAZTmpXnVuoLKkP7RFSvw**

**Roadd.**

-----

**Este tutorial puede ser copiado y/o compartido en cualquier lado siempre poniendo que es de mi autoría y de mis propios conocimientos.**