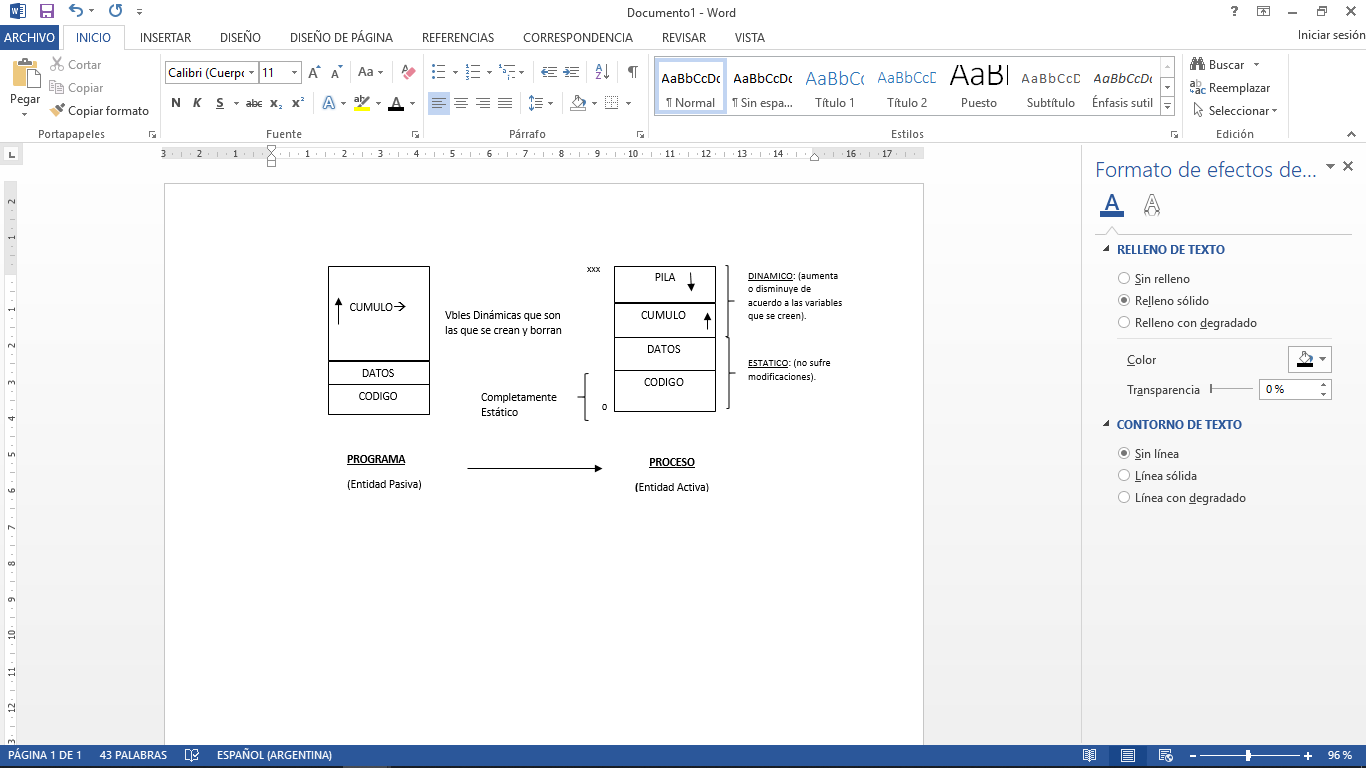


TEMA -2

RESUMEN DE CLASE tEORICA

**Sistemas Operativos**

**GESTION DE PROCESOS**

**PROCESO:** Programa en ejecución. Entidad ACTIVA.

**ENTIDAD PASIVA:** porque en realidad no ejecuta nada, no esta en su fase de proceso.

**PROGRAMA:** ENTIDAD PASIVA.

En un programa pueden haber variables GLOBALES y variables LOCALES.

LAS GLOBALES como se declaran y viven siempre en el código tienen una área de DATOS FIJA. Las LOCALES existen cuando se ejecuta un programa, por lo que podemos decir que no tienen un área fija.

**CUMULO**: variables que se van a crear en forma dinámica. Variables Dinámicas que son las que se crean y se borran. Crece hacia arriba.

**DATOS**: son variables ESTATICAS que se crean y existen siempre.

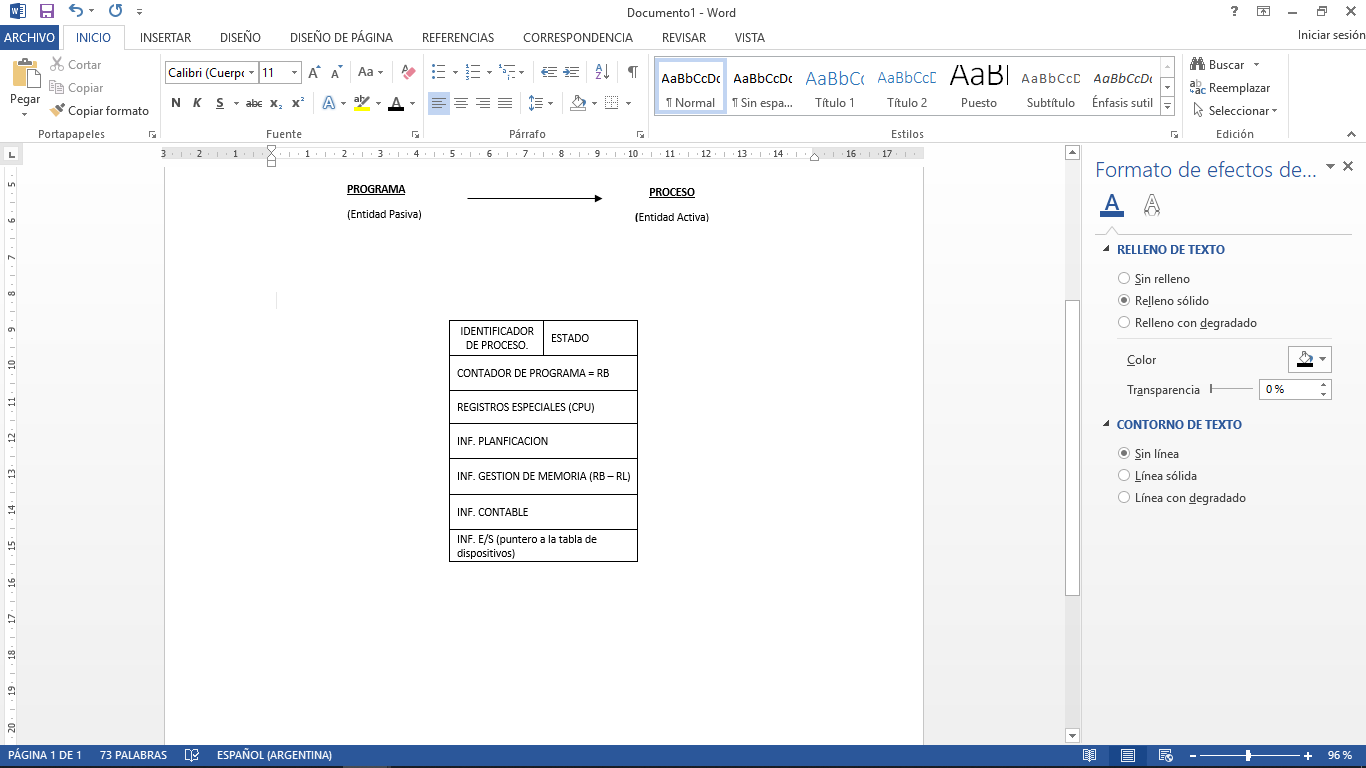
**PROCESO**: ENTIDAD ACTIVA

El programa ya esta en memoria y ejecutándose.

Cada PROCESO a parte de ser creado tiene un identificador (puede ser un número). El proceso cuando se crea y mientras viva en el sistema va a tener distintos estados.

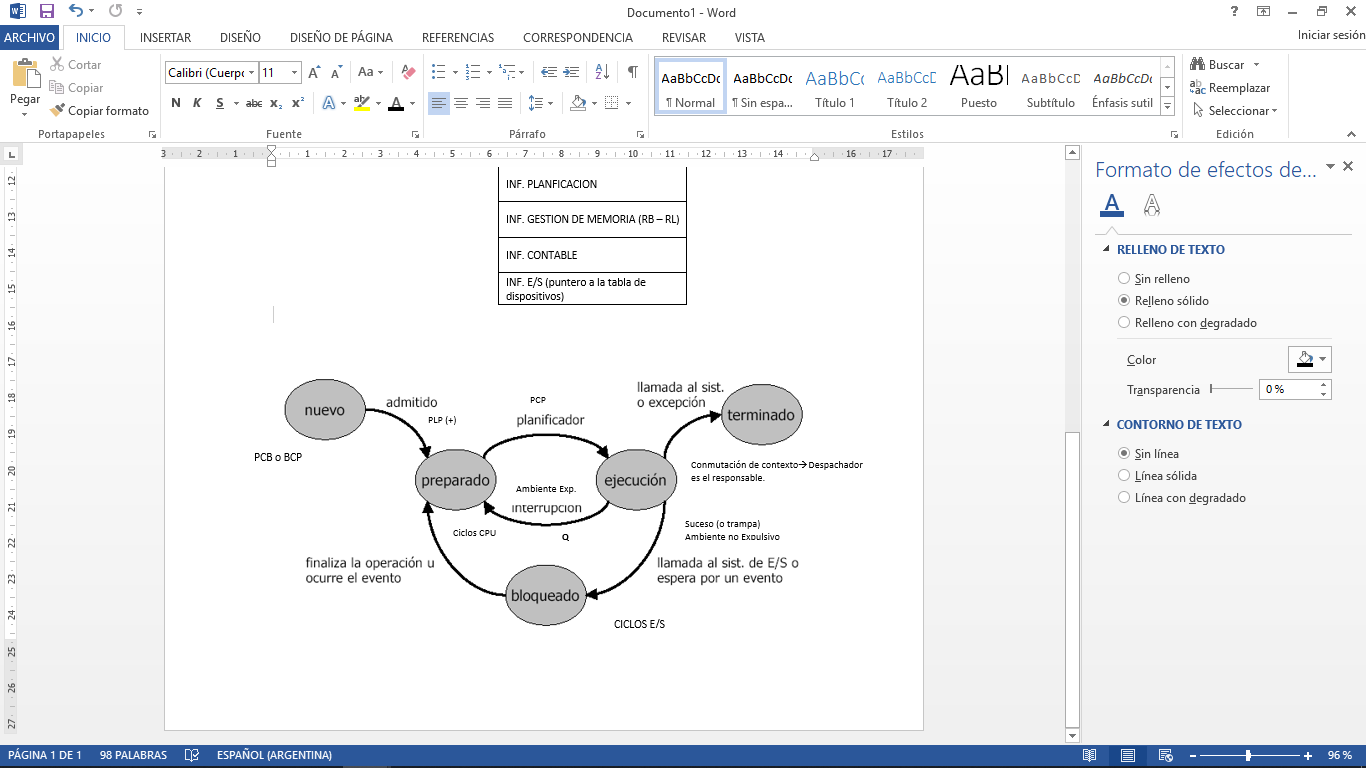
**BCP o PCB**: **Bloque de Control de Proceso** (Tenemos la IMAGEN DEL MICROPROCESADOR).

Es la estructura principal de cualquier proceso y contiene los datos necesarios para que el sistema lo reconozca como proceso. Esa estructura se crea dentro de la tabla de procesos activos del sistema, siendo esta una estructura de datos especial del S.O que se crea en memoria principal y donde contiene todos los procesos activos del sistema.



* IDENTIFICADOR DE PROCESO: es un número de la entradad en la tabla de procesos activos.
* ESTADO: en este caso en la cola de procesos nuevos.
* CONTADOR DE PROGRAMA: es el registro especial que indica en que dirección se esta ejecutando.
* INFORMACIÓN DE GESTION DE MEMORIA: cuanta memoria se necesita para ese proceso.
* INFORMACIÓN DE E/S: cuantos archivos necesita, si abrio algún archivo, estado de periféricos.

…. Mas datos almacenados.

**CICLO DE VIDA DE LOS 5 ESTADOS**

* PREPARADO o LISTO
* ESPERA o BLOQUEADO

**PLP (Planificador a largo Plazo):** Es una rutina de GESTION DE MEMORIA, que automáticamente cuando va ese PCB mira la memoria principal para ver si tenemos esa cantidad de memoria disponible para ese proceso, sino existiera el proceso seguiria esperando.

Si consigue la memoria necesaria donde entra el proceso, ese proceso para al siguiente estado que se llama LISTO.

**LISTO:** En este estado el proceso ya tiene memoria asignada, ya tiene un recurso asignado a él. En este estado el Registro Base va a pasar a tener la primera dirección que ocupa ese proceso. En Listo estan todos los procesos listos para recibir el microprocesador y ejecutarse, por eso se dice que el ESTADO LISTO es una espera en un ESTADO DE COMPETENCIA, o sea los procesos estan en espera y compiten entre ellos para obtener el microprocesador, es decir no estan haciendo nada útil, simplemente estan esperando a que se le entregue el microprocesador para poder ejecutárse.

Cuando el microprocesador este libre va a ser seleccionado por algún proceso para pasar al siguiente estado, siendo EJECUTANDO.

**EJECUTANDO**: El proceso se encuentra dentro del microprocesador, no solo tiene memoria sino que también tiene el micro y el PROCESO se esta ejecutando. El encargado de pasar el proceso del estado Listo al Ejecutando, es el PCP (Planificador a Corto Plazo).

**PCP (Planificador a Corto Plazo)**: Es el planificador que más rápido tiene que funcionar para seleccionar el PROCESO que va a hacer el uso del MICROPROCESADOR. Es el proceso que más rápido debe ejecutarse, para elegir el proceso que esta en LISTO y mandarloal micro para que éste no tenga ejecuciones vacias. Para esto el PLANIFICADOR tiene que llamar a una rutina especial del microprocesador, llamada DESPACHIN o DESPACHADOR, que realiza la CONMUTACION DE CONTEXTO.

**CONMUTACION DE CONTEXTO** (Despachador)

Copiar la imagen de proceso que sale en su PCB y tomar el bloque de control de Proceso del nuevo proceso y cargarlo dentro del microprocesador. Es decir:

1. Copiar el contenido del Micro en el PCB del proceso que sale (sale del micro).
2. Copia el PCB del proceso que entra en el micro o CPU.

Es decir, que el micro tiene que saber qué es lo que estaba haciendo el proceso que está entrando, si no tiene PCB no se puede ejecutar ese proceso.

La CONMUTACION DE CONTEXTO es la que hace la intercalación de datos dentro del microprocesador, indudablemente esta conmutación se debe hacer en MODO SISTEMA, porque accede directamente a los datos del micro.

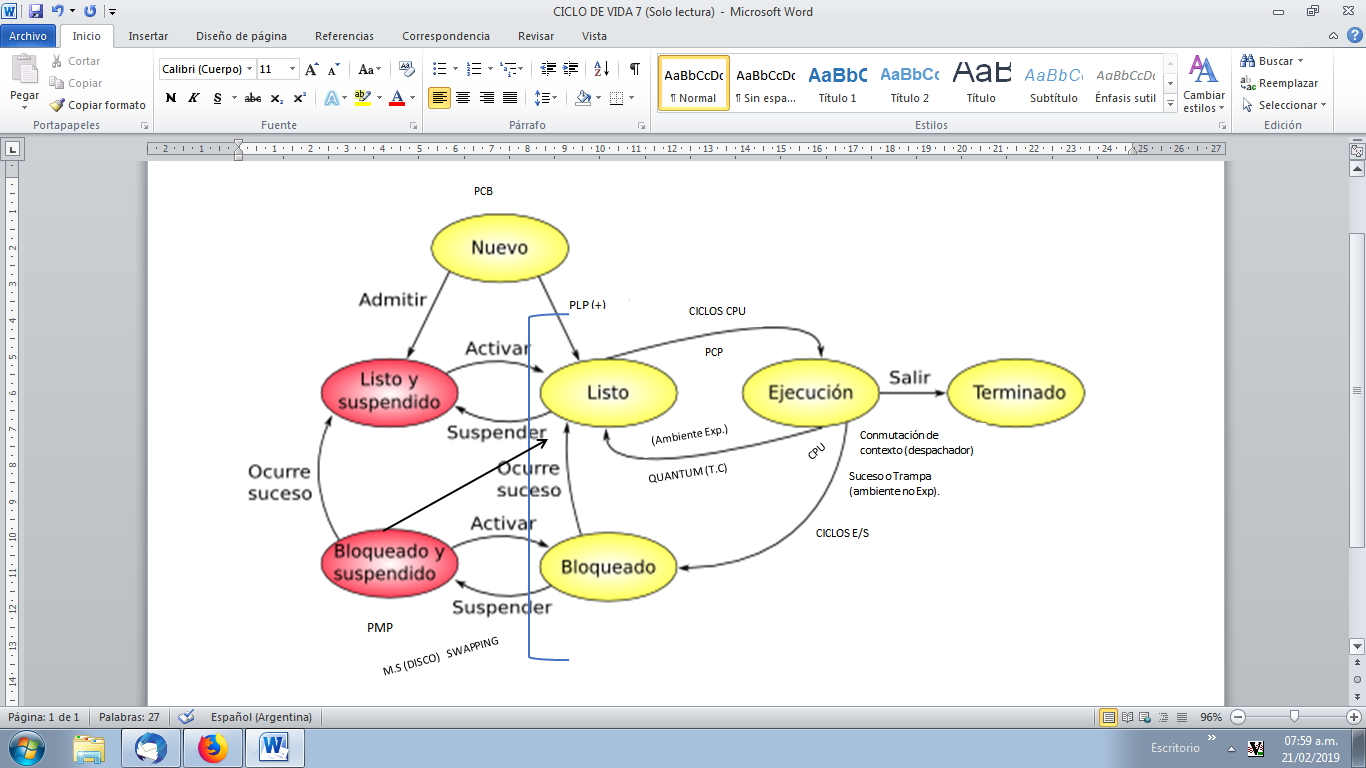
Una vez que se copió la imagen de ese proceso, este ya está apto para seguir ejecutándose con sus propios datos. Cuando está en EJECUTANDO el proceso; este puede salir por tres CONDICIONES del microprocesador:

* **TERMINADO**: es porque finalizo. Este estado es donde cada proceso libera todos los recursos que tiene asignados a él. En realidad, lo que hace es borrar el PCB y el proceso deja de existir. Libera el PCB para que lo utilice otro proceso.
* **EN ESPERA O BLOQUEADO**: Por algún suceso para el estado de ESPERA O BLOQUEADO, puede ser una E/S, una llamada al sistema; por ejemplo: no tengo memoria y como el proceso está solicitando memoria va a pasar a este estado de espera hasta que se le asigne memoria.

Cuando ocurra ese SUCESO o TRAMPA, ese proceso volverá a pasar al ESTADO LISTO que se le asigne nuevamente el microprocesador.

* **TIEMPO COMPARTIDO:** (QUANTUM) Cuando se le termino el tiempo de ejecución que tenía asignado, o sea cuando se expiró el tiempo, el proceso debe abandonar el estado EJECUTANDO y volver a LISTO.

En este caso, la CONMUTACION de CONTEXTO es mucho más importante, porque el proceso es interrumpido en el momento de su ejecución. Cuando un proceso sale voluntariamente, sale por un pedido al sistema, se dice que estamos en un AMBIENTE NO EXPULSIVO. Cuando salimos por tiempo compartido sale involuntariamente, entonces se dice que estamos en un AMBIENTE EXPULSIVO.

**CICLO DE VIDA DE 7 ESTADOS**

¿Por qué existe el CICLO DE VIDAD DE 7 ESTADOS?

Porque, así como existe el planificador de largo plazo y se dice que este incrementa el grado de multiprogramación, suma procesos a la memoria. Pero llega un momento que no hay más memorias y los procesos andan demasiados lentos.

Entonces, surge un nuevo planificado, que se llama PMP (PLANIFICADOR DE MEDIANO PLAZO), aumenta y disminuye el grado de programación o sea agrega o resta programas. El OBJETIVO de PMP es gestionar la memoria de manera tal que todos los procesos se puedan ejecutar mejor. Por ejemplo: los procesos que se encuentran en un estado de ESPERA, como no están haciendo nada útil en vez de esperar en memoria principal los bajamos al DISCO y que esperen en el DISCO liberando memoria que estaban utilizando. Podemos bajar el proceso en forma total o parcial, porque los procesos tienen un área que es estática, que nunca sufre modificaciones. Si no se necesita ninguna parte del proceso se lo baja en forma total al DISCO en un área que se llama ZONA DE INTERCAMBIO, y ese nuevo estado se llama BLOQUEADO SUSPENDIDO.

* **BLOQUEADO SUSPENDIDO**: al bajar, aunque sea solo el área de código, ya se está liberando memoria, eso va a hacer que los procesos están en EJECUTANDO, tengan más memoria para ejecutarse o en su defecto que los procesos que están en LISTO puedan ejecutarse.

Cuando los procesos estén listos para ejecutarse, pero no haya suficiente memoriam surge el nuevo estado:

* **LISTO SUSPENDIDO**: Cuando haya memoria los primeros que van a pasar a LISTO son los procesos que están en este estado, incluso antes de los nuevos procesos. Esto sucede porque estos procesos ya están en un grado de programación.
* **PMP** (Planificador a mediano Plazo): es un planificador que realiza SWAPPING que significa bajar los datos a la zona de intercambio y cuando se pueda volver a subirlos a memoria; siempre y cuando haya lugar. Por esto se dice que el PMP disminuye el grado de multiprogramación, porque baja los procesos. Recordar que el PCB nunca sale de memoria principal porque está dentro del S.O.

**ESTRUCTURAS DEL SISTEMA OPERATIVO**

* **Tabla de Procesos**: aquí se encuentran los punteros al PCB.
* **Tabla de Dispositivos**: donde se encolan en los PCB.
* **Tabla de Procesos en Espera** de ejecución, espera activa, se encolan los PCB.
* **Tabla de Procesos en Gestión de Memoria**: se encolan los PCB.

Todo está en memoria principal. El PCB es la base del Proceso, es la única forma de localizar al proceso que queremos ejecutar, es una entrada a la tabla del estado en el cual se encuentra. Si los PCB no están en memoria principal NO EXISTEN.

Los planificadores son rutinas de S.O que van a ser llamados a la memoria cuando se los necesite. El PCP es una rutina que tenemos en memoria constantemente porque se tiene que ejecutar rápido, al igual que el DESPACHADOR porque en todo momento va a estar realizando CONMUTACIONES DE CONTEXTO.

**OPERACIONES BASICAS DE PROCESOS**

* CREACIÓN DE PROCESO: es cuando se produce la creación del PCB y por lo tanto va a pasar al estado de Nuevo.
* ELIMINACION DE PROCESOS: significa que se va a abortar un proceso o matar un proceso.

Un proceso puede crear a su vez otros procesos, de esa forma obtenemos la relación PADRES e HIJOS. El proceso que crea otro proceso se llama PROCESO PADRE y el proceso que es creado por otro proceso mayor se llama HIJO.

Es una estructura de árbol, ningún hijo puede existir sin padre. Si se borra al proceso padre, este proceso deberá mandar a borrar o cancelar a los procesos hijo. Es decir, EN EL SISTEMA NO EXISTEN HIJOS SIN PADRE.

Podemos decir que existen dos tipos de procesos:

* **PROCESO INDEPENDIENTE**: que es aquel proceso que no depende de ningún otro, por lo tanto, no es hijo ni padre. Solo depende de los recursos del sistema.
* **PROCESOS COOPERATIVOS**: son procesos que tienen comunicación con otros procesos o dependen de otros. Como por ejemplo la relación padre e hijo.

Se llaman cooperativos porque distribuyen su trabajo para realizar mayor trabajo a mayor velocidad.

**CLASE ESPECIAL DE PROCESOS HIJOS LLAMADA HILOS**

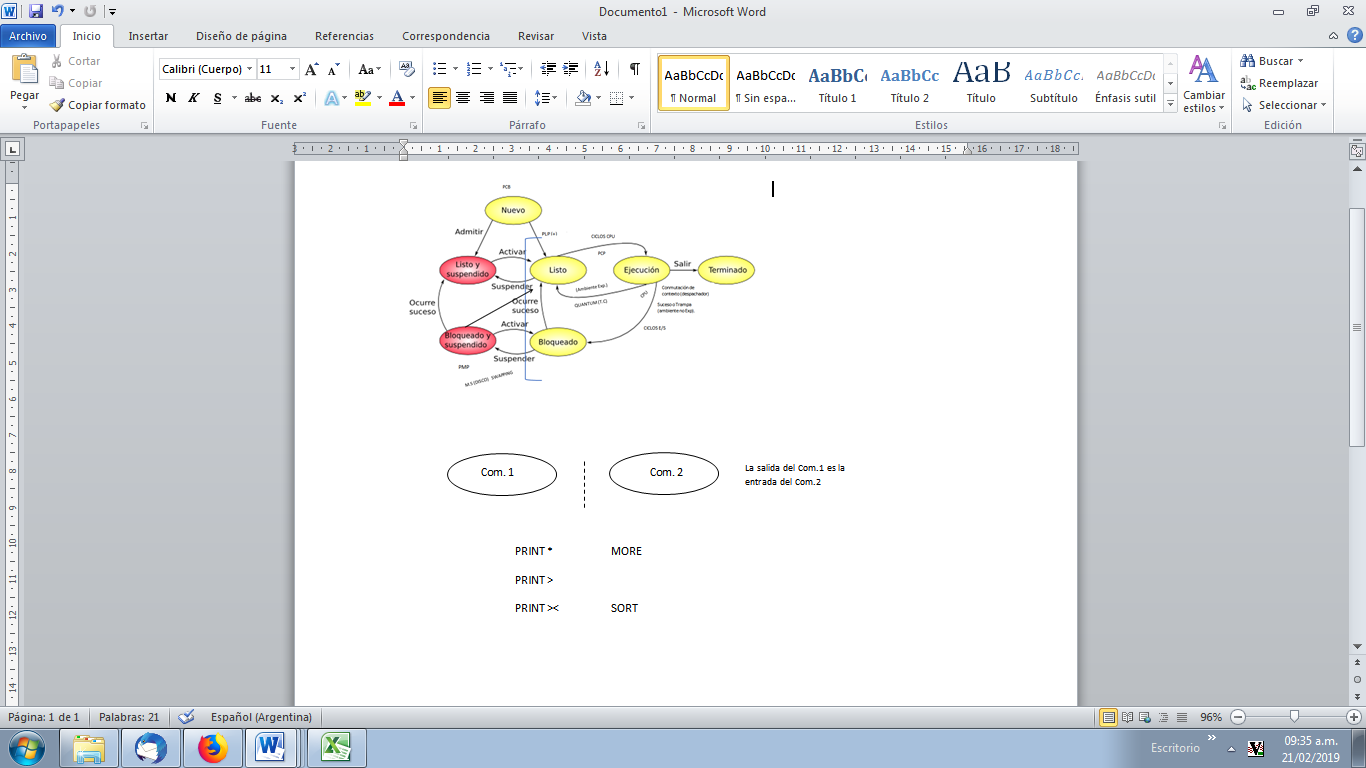
Los HILOS son procesos hijos que viven dentro del ambiente del padre, pero tienen una estructura muy especial, comparten los recursos del padre.

Los HILOS tienen su propio PCB mucho más pequeño donde solo contienen la pila, el código y una pequeña porción de datos, en realidad son los registros generales necesarios. Todo lo demás está dentro del ambiente padre. Indudablemente los HILOS son procesos altamente COOPERATIVOS, porque están compartiendo la memoria del padre.

Como trabajan dentro del mismo ambiente que el padre, los HILOS necesitan sincronización para su ejecución.

Todo PROCESO COOPERATIVO necesita una forma de Comunicación, estas formas son:

* POR MEMORIA COMPARTIDA: existe un área de memoria que comparten los procesos y por lo tanto, necesitan de la SINCRONIZACION para no perder datos que puede ser por:
* **POR TUBERIAS**: son comandos especiales del S.O que permite pasar los datos de un proceso hacia otro.



* **POR MEDIO DE MENSAJES**: lo hace a través de la red. El paso de mensajes básicamente tiene dos instrucciones:

SEND {enviar}

RECEIVE {recibir}

El Paso de Mensajes ya está sincronizada por sí sola, el paso de mensaje puede ser:

* POR COMUNICACIÓN DIRECTA o INDIRECTA
* POR COMUNICACIÓN SINCRONICA o ASINCRONICA
* ALMACENAMIENTO EN BUFFER EXPLICITO o AUTOMATICO

**La Comunicación Directa** es cuando el que envía, nombra al que receptor. Es Bloqueante o Sincrónica, es decir se envía un mensaje y el emisor no puede hacer otra cosa hasta que el receptor conteste. Comunicación Directa y Sincrónica, por ejemplo, chatear.

**Comunicación Asincrónica o Indirecta:** es cuando usamos algún tipo de Buzón, eso sería el correo electrónico. Es Asincrónica porque al enviar un mensaje no se sabe con certeza cuando el receptor leerá dicho mensaje.

**Con almacenamiento de Buffer:** existen tres tipos de buffer.

* + - **El Buffer Cero:** Capacidad cero sería cuando tenemos comunicación directa y sincrónica, es decir no tenemos un buzón, no tenemos un buffer o no vemos o no nos comunicamos.
    - **Buffer Limitado:** existe un lugar de memoria para almacenar, por eso son comunicaciones indirectas, asincrónicas y son no bloqueantes, porque yo envió el mensaje y puedo continuar haciendo otra cosa, no tengo que esperar a que me contesten. Buffer Limitado quiere decir que tienen un espacio limitado de almacenamiento, o sea que si se envía un mensaje y el buffer se encuentra lleno, rechazar la conexión porque no tiene donde guardar.
    - **Buffer Ilimitado:** quiere decir que se puede enviar todos los mensajes que quiera.

**HILOS**

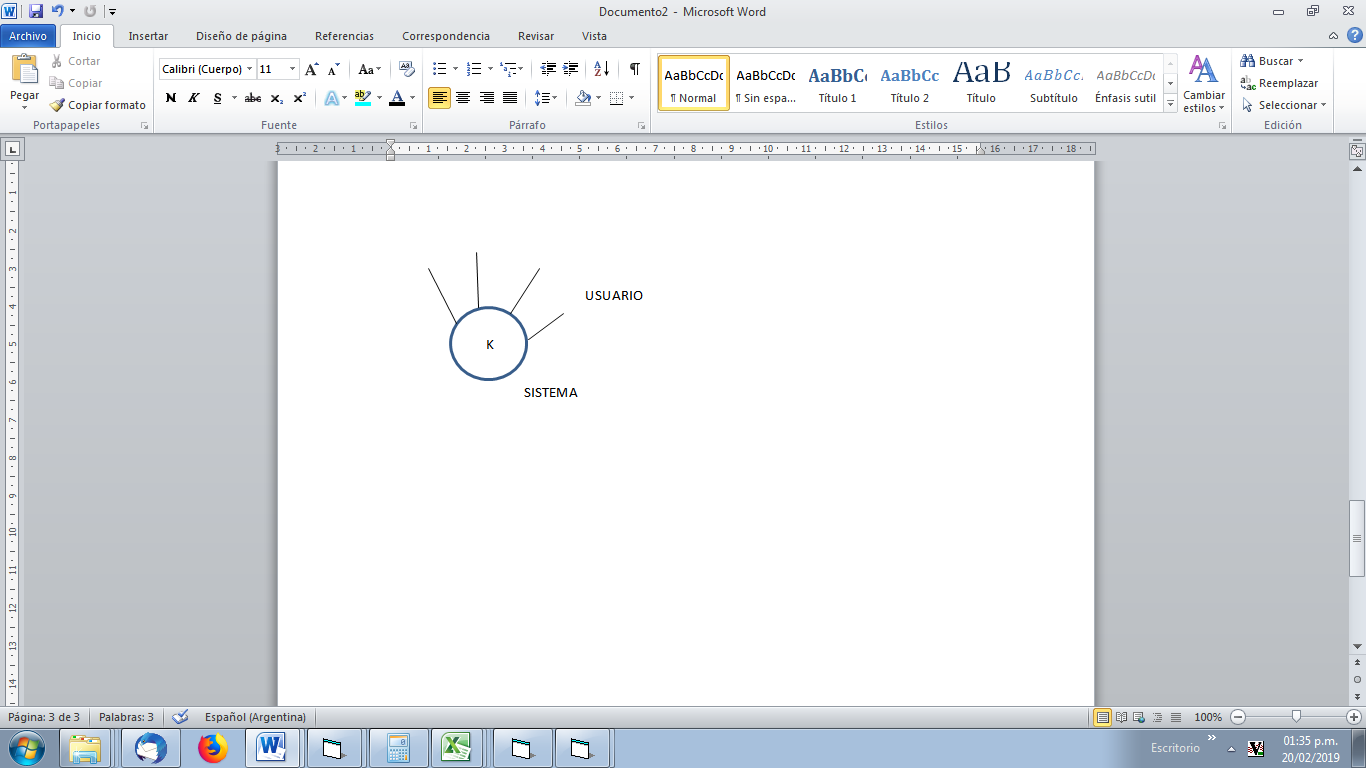
Un HILO es un proceso ligero, porque se dice que un hilo es una UNIDAD BASICA DE EJECUCION. Es una única tarea pequeña y liviana.

**Ventaja**

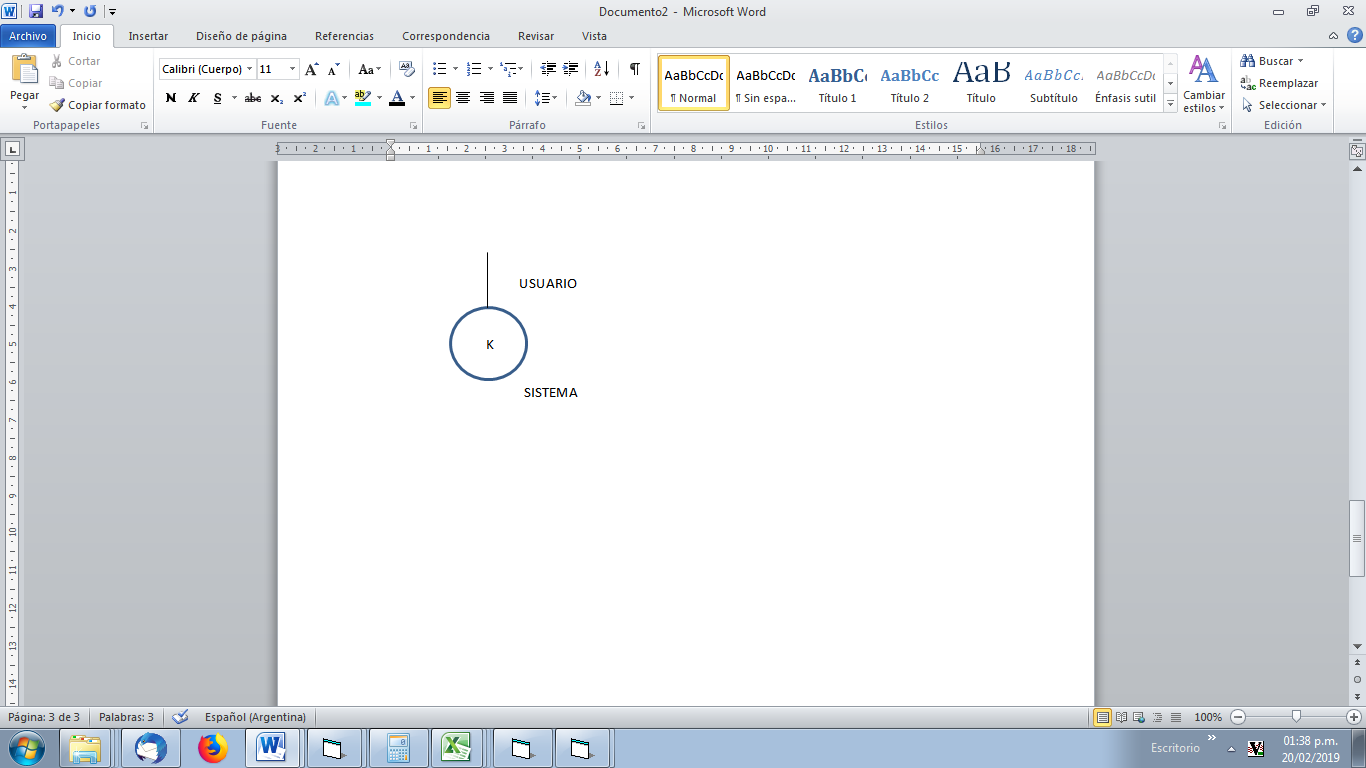
La ventaja de los hilos es que como su Conmutación de Contexto es más rápida porque vive dentro del ambiente del padre y su PCB es más pequeño. Podemos decir que tenemos dos tipos de hilos:

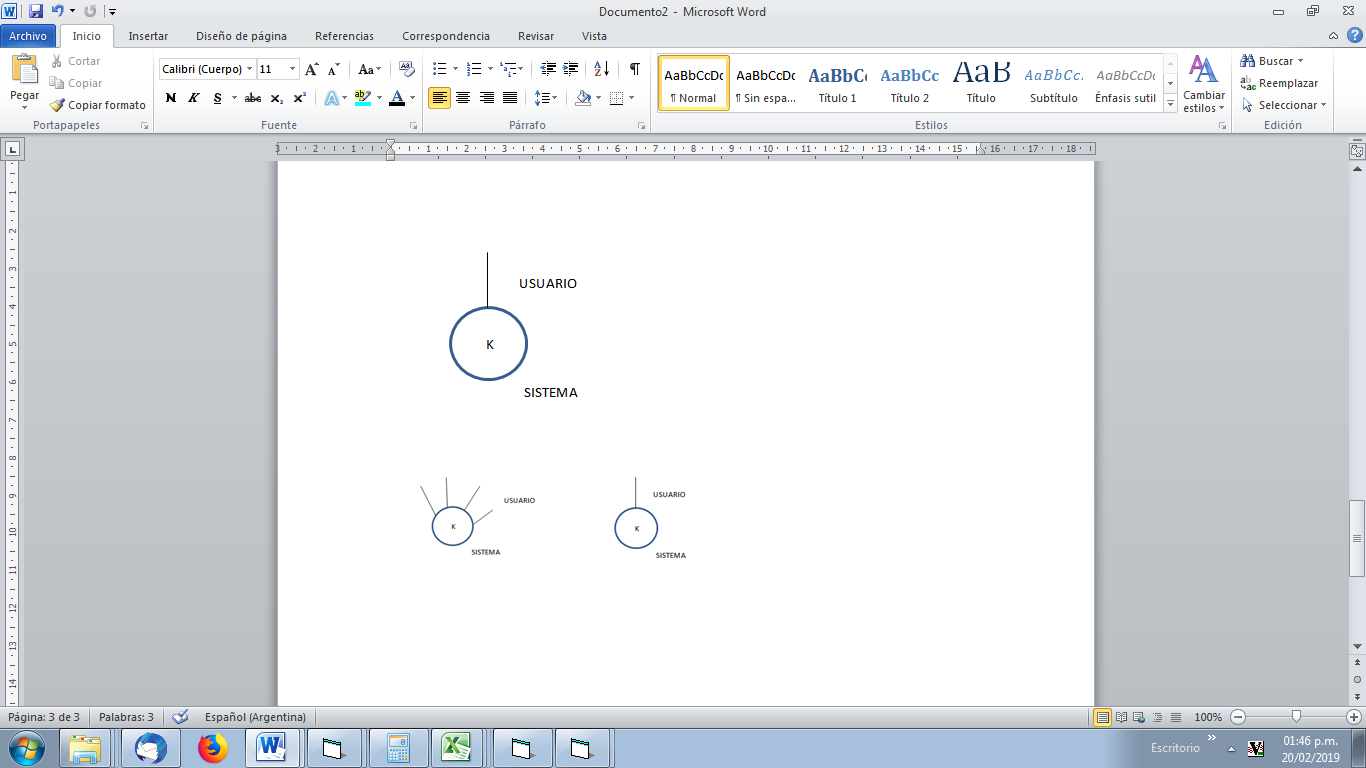
* Los Hilos a nivel Usuario
* Los Hilos a nivel Sistema

**ESTRUCTURAS DE HILOS**

MODELO MUCHOS A UNO: tenemos muchos hilos a nivel Usuario por un solo hilo a nivel Sistema, eso quiere decir que los hilos están dentro del PCB del padre, pero se ejecutan por un solo hilo del sistema. Indudablemente estos hilos no pueden ejecutarse en forma paralela.

* MODELO UNO A UNO: que por un hilo de usuario tenemos un hilo del sistema. Por lo tanto tenemos paralelismo.



* MODELO MUCHO A MUCHOS: lo que este modelo hace es que de muchos del usuario podemos tener a muchos del sistema (a uno). La idea es que puedan ser tanto secuenciales como paralelos.

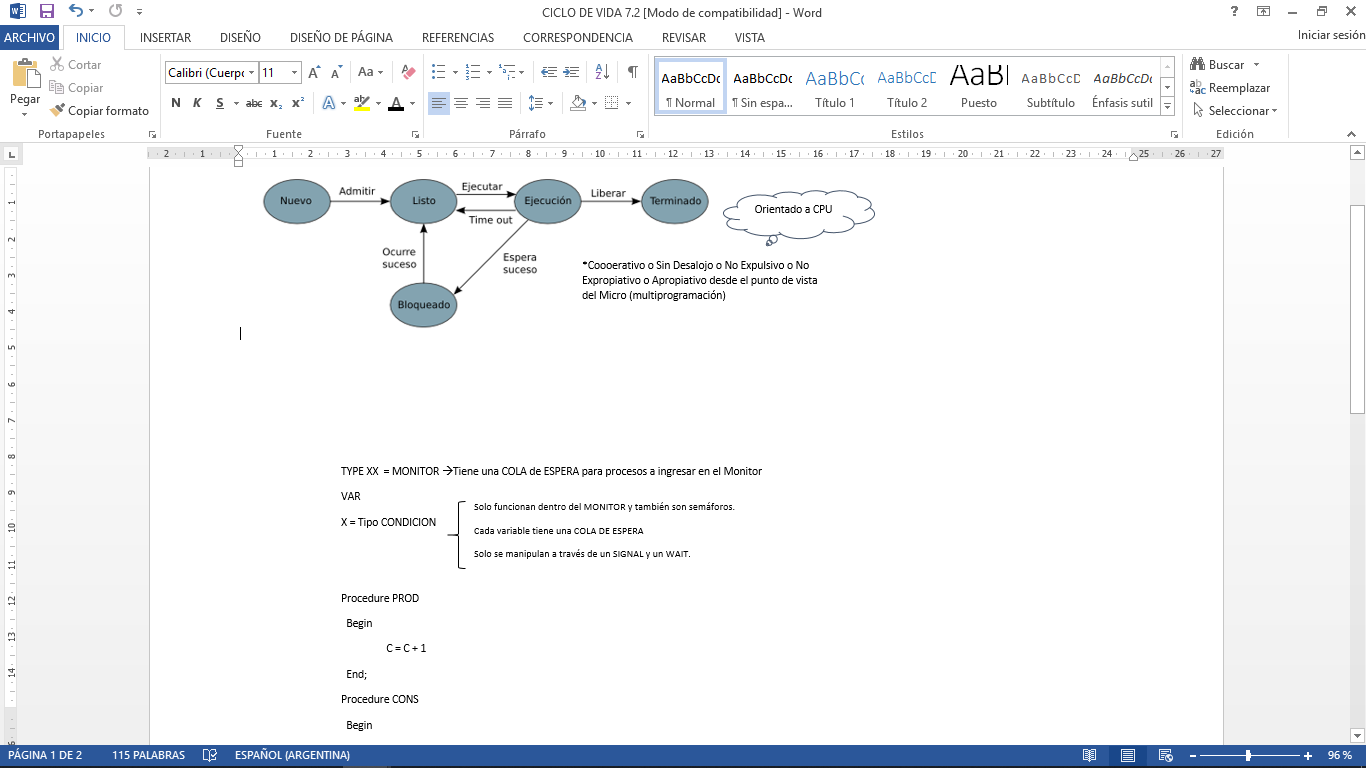
**BIBLIOTECA DE HILOS o HEBRAS**

* Posix Threads (biblioteca universal y estándar de los S.O): se puede usar a nivel Usuario y a nivel Sistema.
* Win 32 🡪Nivel Sistema
* Java 🡪Nivel Usuario

**PLANIFICACION DEL CPU**

Estados Básicos Tiempo Compartido 🡪QUANTUM de TPO

\*INTERACTIVO:

* Con desalojo
* Expulsivo
* Expropiativo
* No Apropiativo

**LISTO**:

Esta cargado en memoria principal.

Esta en un estado de competencia, es decir los procesos que estan en la cola de LISTO, no estan haciendo nada útil, porque estan en un estado de espera por el microprocesador.

¿Quién selecciona el proceso que esta en cola para ejecutarse?

El que selecciona es el PCP, este planificador tiene que ser el más rápido, porque el objetivo es elegir el proceso y rápidamente llevarloa un estado de EJECUCIÓN.

¿Cuál es el objetivo del PLP?

El objetivo es conseguir memoria para hacer la transferencia.

**El PCP** trabaja conjuntamente con el DESPACHADOR siendo su objetivo realizar el CAMBIO DE CONTEXTO que consiste en guardar el contenido del microprocesador del proceso que sale en el PCP o sea resguardar los datos y cargar la imagen del proceso que entra en el micro para que el proceso pueda seguir ejecutandose.

**LATENCIA DE DESPACHO:** es el tiempo de ejecución del despachador en el que se demora en hacer las dos transferencias. Tiene que ser un teimpo lo más corto posible, o sea quiere decir que el despachador tiene que ejecutar esa transferencia de datos lo más rápido posible y sin interrupción, para que pueda continuar ejecutándose el nuevo proceso.

Esa LATENCIA DE DESPACHO es muy importante de tener en cuenta cuando tenemos procesos INTERACTIVOS, o sea cuando tenemos Tiempo Compartido porque el tiempo que se le da a cada proceso tiene que ser mayor que el Tiempo de Latencia de Despacho.

**BLOQUEADO**: el estado bloqueado se da cuando el proceso que se esta ejecutando pide una E/S, la ejecución de un procesoque el proceso sale por Hijo o dispara Hilos. Es decir que el proceso sale por su voluntad del microprocesador. Por eso se dice que cuando sale por aquí, ese proceso es COOPERATIVO o es un proceso SIN DESALOJO o NO EXPULSIVO o NO EXPROPIATIVO o APROPIATIVO desde el punto de vista del microprocesador. Porque desde el punto de vista del microprocesador, quiere decir que es sin desalojo por lo que no se esta expulsando al proceso. Se lleva acabo la MULTIPROGRAMACION, es decir que el proceso sale en forma voluntaria y se aprovechan los tiempo muertos de E/S o los tiempos muertos de algún suceso.

**TIEMPO COMPARTIDO**: Cuando tenemos Tiempo Compartido tenemos que tener un QUANTUM de TIEMPO que sea mayor a la latencia de despacho. Quiere decir que estamos en un S.O que es INTERACTIVO porque los procesos entran y salen.

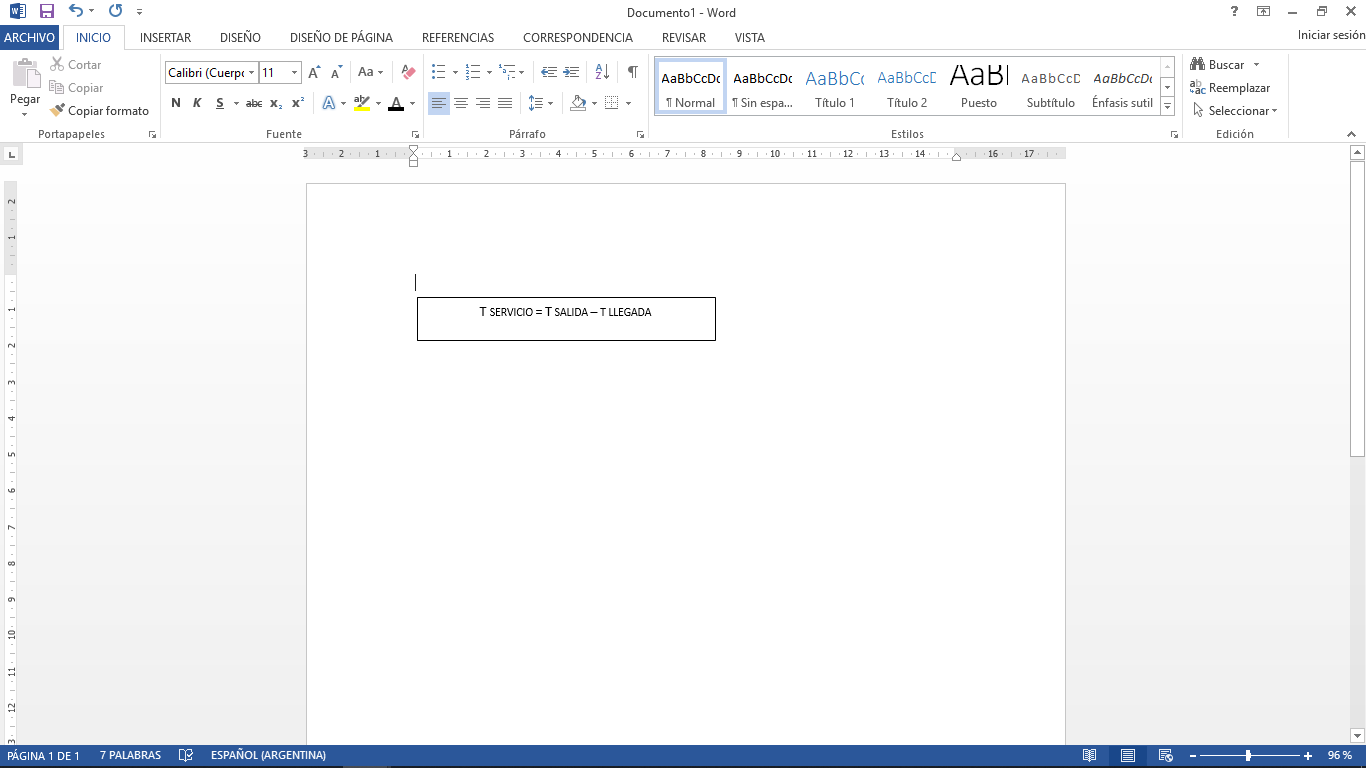
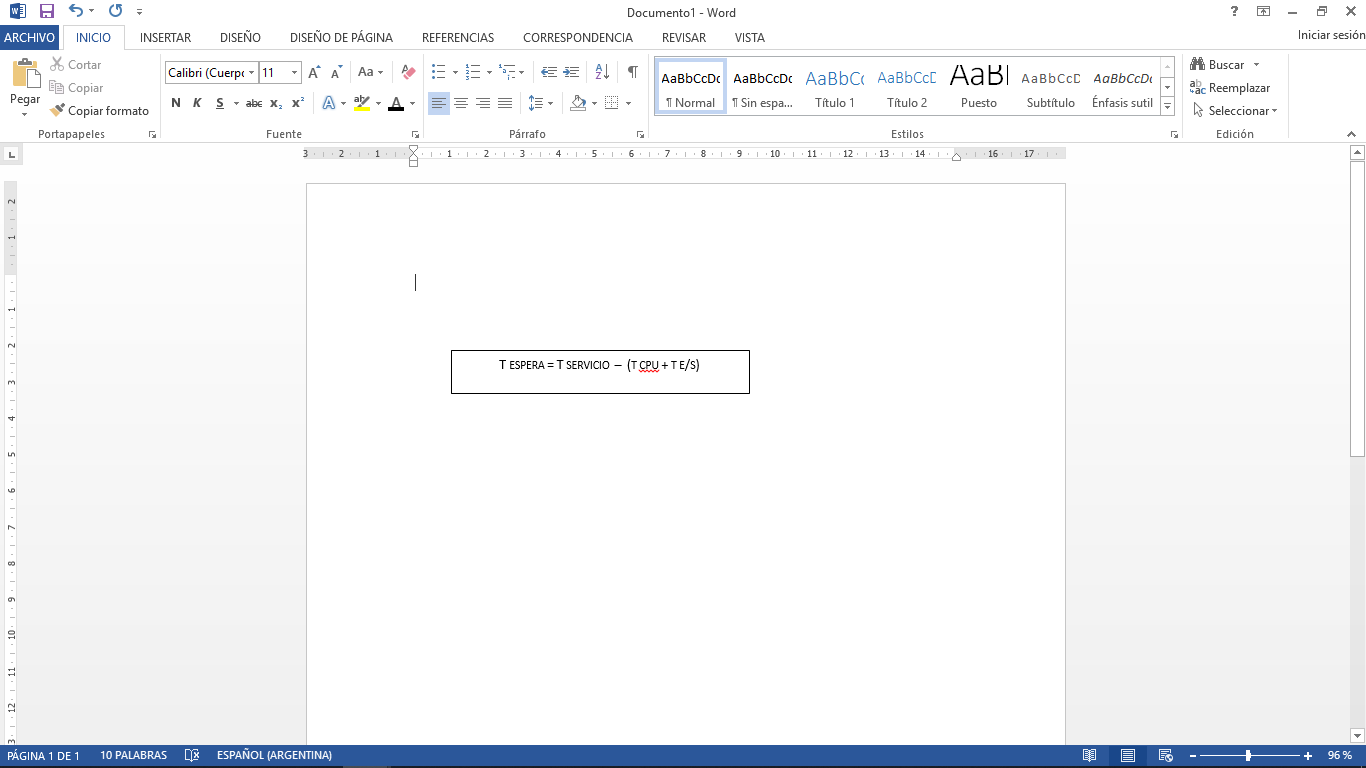
Cuando sale por aquí, quiere decir que el proceso es DESALOJADO del microprocesador, lo que significa que sale en contra de su voluntad. Entonces se dice que es: CON DESALOJO, EXPULSIVO, ENPROPIATIVO, NO APROPIATIVO.

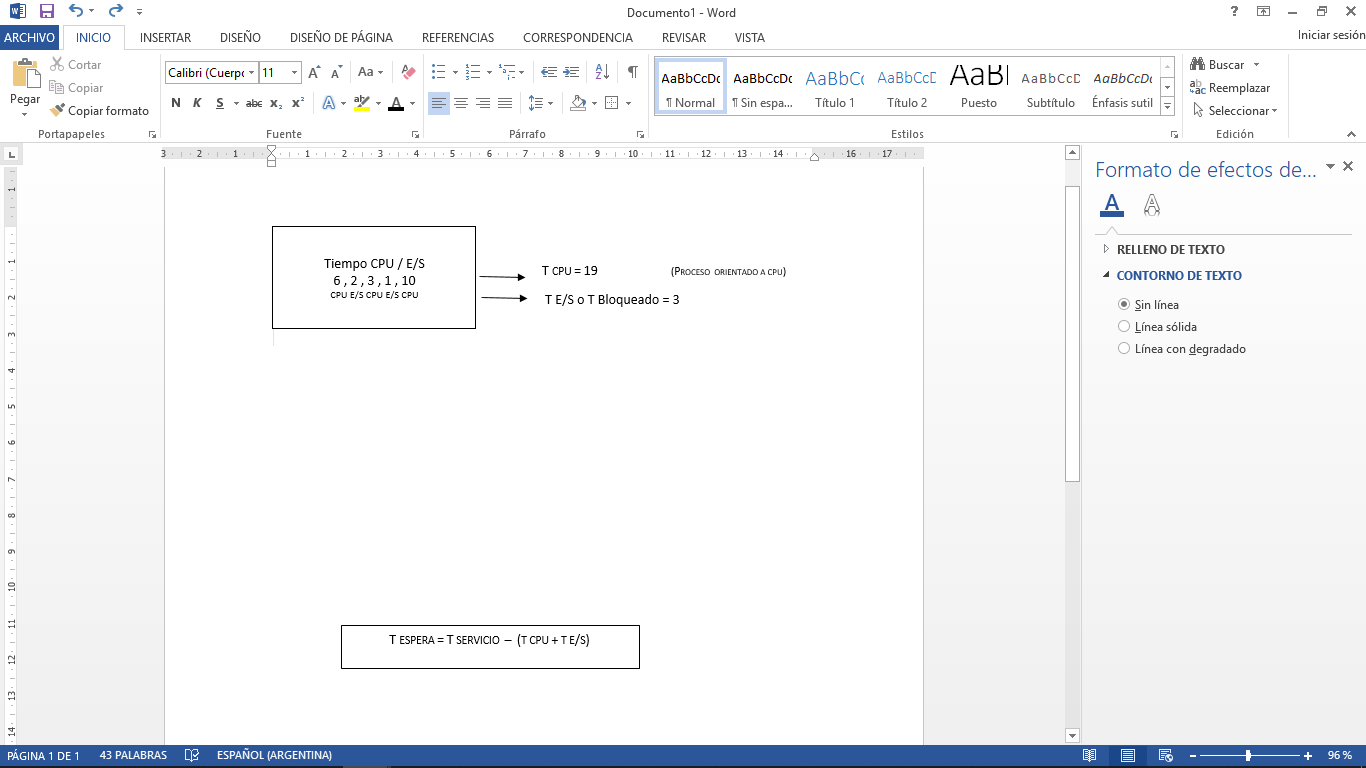
**RAFAGAS DE CPU**: cuando un proceso tiene mucho tiempo de microprocesador, por ejemplo un proceso puramente matemático generalmente tiene mucho tiempo de ejecución en el microprocesador, porque tiene mucho cálculo matemático y pocas E/S. Por eso se dice que este tipo de proceso esta fuertemente orientado a RAFAGAS DE CPU.

**RAFAGAS DE E/S**: por ejemplo un proceso de liquidación de sueldo o cualquier proceso administrativo, normalmente tienen cortas ráfagas de CPU y largos tiempos en BLOQUEADO. Por eso un proceso de ese tipo se dice que esta fuertemente orientado a RAFAGAS DE E/S.

De los tres estados Básicos de un proceso, los dos estados útiles son: el EJECUTANDO porque tiene rafagas de CPU y el BLOQUEADO porque esta bloqueado en espera que otro le de el resultado, en este caso podemos decir una operación de E/S.

**CRITERIOS QUE TIENE QUE USAR EL PCP**

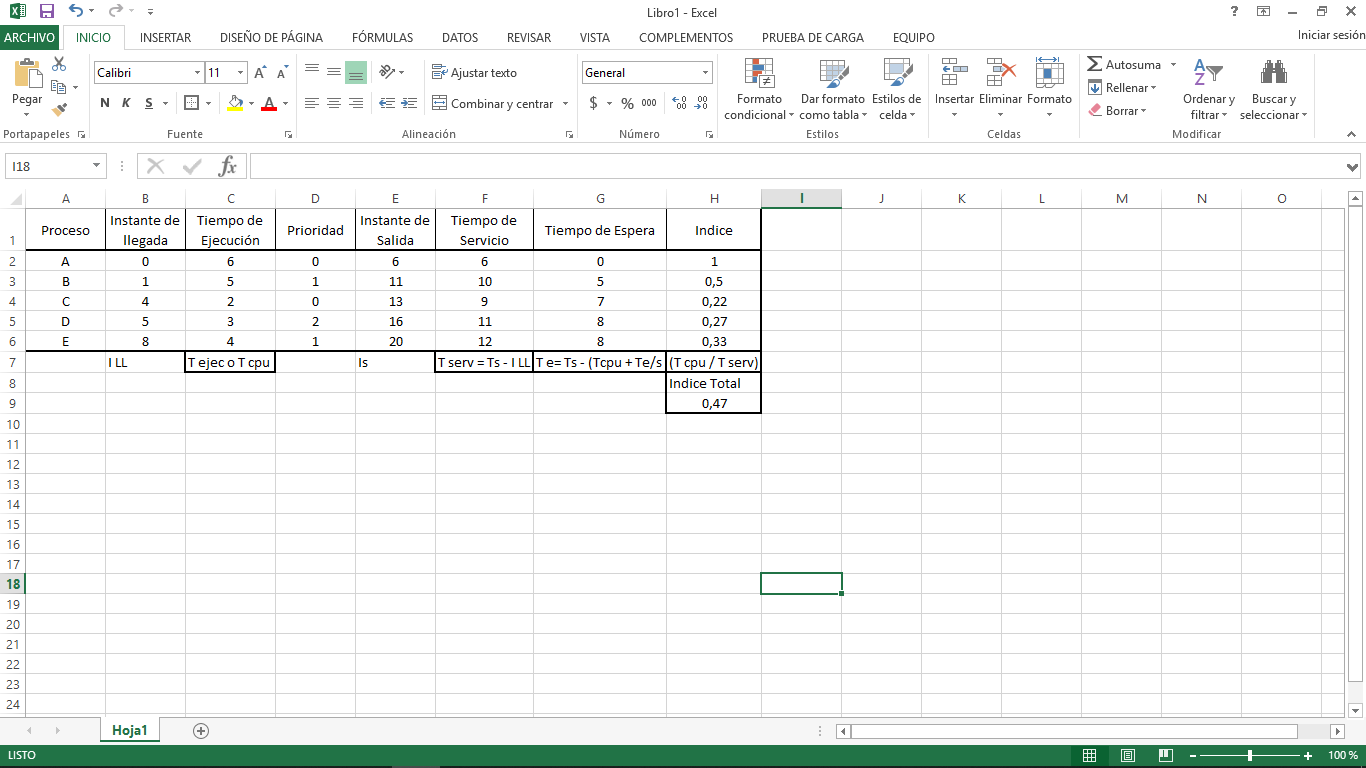
* **Objetivo Principal y Fundamental**: es que el microprocesador no tenga tiempos muertos, o sea que se ejecute el mayor tiempo posible los procesos.
* **Utilización de la CPU:** se estima que la utilización de la CPU tiene que variar entre un 40% y un 90% de uso, lo óptimo sería que siempre este en el 90%.
* **Tasa de Procesamiento: e**s el rendimiento que tiene el microprocesador, es el número de procesos que salen del sistema por unidad de tiempo.
* **El Tiempo de Ejecución o Tiempo de CPU:** es la cantidad de tiempo que un proceso esta en el microprocesador.
* **Tiempo de Servicio o Tiempo de Retorno:** es el tiempo total que un proceso dura en el sistema, desde el instante que llega hasta el instante que sale.
* **Tiempo de Espera**: es el tiempo que un proceso esta en competencia sin realizar nada útil, o sea simplificando es el tiempo que esta en LISTO. Este tiempo tiene que ser lo más corto posible.

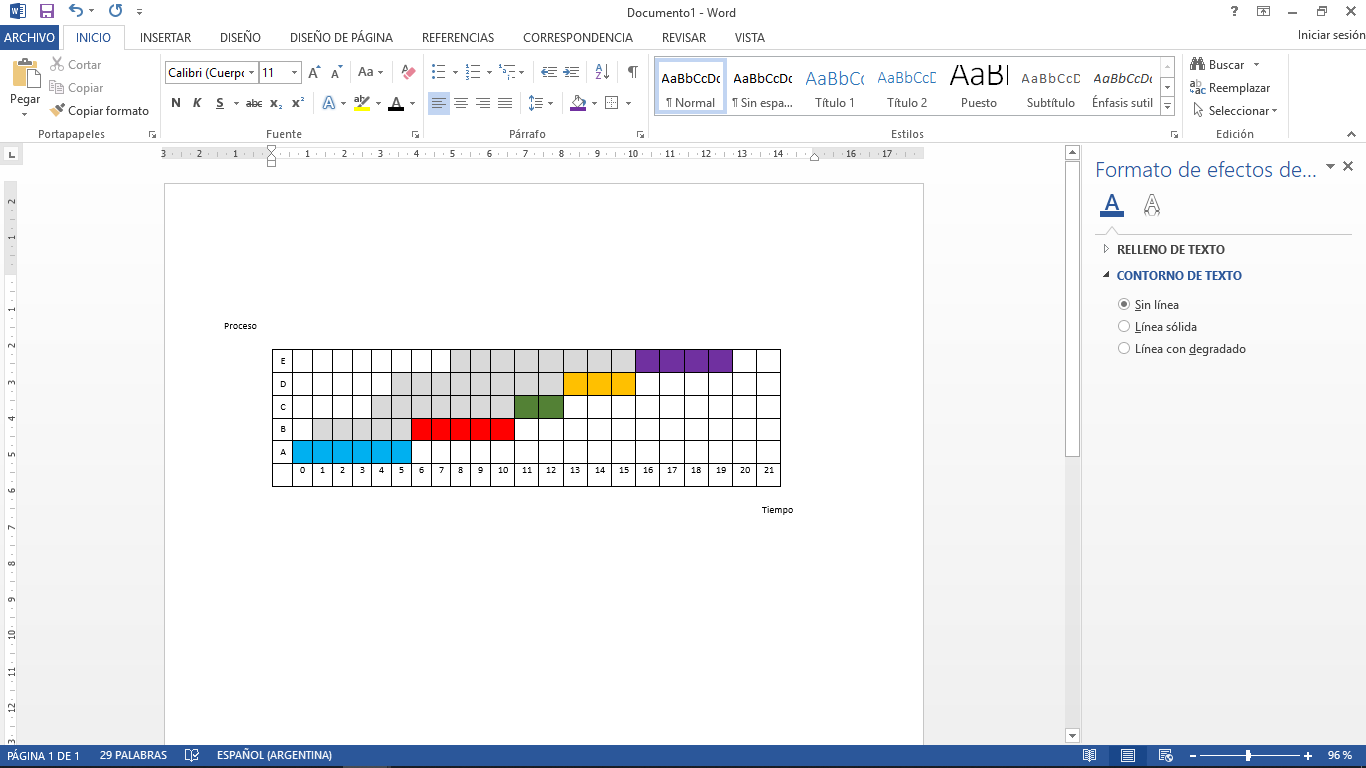


**Tiempo de Respuesta**: es un tiempo que se usa hoy en día para procesos interactivos. Es el tiempo que demora el proceso para emitir su primera respuesta, o sea es el tiempo que demora el proceso en hacer el primer acceso a bloqueado. Cuán rápido un proceso da señales de vida.

**ALGORITMOS DE PLANIFICACION**

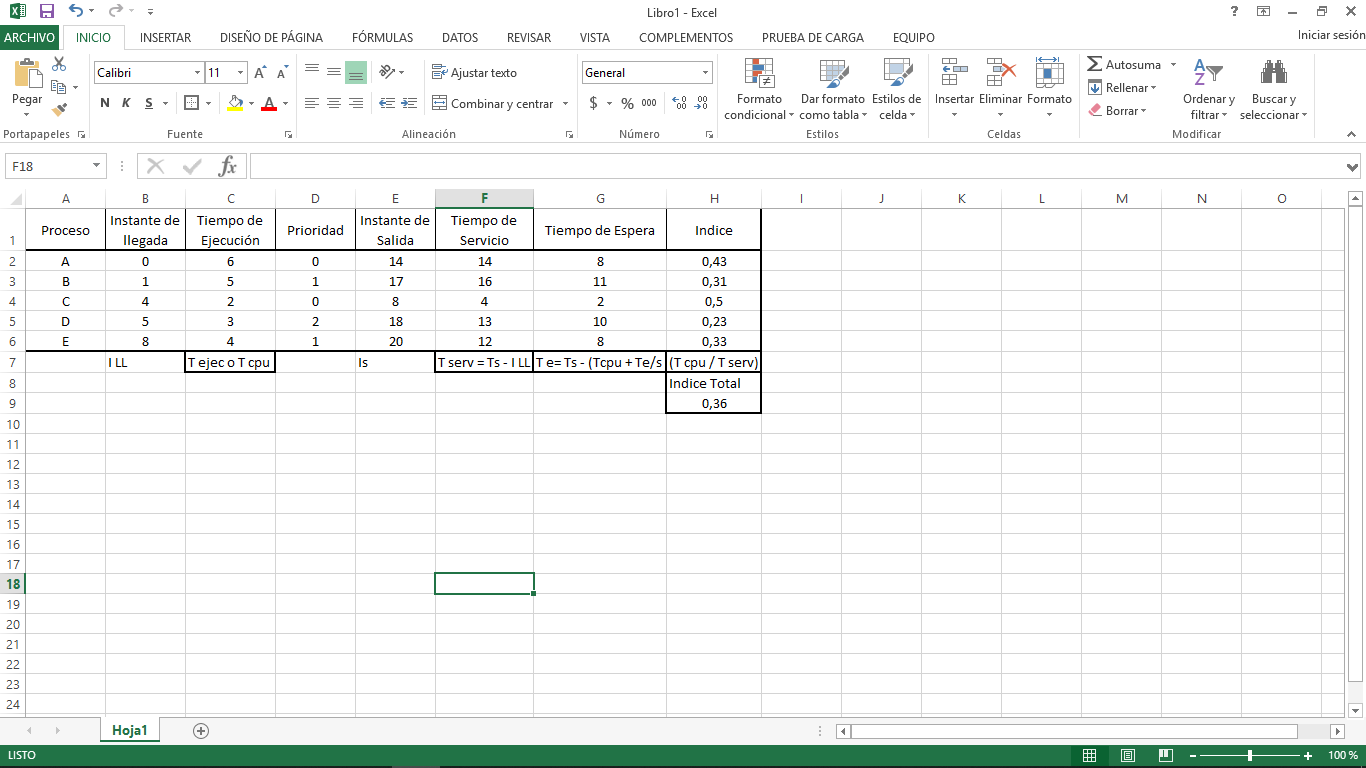
* **FCFS** (Primero en llegar, primero en ser Servido)

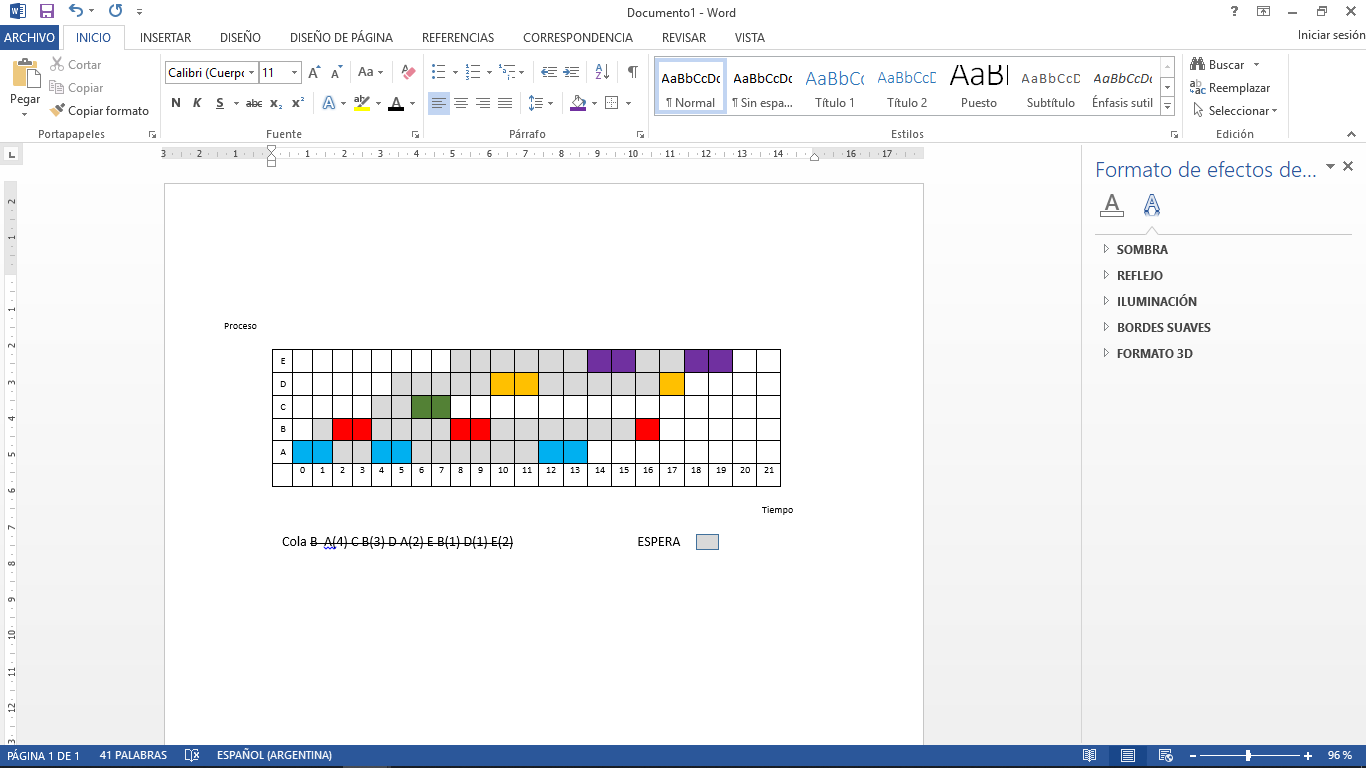
Es el más sencillo de hacer y también el más justo, es el más predecible. Tiene uan cola de espera tipo FIFO. Es un proceso puramente apropiativo, es no expulsivo, sin desalojo, es decir que los procesos se van encolando a medida que van llegando.

Indice: varia entre 0 y 1, cuanto más se acerque a 1 más óptimo es, porque tiene menos tiempo de espera.

* **ROUND ROBIN (R-R)**

Puramente No Apropiativo, expulsivo, con desalojo, es para tiempo compartido. Es muy importante mantener la cola (Fifo).

Con Quantum = 2



**En el instante 4 ocurren dos cosas:**

Termina B y entra C, como la salida de B demora porque hay una latencia de despacho.

¿Quién llega primero a la Cola? Llega C, y luego va a entrar a la cola B, porque en este instante B esta en el micro y tiene que producirse un cambio de contexto, o sea tiene que ejecutarse el despachador, esto tiene un tiempo, en cambio C entra al sistema y no tiene ninguna demora, es inmediato.

Conclusión: en caso de empate, el primero que llega a la cola, va a ser el que llega no el que sale del micro. ¿Por qué ocurre esto? Porque el proceso que sale del micro tiene una LATENCIA DE DESPACHO, un tiempo que demora por el cambio de contexto.

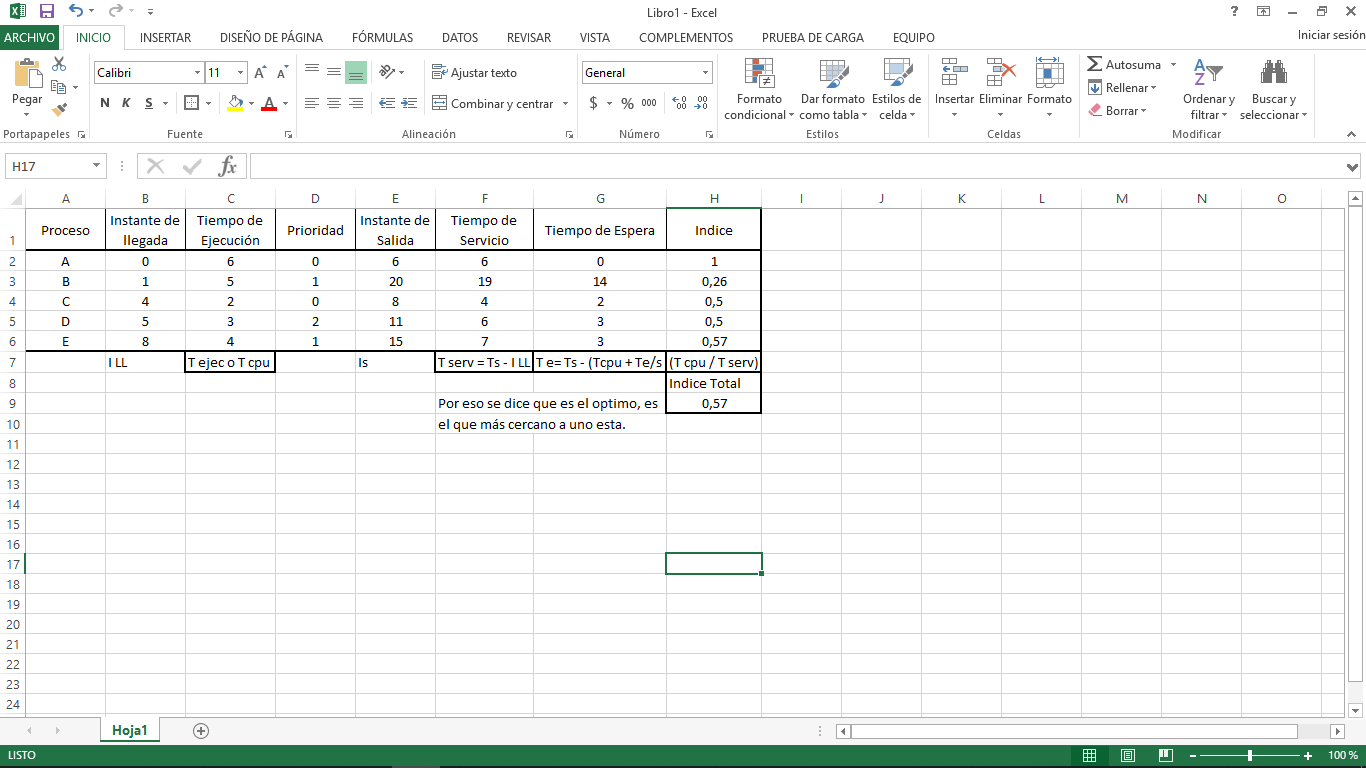
Este algoritmo prioriza aquellos procesos con ráfagas de CPU cortas, como por ejemplo C.

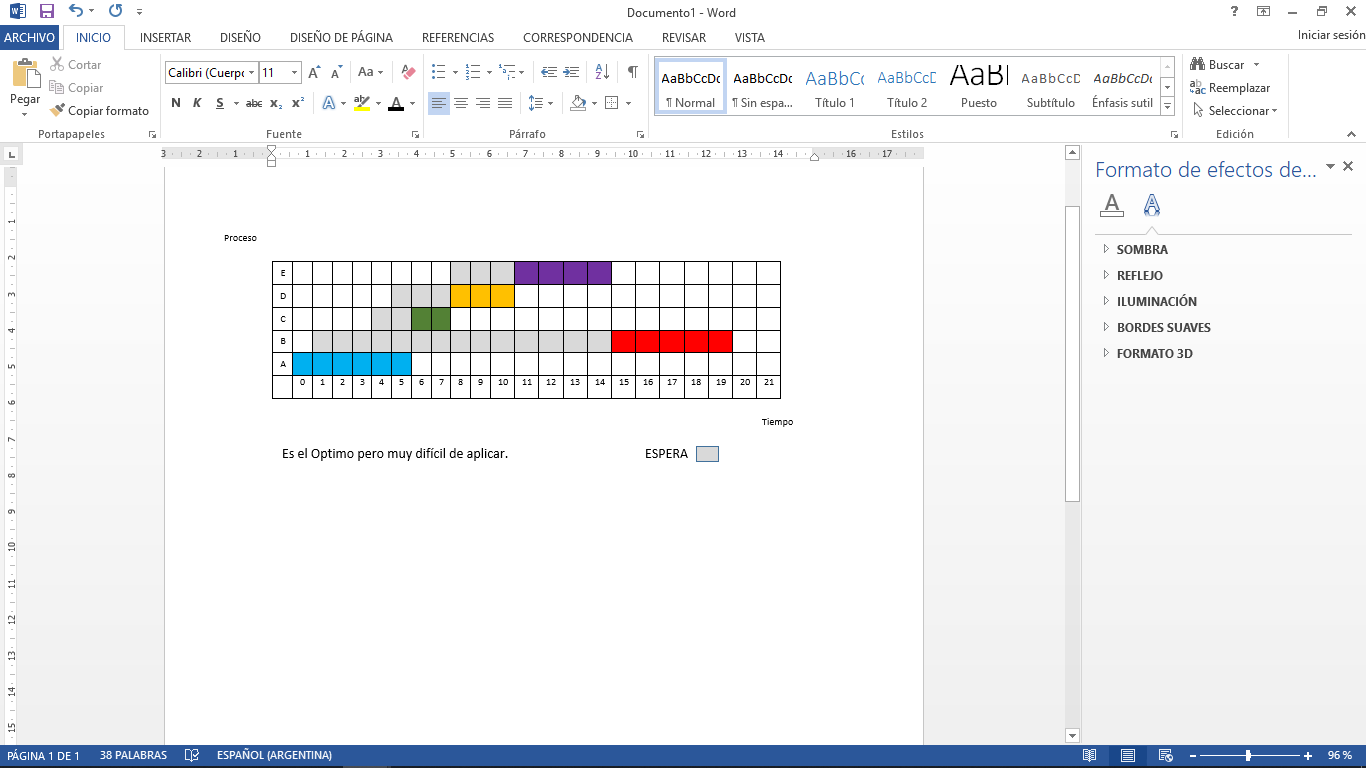
* **EL SIGUIENTE TRABAJO MAS CORTO (SJF)**

Este algoritmo podemos decir que es el Optimo, pero es dificil de realizar. Tiene dos con desalojo y sin desalojo.

**SJF:**

* Sin Desalojo 🡪SJN: Primero el trabajo más corto.
* Con Desalojo 🡪 SRT: Primero el trabajo más corto restante.

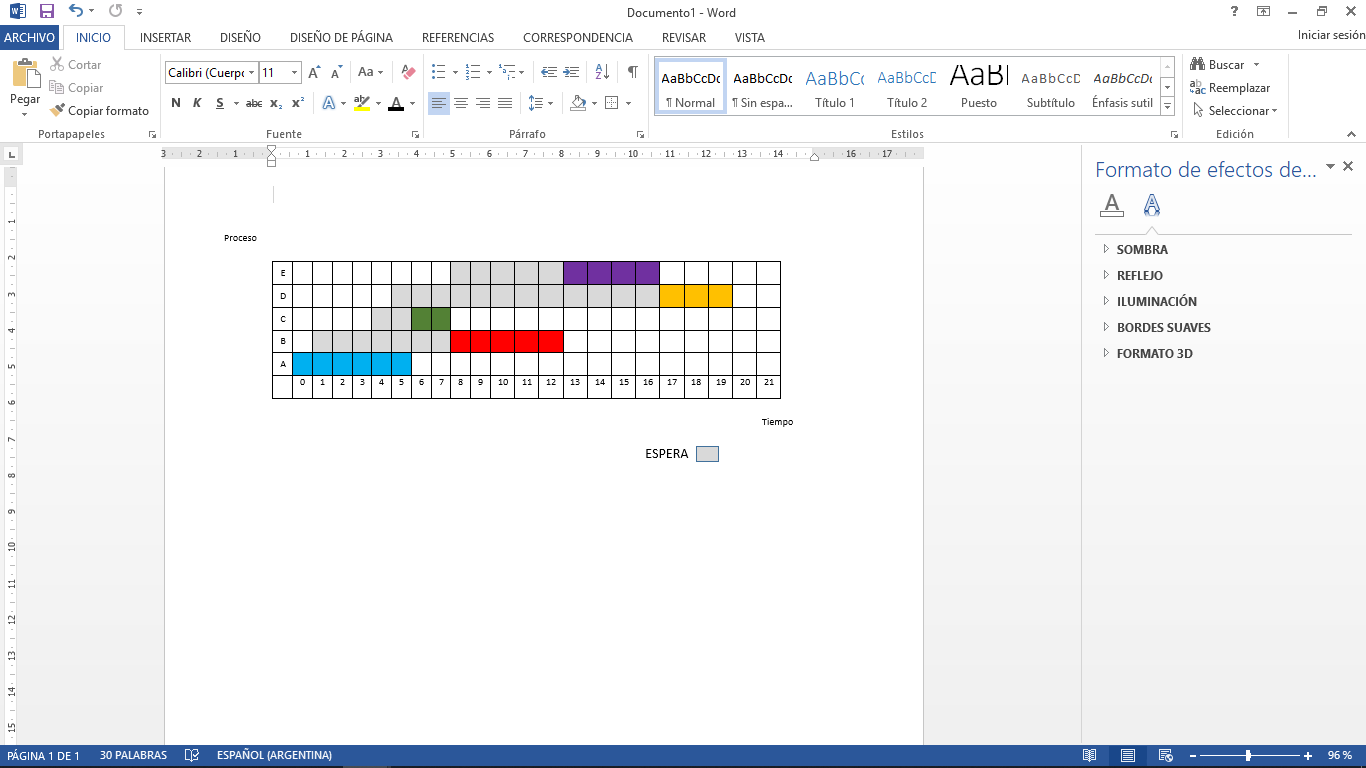
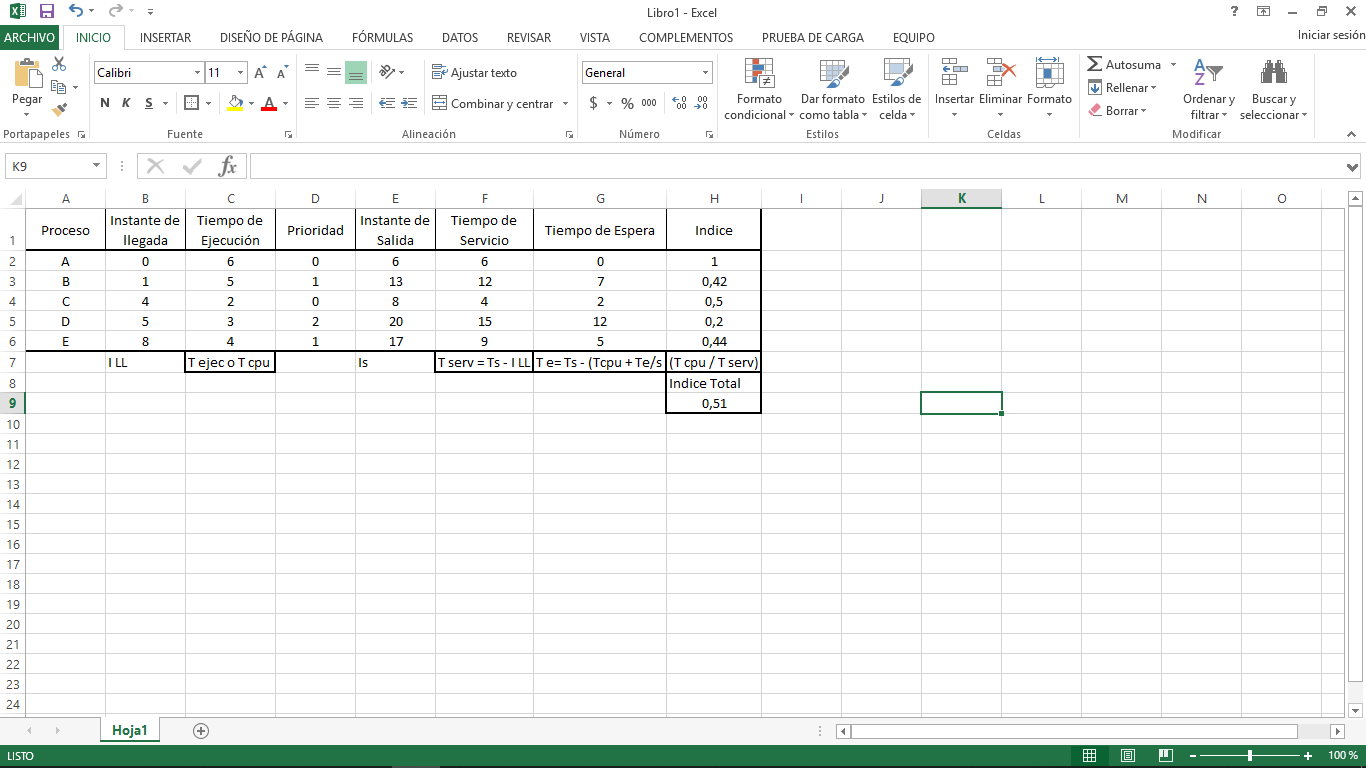




**POR PRIORIDADES**

La prioridad es un valor que pone cada programador según el grado de importancia del programa. La prioridad siempre es un número entero incluyendo el cero, a mayor número menor prioridad.

Cuando la prioridad existe es un algoritmo con DESALOJO, es decir que los que tienen mayor prioridad sacan a los de menor prioridad.Cuando es SIN DESALOJO se forma una cola FIFO pero ordenada por prioridad, es decir que no es por orden de llegada sino que se ordena por prioridad.



**PROBLEMA QUE PRESENTA**

**BLOQUEO INDEFINIDO O INANICION**

Es decir por ejemplo si tenemos con prioridad 4 y todos los procesos que entran tienen mayor prioridad (número menor), el proceso 4 nunca va a tener el microprocesador; eso se llama INANICION cuando el proceso esta en la cola de LISTO indefinidamente por eso se dice que esta en un bloqueo indefinido.

Para evitar la INANICION existe un mecanismo dentro del S.O que se llama ENVEJECIMIENTO, que es cuando un proceso esta mucho tiempo en un estado de ESPERA se le incrementa la prioridad o sea se le da una prioridad más alta. En sintesis consiste en incrementar la prioridad a un proceso después de un lapso determinado de espera.

* **COLA DE MULTINIVEL**

La cola de multinivel, es cuando tenemos varias colas para un microprocesador y a cada cola le damos un nivel de prioridad. La facilidad es que dentro de cada cola se puede tener distintas planificaciones.

El problema es que también se presenta la INANICION, por esta razón se ha hecho la COLA MULTIPLE NIVEL CON RETROALIMENTACION, que también es una práctica que trata de evitar la INANICION.

**¿Qué ocurre si tenemos MULTIPROCESADORES?**

* Si tenemos multiprocesadores pero tenemos uan sola COLA no hay problema, estamos en presencia de un MULTIPROCESADOR SIMETRICO, cuando un procesador se desocupa va a ir a sacar el próximo proceso de la COLA.
* Si tenemos multiprocesadores y estamos en un sistema ASIMETRICO que tienen un jefe y ese jefe administra la cola, ya va a estar sincronizado por que ese jefe va a mandar los trabajos a los procesadores esclavos según su criterio.

Ahora podemo llegar a tener el problema de que los cambios de contexto demoren mucho tiempo, porque existe LA AFINIDAD DE CARGA, que quiere decier que si un proceso siempre se ejecuta en el mismo microprocesador ya va a tener acomodado los registros asociados de memoria que serian los que van al CACHE por lo tanto la conmutación de contexto sería menor, en cambio en otro que nunca uso las direcciones de CACHE van a ser otras porque el microprocesador elige otras direcciones de memoria, es decir la CPU va a tener que realizar más procesos para encontrar las direcciones de memoria.

Si cada procesador tiene su propia planificación, no solo vamos a tener el problema de la afinidad de carga, sino que se nos presenta el EQUILIBRIO DE CARGA. Sucede cuando un microprocesador se queda sin trabajo entonces lo que tiene que hacer es sacar cargs a otros microprocesadores que estan sobrecargados.

**PLANIFICACION DE HILOS**

La planificación de hilos lo podemos tener a nivel de Kernel y a nivel de usuario.

Los hilos a nivel Kernel ya estan planificados por el microprocesador entonces son más fáciles, pero a nivel de usuario son sincronizados por el usuario o sea tiene que ver la planificación que le dara. O sea que a nivel de usuario la planificación va a estar gestionanda por la biblioteca API, y a nivel de Kernel esta planificado por el S.O.

**EVALUACION DE ALGORITMOS**

1. Forma determinista por medio de una GANTT (esta hecho en el plano).
2. Por modelo de cola, que es probabilidades y estadísticas.
3. Con Simulación, se simulan colas y se ven los comportamientos según el algoritmo.
4. Simplemente con la Implementación, programar y probar.