

Tecnológico de Costa Rica



## Sistemas Operativos Avanzados

Apuntes de Clase: 06 febrero

Cinthya Salazar Masis

# Contenido

Aviso Importante .....	3
Discusión del programa del curso.....	3
1. Contacto.....	3
2. Apuntadores.....	3
3. Resúmenes .....	4
4. Presentaciones Orales.....	4
5. Proyectos .....	5
6. Paper de Investigación .....	5
7. Tareas y Quices .....	5
8. Evaluación .....	5
Repaso de Conceptos Básicos .....	6
Nacimiento de los Lenguajes de Alto Nivel.....	6
John William Mauchly .....	6
John Adam Presper Eckert.....	6
ENIAC – Electronic Numerical Integrator And Computer .....	6
EDVAC - Electronic Discrete Variable Automatic Computer .....	7
John von Neumann .....	8
Arquitectura de von Neumann .....	8
Unidad de Control .....	9
Diseño de la Unidad de Control .....	9
Unidad de Control Alambrada .....	10
Sir Maurice Vincent Wilkes .....	10
EDSAC - Electronic Delay Storage Automatic Computer .....	11
Unidad de Control Microprogramada .....	11
¿Aplicación Microprogramada? .....	12
Lenguaje Máquina .....	12
Lenguaje Ensamblador .....	13
John Backus .....	13
4 Grandes.....	14
Lenguajes de Alto Nivel .....	14
Sistemas Operativos Primitivos .....	15



## Aviso Importante

Lectura y resumen para la clase del 13 de febrero:

**0101**

**0103**

**0106**

**0107**



## Discusión del programa del curso

### 1. Contacto

- Dr. Francisco Torres Rojas
- Correo: [torresrojas.cursos@gmail.com](mailto:torresrojas.cursos@gmail.com)
- Las dudas preferiblemente al foro por si en caso de que otra persona tenga la misma duda pueda ser aclarada para todos.

### 2. Apuntadores

- Cada clase tendrá apuntadores oficiales, el número de apuntadores por clase lo define el profesor dependiendo del número de estudiantes que haya en el curso, así como también el número de veces que cada estudiante deba ser apuntador.
- Los apuntes deben subirse al foro de curso antes de la siguiente clase.
- Foro: [www.ic-itcr.ac.cr](http://www.ic-itcr.ac.cr)
- Si el estudiante no tiene acceso al foro debe enviar un correo para solicitarlo al soporte del TEC.
- El formato para el nombre del archivo en PDF debe ser el siguiente: **SOA17mmdd-K.pdf**, donde **mm** es el mes, **dd** la fecha del día y **k** el número secuencial para distinguir los archivos de la misma clase.
- Los apuntes son revisados por el profesor y cuenta para la nota final.

### 3. Resúmenes

- Leemos papers viejos porque es importante entender la historia.
- Los resúmenes se entregan en forma impresa.
- Cada resumen es por separado, es decir que si son 4 papers se deben entregar 4 resúmenes.
- Los resúmenes pueden ser escritos en inglés.
- El resumen no debe ser menor ni mayor a 4 páginas.
- Cada paper asignado por el profesor para leer debe venir con un resumen y aparte responder las siguientes preguntas:

**¿Cuál es el problema que plantea el paper?**

**¿Por qué es un problema interesante o importante?**

**¿Qué otras soluciones se han intentado para resolver este problema?**

**¿Cuál es la solución propuesta por los autores?**

**¿Qué tan exitosa es esta solución?**

- En la esquina del documento se debe escribir en un cuadro el número del paper en tamaño más grande. Ejemplo:

Título del paper	1201
------------------	------

### 4. Presentaciones Orales

- El profesor asigna un paper para exponer con formato de conferencia científico.
- La presentación no puede durar más de 20 minutos o menos de 16 minutos.
- Las presentaciones no se cambian de fecha.
- La ausencia el día de la presentación.

## 5. Proyectos

- Los proyectos serán realizados en grupos de 4 personas.
- Si se quieren cambiar los grupos se le debe consultar al profesor.
- Son 3 proyectos durante el curso.
- Algunos proyectos serán revisados mediante demos en una fecha establecida.
- Los proyectos deben cumplir al pie de la letra las especificaciones dadas por el profesor.

## 6. Paper de Investigación

- El paper de investigación se debe realizar en grupos.
- El profesor publicará en el foro una lista de temas, pero los grupos también puede proponer su tema.
- El tema debe ser negociado con el profesor mediante correo electrónico.
- El paper debe ser hecho en LATEX y escrito en inglés.

## 7. Tareas y Quices

- Se acordó con el profesor que se eliminarán las 2 notas más bajas, por lo que solo serán tomadas en cuenta 8 noches.

## 8. Evaluación

La evaluación del curso fue aprobada por todos los estudiantes presentes.

Trabajo en clase	Apuntes, Participación en clase, foros
Presentación Oral	Paper
Quices y Tareas	8 notas y se eliminan las dos más bajas
Parciales	La mejor nota obtenida vale 15% y el otro 10%
Paper de Investigación	En grupos
Proyectos Programados	



## Repaso de Conceptos Básicos

### Nacimiento de los Lenguajes de Alto Nivel

#### John William Mauchly

- Físico U.S.A (1907 - 1980)
- Ph. D. Física – Johns Hopkins University
- Se interesó por la electrónica en 1941

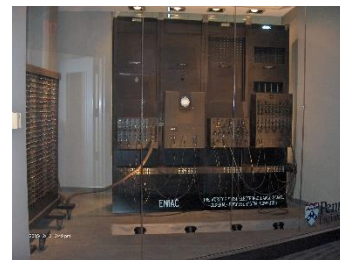
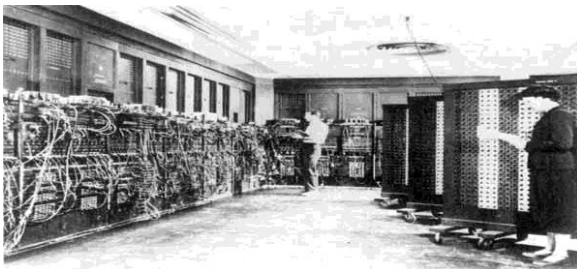


#### John Adam Presper Eckert

- Ingeniero Eléctrico U.S.A (1919 – 1995)
- Conoce a Mauchly en 1941

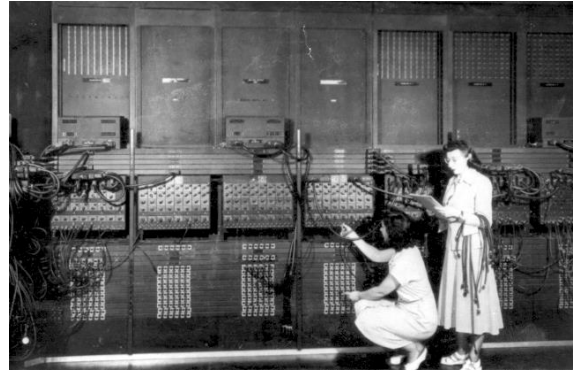


### ENIAC – Electronic Numerical Integrator And Computer



- Computadora Electrónica, U.S.A (1946 – 1955)
- Inventada y fabricada por Mauchly y Eckert – U. Penn
- Primera computadora electrónica “programable”.

- Programable se refería a cambiar cables de un lado a otra para ejecutar diferentes funciones.
- Costó 500 dólares.
- Funcional la mitad del tiempo.
- Giant Brain – Publicidad hecha.
- Las mujeres se encargaban de la programación.
- Resolvía problemas de balística del ejercito de U.S.A, pero se podía cambiar programa.



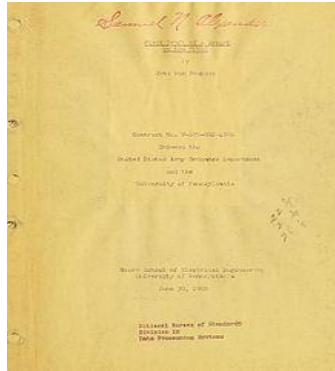
## EDVAC - Electronic Discrete Variable Automatic Computer



- Computadora Electrónica Binaria, U.S.A (1949 – 1961).
- John Mauchly y Presper Eckert, otro proyecto más de la Universidad de Pensilvania.
- Se empezó a trabajar en ella antes de que ENIAC estuviera lista y se creó para corregir los errores de esta.
- Funcional 20 horas diarias
- MTBF (promedio del tiempo entre fallos de un sistema) de 8 horas.
- Memoria de 1000 palabras de 44 bits (6k), 6000 tubos al vacío y 56Kw de energía
- 30 personas en cada turno de 8 horas.

- El punto más importante que marcó esta computadora fue el **programa en memoria**. Ahora el programa se encuentra en memoria y no en los cables.
- Crea el software y gracias a esto hoy existimos.

## John von Neumann

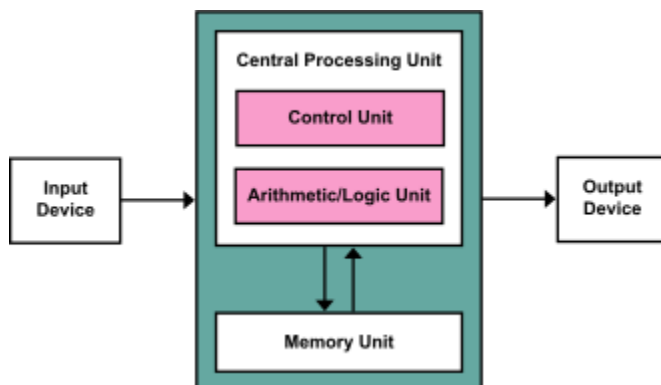


- Matemático, Hungría/U.S.A (1903 – 1957)
- Uno de los mejores matemáticos del siglo XX
- Contribuciones en muchos campos:

**Arquitectura de Computadores:** Ciencia de la Computación, Investigación de Operaciones, Teoría de Juegos, Teoría de Conjuntos, Economía, Física, etc.

- Formo parte del proyecto Manhattan – Bomba atómica.
- Miembro original del IAS en Princeton (Instituto de Estudios Avanzados).
- **First Draft:** Todo lo que escucho en una reunión lo mejoré y lo puso en el paper como si fuera suyo.

## Arquitectura de von Neumann

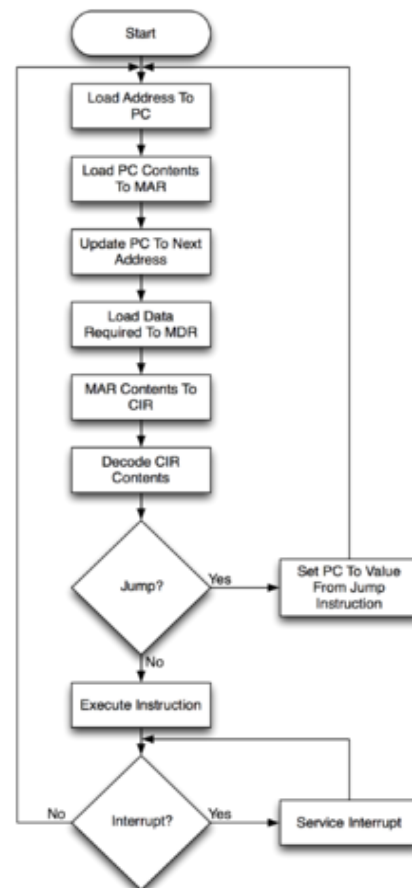




- Propuesto en 1945 por von Neumann – ¿Ideas robadas?
- Antes las computadoras eran de programa fijo (ENIAC).
- Programar era rediseñar el alambrado.
- Ideas novedosas: Programa que se lee de una memoria, ciclo de fetch, conjunto de instrucciones, software.
- La unidad aritmética lógica es la encargada de realizar cálculos.

## Unidad de Control

- La unidad de control es hardware responsable de ejecutar el ciclo de fetch:
  - 1) Recuperar de memoria la siguiente instrucción.
  - 2) Actualizar **PC**.
  - 3) Decodificar instrucción.
  - 4) Decodificar argumentos (modos de direccionamiento).
  - 5) Ejecutar instrucción.
  - 6) Revisar interrupciones.
  - 7) **GO TO 1**



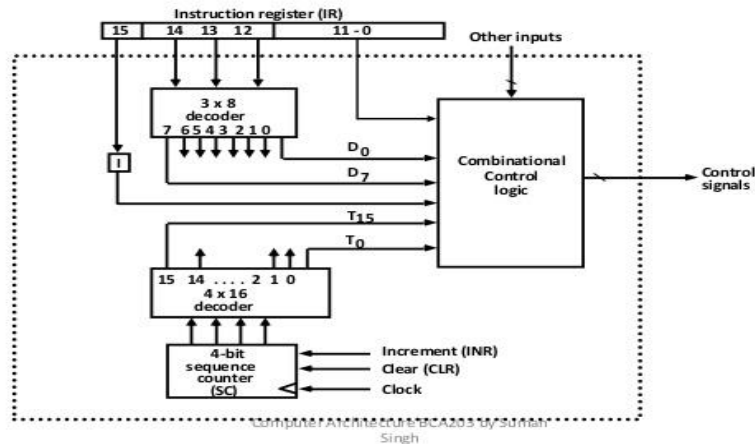
## Diseño de la Unidad de Control

- Organización de Computadoras vs Arquitectura.
- La UC establece el lenguaje máquina o arquitectura de la computadora.
- 2 posibilidades de diseño: Alambrada y Microprogramada.

## Unidad de Control Alambrada

### Hardwired Timing And Control Unit

#### Control unit of Basic Computer



- Se implementa con compuertas lógicas.
- Una arquitectura alambrada es más rápida y eficiente que unidades de control microprogramadas
- Cada instrucción de lenguaje de máquina tiene sus circuitos correspondientes.
- Arquitectura fija, para cambiar el conjunto de instrucciones hay que rediseñar la unidad de control.
- Poco o nada flexibles.

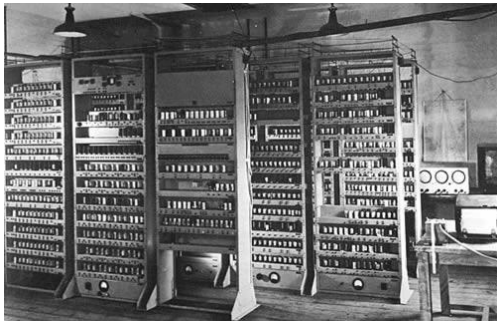
## Sir Maurice Vincent Wilkes

- Físico Inglés (1913 – 2010)
- Uno de los fundadores de Ciencia de la Computación y de la Investigación de Operaciones.
- Muy impresionado con el paper de von Neumann respecto a EDVAC.



- En 1951 inventa el concepto de microprogramación para el diseño de Unidades de Control en CPUs.
- Contribuye con el concepto de subrutinas para FORTRAN.
- Turing Award 1967.

## EDSAC - Electronic Delay Storage Automatic Computer



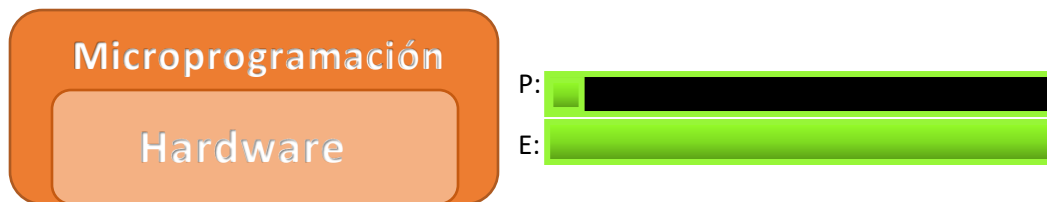
- Computadora electronica U.K (1946 - 1958).
- Inventada y fabricada por Maurice Vincent Wilkes

## Unidad de Control Microprogramada

- Microprograma (No se puede programar).
- Microprogramable (Se puede programar o modificar).
- Hay una organización de computadora subyacente.
- La arquitectura se crea sobre esta organización.
- El conjunto de instrucciones se define sobre la organización.
- Cada instrucción de lenguaje máquina se construye con múltiples microinstrucciones, tomadas de una micromemoria.
- Más flexible pero más lenta.
- Lo único real es el hardware.
- Se rodea con microprogramación para darle un comportamiento.
- Creamos el lenguaje máquina – Arquitectura.

## ¿Aplicación Microprogramada?

- Teóricamente es posible, pero implica demasiado tiempo.
- Productividad programador: cantidad de líneas de código buenas por unidad de tiempo.
- Eficiencia computador = Cantidad de recursos que consume para producir un resultado.



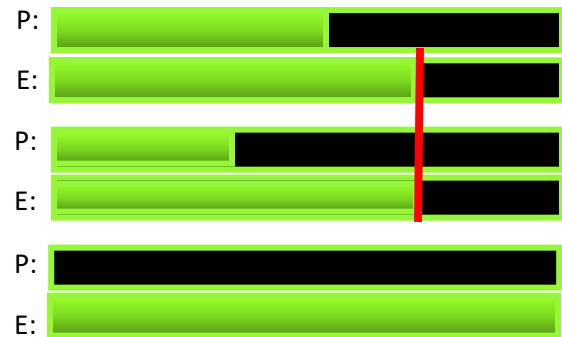
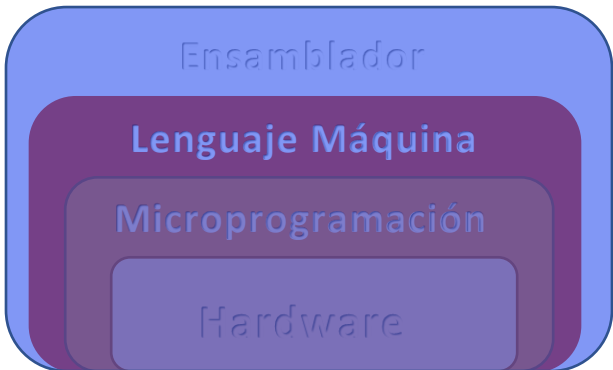
## Lenguaje Máquina

- 0's y 1's – Encima de la microprogramación.
- No todo lo que se puede hacer en microprogramación se puede hacer en lenguaje máquina.
- La microprogramación define el lenguaje máquina.
- Mucho más fácil que microprogramar.
- Responsabilidad del programador en lenguaje máquina:
  - 1) Colocar el programa en RAM
  - 2) Códigos de operación
  - 3) Modos de direccionamiento
  - 4) Formato de argumento
  - 5) Calcular tamaños y desplazamiento
- No portátil.
- 



## Lenguaje Ensamblador

- Todo lo que se puede hacer en ensamblador se puede hacer en lenguaje máquina.
- Programar en lenguaje máquina es tedioso y propenso a errores.
- En los 1950's inventan ensamblador.
- Programa que recibe nemónicos y nombres simbólicos con cierta sintaxis simple y genera lenguaje máquina.
- Calcula desplazamientos.
- Pseudoinstrucciones y macros
- Mucho más fácil – No portátil

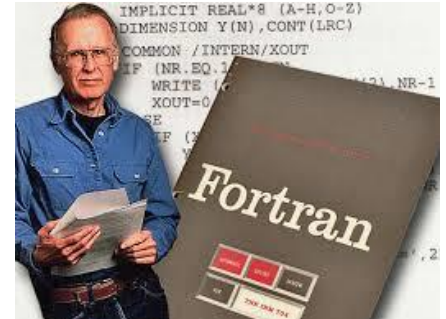


## John Backus



- Matemático y Científico de la computación, U.S.A (1924 – 2007)
- M. Sc. en Matemáticas – Columbia University.
- Trabajó en IBM desde 1950 programando en lenguaje máquina y ensamblador.

- Llego a IBM por casualidad, después de haberse graduado sin saber aún que era una computadora.
- Su primer programa calculaba las posiciones de la luna en lenguaje máquina.
- Inventó Speedcoding para hacer más fácil la tarea de programar.
- Propuso el proyecto FORTRAN para la IBM 704 en 1953.
- Retos:
  - ¿Es posible del todo?
  - ¿Durará mucho en compilar?
  - ¿Generará código eficiente?
- Terminaron en 1956 – primer lenguaje de alto nivel de uso extendido.
- En 1958 crea la BNF para describir cualquier lenguaje de programación libre de contexto.
- En 1977 inventa FP lenguaje de programación funcional.
- Turing Award 1977.



#### 4 Grandes

- Entre 1956 – 1960



**John Backus**  
FORTRAN



**John McCarthy**  
LISP



**Grace Hopper**  
COBOL

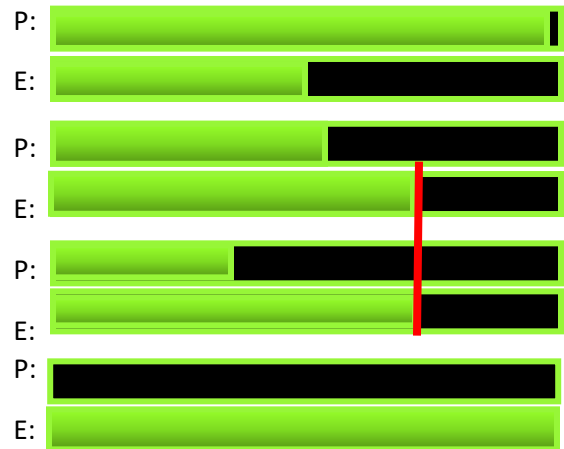


**Peter Landin**  
ALGOL

#### Lenguajes de Alto Nivel

- Existen desde 1958.
- Más fácil programar, mayor productividad.
- Definen una nueva capa.
- Independiente del hardware.
- Portabilidad.
- Abstracción de programación.

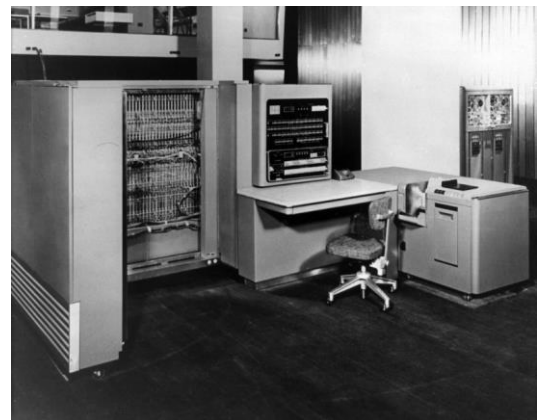
- El hardware se vuelve irrelevante.



## Sistemas Operativos Primitivos

### IBM 701

- Las computadoras no se compraban, se alquilaban.
- Primera computadora comercial (1952).
- Solo se fabricaron 19 computadoras.
- Configuración:
  - 4096 palabras de 36 bits
  - 150 mil inst./seg
  - 150 tarjetas/seg. (I)
  - 100 tarjetas/seg. (O)
- Se alquilaba a \$23,750 el mes (\$211,645 actuales)
- MTBF: 30 min



### IBM 704

- Primera computadora con punto flotante.
- Se vendieron o alquilaban unas 200 máquinas.
- Alquiler \$35 550 al mes (\$303.745 actuales).
- Doble de rápida que la 701 pero incompatible.
- Dispositivos para copiar tarjetas a cinta off-line.
- Memoria de núcleos de ferrita.



- MTBF: 8 horas.
- FORTRAN disponible

### Máquina dedicada



- Uso típico de computadoras en los 1950. Se reservaban horas de máquina.
- La computadora estaba en un solo lugar y si alguien quería usarla debía ir hasta ahí y usarla durante un tiempo determinado.
- El programador / científico / ingeniero era responsable de hacer todo.
- Secuencia de boot manual, lenta y propensa a errores.

### Uso típico – Caso ideal

- 1) Configurar todos los tableros de control.
- 2) Colocar FORTRAN en lectora de tarjetas. Dar boot al computador.
- 3) Colocar fuente FORTRAN en la lectora. Leer. ¿Varias pasadas?
- 4) Si no hay errores, el compilador perfora programa equivalente en ensamblador.
- 5) Colocar Ensamblador en lectora de tarjetas. Dar boot al computador.
- 6) Colocar ensamblador generado por compilador en lectora de tarjetas. Leer. ¿Varias pasadas?
- 7) Ensamblador perfora programa equivalente en lenguaje máquina.
- 8) Colocar lenguaje máquina en lectora de tarjetas. Dar boot al computador. Nuestro programa corre.
- 9) Recolectar listados de impresora.
- 10) Siguiendo usuario, GO TO 1



## Ventajas

- Absoluto control del programador.
- Pocos expertos al inicio.
- Fácil de establecer responsabilidades y controles. Si hay alguien realizada algo era muy fácil darse cuenta quién porque cada usuario tenía un espacio asignado para utilizar la computadora.

## Desventajas

- Trabajo repetido entre un usuario y otro.
- No hay estándares
- Programadores deben ser expertos en hardware.
- Uso ineficiente de los recursos.



## Mejorando Máquina Dedicada

- Idea administrativa: Se crea el operador de máquinas.
- Se recolectan trabajos y se factorizan pasos.
- Configuración estándar
- Procesamiento en lotes. Procesamiento batch.
- Gran incremento del throughput del Centro de Cómputo. (Cantidad de unidades terminadas por unidad de tiempo).

## Productividad del Centro de Cómputo



### ¿Se necesita un operador?

- La idea del operador funcionó muy bien.
- Trabajo casi mecánico.
- Todos los programas batch: compilar – ejecutar – reporte
- ¿Cómo nos deshacemos del operador?
- Software

