

Tecnológico de Costa Rica  
Escuela de Ingeniería en Computación  
Compiladores e Intérpretes

---

**8 de febrero de 2017**  
Samantha Arbuola León

<b>Notas sobre el Curso</b>	<b>1</b>
<b>Proyecto 0: Un MICRO-Compilador</b>	<b>3</b>
<b>Introducción al curso:</b>	
<b>Historia: Nacimiento de los lenguajes de Alto Nivel</b>	<b>4</b>
John William Mauchly	4
John Adam Presper Eckert	4
ENIAC	4
EDVAC	4
John von Neumann	5
Arquitectura von Neumann	5
Unidad de Control	5
Ciclo de Fetch	5
Diseño de la Unidad de Control	6
Unidad de Control Alambrada	6
Sir Maurice Vicent Wilkes	6
EDSAC	6
Unidad de Control Microprogramada	6
¿Aplicación Microprogramada?	7
Productividad vs Eficiencia	7

## Notas sobre el Curso

### Apuntadores

Habrán 2 apuntadores por clase, será rotativo y cada estudiante será apuntador al menos 2 ocasiones. Su responsabilidad es anotar todo lo sucedido en la clase para posteriormente redactar un documento, para la construcción del documento se permiten ilustraciones, gráficos o diagramas para esclarecer alguna explicación, no se permitirá que el documento sea una secuencia de fotografías. Los apuntes tendrán un valor dentro del rubro de trabajo en clase de la evaluación. La fecha de entrega será 4 días naturales, se esperan que los apuntes sean de calidad, se debe subir al [foro](#) un documento en formato **pdf** con nombre **COM17MMDD-k.pdf** donde MM corresponde al número de mes, DD al número de día y k es el distintivo del apuntador. Al ser un pdf este puede ser generado en cualquier editor de texto. Los apuntadores no serán asignados, cada estudiante debe ofrecerse al inicio de la lección.

### Contacto con el profesor

Correo electrónico: [torresrojas.cursos@gmail.com](mailto:torresrojas.cursos@gmail.com)

Foro: <http://www.ic-itcr.ac.cr/index.php/foro>

Consulta: *posiblemente después de clase, el horario está por definirse*

### Quiz

Los quiz se realizarán 1 por semana los miércoles a las 9:30 am, se evaluará la materia vista en la semana y no evaluada desde el último quiz.

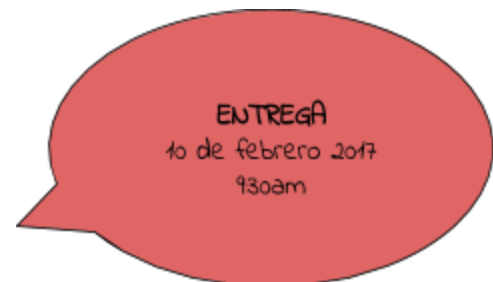
**Tareas** Las tareas se deben entregar impresas al inicio de la lección de la fecha de entrega, se deben realizar en [Latex](#) (instalador [Ubuntu-Debian](#), [videotutorial](#), [tutorial](#)).

ya que esto disminuye el valor porcentual de cada una y beneficia a los estudiantes. El profesor propuso que a final de semestre si hay muchas tareas se eliminen las 2 o 3 con calificación más baja.

### Ficha de Contacto

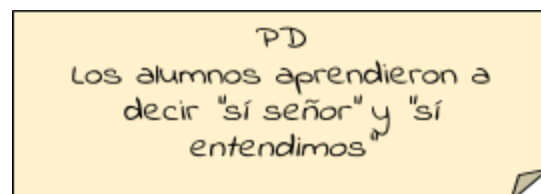
Se debe entregar una ficha de contacto para el archivo del profesor; esto con el fin de realizar anotaciones sobre el estudiante en su reverso para referencias futuras. La ficha debe contener:

- Foto tipo pasaporte del estudiante
- Nombre completo
- Correo electrónico
- Teléfono personal
- Teléfono para contacto en caso de emergencia
- Áreas de Interés
- Información Adicional



### Proyectos

Se desarrollaran en Linux en el lenguaje C

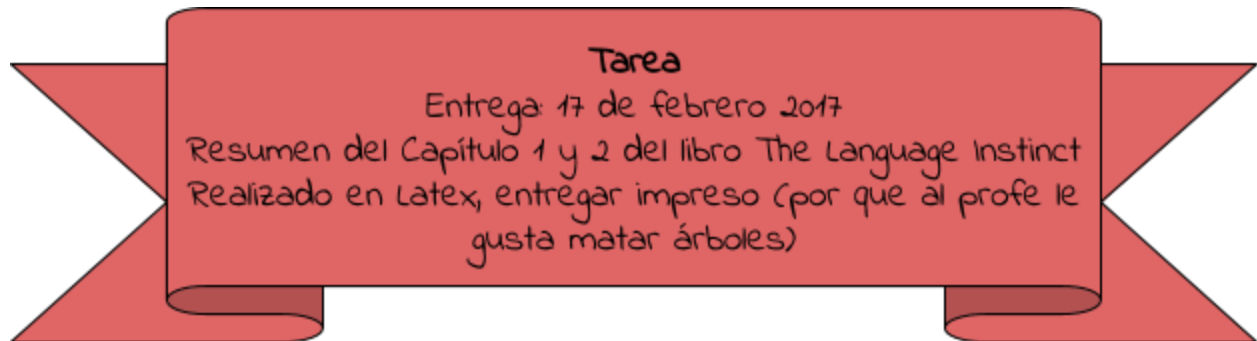


## Libros

No hay un libro oficial para el curso, se estará trabajando con material de diferentes libros más es importante contar con un libro sobre compiladores para consulta y estudio.

El profesor asignará lecturas tanto de la bibliografía recomendada en la carta al estudiante como de otros escritos.

que devolverlos la Recordar:



## Fraude Académico

Habrà cero tolerancia al fraude, ya sea el intento exitoso o no, ya que el objetivo de las evaluaciones es demostrar *qué saben hacer*. Si un estudiante hace fraude se le aplicará el castigo más severo permitido por el reglamento de TEC.

Con esto finaliza la explicación sobre la evaluación del curso



## Proyecto 0: Un MICRO-Compilador

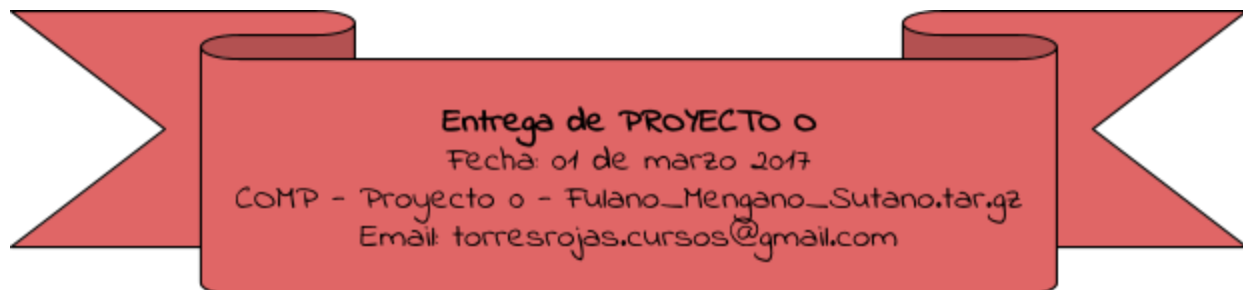
Debe desarrollar un compilador para el lenguaje Micro, el capítulo 2 de “*Crafting a Compiler with C*” de Fischer y LeBlanc Jr. describe lo que se solicita y para ello se debe conseguir e instalar un simulador para Raspberry Pi. Esto ya que el compilador en C debe generar código en ARM para ser ejecutado en el simulador de Raspberry Pi corriendo sobre Linux. Debe familiarizarse con el software requerido y el lenguaje ARM.

En Micro contará con restricciones como:

- El único tipo de datos es Integer
- Los identificadores se declaran implícitamente con un máximo de 32 caracteres
- Los literales son hileras de dígitos
- Palabras reservadas: `begin`, `end`, `read`, `write`.
- Las palabras `begin` y `end` son los delimitadores de los programas y las líneas deben finalizar con `;`
- Hay 2 tipos de instrucciones: asignaciones y operaciones
- Los comentarios inician con `--` y terminan con el fin de línea

Se deben incluir las extensiones descritas en los ejercicios 8 y 10 del capítulo 2.

**Recordar:** TODO EN C, DEBE COMPILAR Y EJECUTAR EN LINUX.  
Grupos de trabajo de 3 personas.



## Introducción al curso: Historia: Nacimiento de los lenguajes de Alto Nivel



**John William Mauchly**

(1907-1980) Estadounidense. PhD Física, Johns Hopkins University.



**John Adam Presper Eckert**

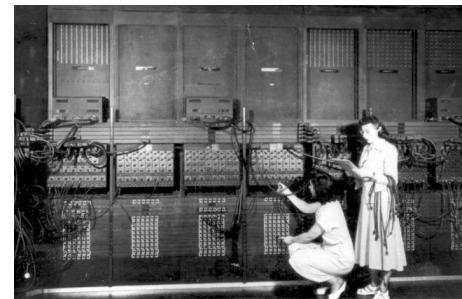
(1919-1995) Estadounidense, Ingeniero Eléctrico. Conoce a Mauchly en 1941

### ENIAC

(1946-1955) *Gian Brain*. Electronic Numerical Integrator And Computer

Creada en la Universidad de Pensilvania por Mauchly y Eckert, con un costo monetario de \$500mil de 1946, es decir, unos \$6,700,006.37 actuales según la [calculadora de inflación](#).

ENIAC fue el primer computador de Propósito General y programable; era programable al intercambiar las conexiones mediante el cableado que comunicaba los circuitos, era 1000 veces más rápido que cualquier computador de la época. Programada por mujeres para resolver problemas de balística, ENIAC contó con un rendimiento del 50% del tiempo.



### EDVAC

(1949-1961) Electronic Discrete Variable Automatic Computer

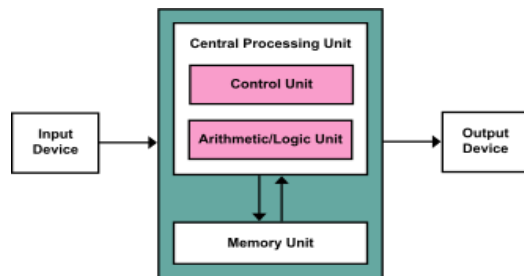
2da computadora creada por Mauchly y Eckert, era electrónica binaria y permitía 20 horas de trabajo continuo con un MTBF de 8 horas sin presentar problemas físicos. Contó con 1000 palabras 44 bits de memoria, alrededor de 6KB. Contó con 6000 tubos de vacío (válvulas termoiónicas), su dimensión física ocupó  $46m^2$ , consumía 56 KW de energía y laboraban 30 personas por cada 8 horas. Su costo fue similar al de ENIAC. Al equipo de trabajo se unió como consultor John von Neumann.



**John von Neumann**

(1903-1957) Matemático, considerado una de las mentes más brillantes de la historia, trabajó en campos como Arquitectura de Computadores, Ciencias de

Computación, Teoría de Juegos, Teoría de Conjuntos, entre otros. Parte del Proyecto Manhattan(bomba atómica 2da Guerra Mundial) y miembro de IAS (Institute for Advanced Study, donde se albergaba a científicos tras emigrar a causa de la 2da Guerra Mundial). Publica **First Draft** en 1945, tras una reunión con Mauchly y Eckert donde se compartió sobre el proyecto EDVAC (por lo que es visto como plagio), se populariza como el nombre de *Arquitectura von Neumann*.



## Arquitectura von Neumann

Revoluciona la computación, ya que cuenta con un conjunto de instrucciones y se lee un programa desde memoria(lo que introduce el concepto de *Software*) por medio del *Ciclo de Fetch*(es ejecutado en la unidad de control).

### Unidad de Control

Es hardware cuya función principal es ejecutar el ciclo de Fetch y cuenta con un Program Counter para llevar el control de las ejecuciones.

### Ciclo de Fetch



### Diseño de la Unidad de Control

Organización de Computadores: Son los registros y transferencias. Hace posibles varios diseños de UC

Arquitectura de Computadores: Está sobre la Organización de Computadores, consiste en un conjunto de instrucciones en lenguaje máquina las cuales son creadas por la UC.

La organización es capaz de modificar el lenguaje máquina para la arquitectura, en este encontraremos: alambrada y microprogramada.

### Unidad de Control Alambrada

Permite ver cómo están alambrados los circuitos de cada instrucción en el componente correspondiente. Para cambiar el conjunto de instrucciones hay que rediseñar el circuito, añadir componentes y revisar las conexiones.

Ventajas:

Es rápida y eficiente

Desventaja:

poco o nada flexible



### Sir Maurice Vincent Wilkes

(1913-2010) Físico británico. Uno de los fundadores de la Ciencia de la Computación e Investigación de Operaciones, acreedor del Turing Award 1967. Fue impresionado por Popper.

En 1951 inventa la microprogramación para diseñar UC en CPU, de donde también nace el concepto de subrutina.

### EDSAC

(1946-1958) Electronic Delay Storage Automatic Computer. Creado por Sir Maurice.

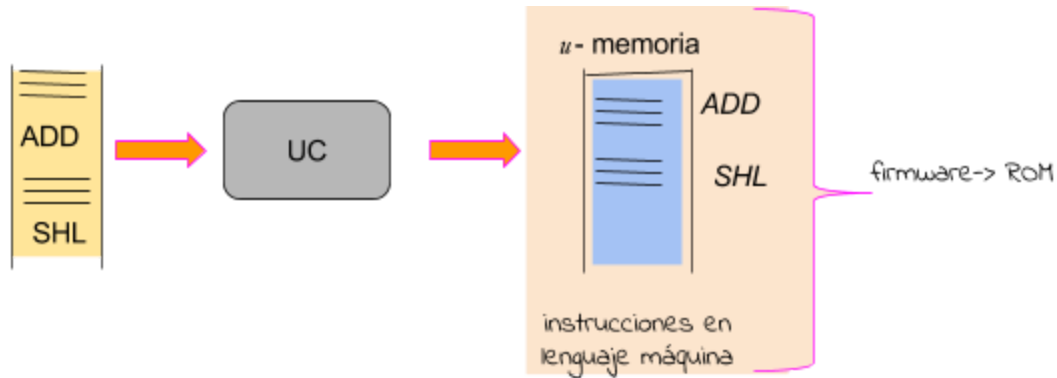
### Unidad de Control Microprogramada

Tomando que la organización crea el lenguaje máquina y define el conjunto de instrucciones se da la organización subyacente, donde cada instrucción en lenguaje máquina se construye con múltiples micro instrucciones tomadas de una micro memoria.

Es mucho más flexible pero más lenta y con un costo de producción menor comparada a la UC Alambrada.

Lo único *real* es el hardware, el cual se rodea de microprogramación para darle un comportamiento, creando el lenguaje máquina y con esto teóricamente podríamos hacer cualquier cosa.

La instrucción es recibida por el UC quien a su vez manda a ejecutar una serie de pases que se encuentra en la micro memoria, donde cada bloque corresponde a una instrucción.

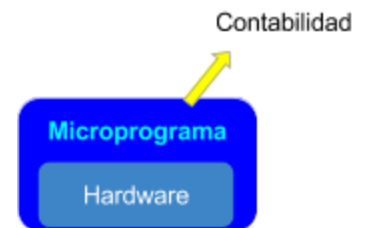


**Recordemos:** Microprogramada != Micro programable.

La segunda es micro programada y es modificable, la primera no es modificable.

## ¿Aplicación Microprogramada?

Para ilustrar: una contabilidad microprogramada requiere de mucho tiempo y va a ser una *tortura* pero será una contabilidad súper eficiente y rápida. La mayor dificultad para hacer un microprograma es la brecha semántica.



## Productividad vs Eficiencia

**Productividad:** es una métrica sobre la cantidad de trabajo generado, en nuestro caso, de líneas de código sin errores por unidad de tiempo al desarrollar un sistema.

**Eficiencia:** cantidad de recursos que un sistema en funcionamiento requiere por cada resultado

