# Generalidades

## Introduccion

En la actualidad el proceso de enseñanza aprendizaje ha ido evolucionando a medida que la tecnología ha ingresado a este campo y le ha proporcionado herramientas para que la educación sea más eficiente y logre mejores resultados.

Hoy en día las herramientas interactivas orientadas a apoyar el proceso de enseñanza-aprendizaje han tomado mucha fuerza, esto debido a la posibilidad que proveen estas herramientas de aplicar los conocimientos teóricos avanzados en clases en ejercicios prácticos, facilitando la visualización del problema.

Una ventaja que otorgan las herramientas interactivas en el área de software educativo es que estas herramientas suelen ser fácilmente adaptables a las necesidades del usuario final, debido a que pueden tomar funcionalidades y características de otros tipos de software educativos como ser simuladores o sistemas con bases de datos.

A lo largo de los años, en la materia de Arquitectura de Computadoras perteneciente a la rama de la informática, el estudio del Ciclo de Instrucción del computador ha traído consigo serios problemas en el proceso de enseñanza aprendizaje y se han probado varios métodos de enseñanza y optado por distintas herramientas para llevarlo a cabo, pero los problemas persisten.

## Antecedentes

### Antecedentes organizacionales

En la actualidad, muchas universidades cuentan con la materia de Arquitectura de Computadoras dentro del pensum establecido para la carrera de Ingeniería de Sistemas. La Universidad Católica Boliviana “San Pablo” no es la excepción,

Dentro del contenido de esta materia, uno de los temas en los que se hace mayor énfasis es el Ciclo de Instrucción del Computador.

El ***objetivo principal*** de este tema es que el estudiante comprenda la secuencia exacta de pasos que debe realizar el CPU, dependiendo de su arquitectura y formato de instrucción (número de direcciones, CO, tipo de direccionamiento), y cómo este influye en la velocidad de ejecución de un programa.

Un ***ciclo de instrucción*** es el período que tarda la unidad central de proceso (CPU) en ejecutar una instrucción en lenguaje máquina. Este comprende una secuencia de acciones que se deben llevar a cabo para ejecutar cada instrucción en un programa.

Para que un CPU realice una tarea, primero debe buscar cada instrucción en la memoria principal y luego ejecutarla. A estos pasos se los conoce respectivamente como ciclo de captación y ciclo de ejecución, y el último depende de varios factores descritos a continuación.

En el ciclo de captación se realiza la búsqueda de la instrucción en la memoria principal y su respectiva decodificación, posteriormente en el ciclo de ejecución se ejecuta la instrucción y finalmente se almacena el resultado.

El número de pasos en el ciclo de captación generalmente es similar, sin importar la arquitectura del CPU, ni la instrucción que se haya dado. En cambio, el número de pasos que hay en el ciclo de ejecución sí depende de la arquitectura del CPU y del formato de instrucción, es decir, del código de operación (CO), del número de direcciones de la instrucción, del tipo de de direccionamiento y de los registros que contenga (AR y DR o banco de registros).

Los ejercicios de ciclo de instrucción pueden ser resueltos, ya sea en binario o en hexadecimal, así como también en lenguaje ensamblador.

En el desarrollo de la enseñanza del tema, se dedica una cierta cantidad de clases para explicar este proceso detalladamente al alumno, primeramente mostrándole la interacción de los componentes del CPU durante el ciclo de instrucción en binario. Posteriormente, para facilitar y acortar la tarea, cuando ya se ha captado el concepto en binario, se procede a utilizar la notación hexadecimal y finalmente se empieza a aplicar el lenguaje ensamblador en todo el proceso.

En binario se enseña la distribución de los bits en los registros (bits utilizados para CO y para direccionamiento), y dependiendo de estos se calcula el rango de valores que se podrán utilizar y el espacio de memoria que podrá accederse. Esto da una idea clara respecto a que los recursos son limitados.

En hexadecimal empieza a mostrarse a más profundidad la interacción entre CPU y memoria principal. En esta parte se ve como se utilizan los recursos del procesador destinados a la interacción con memoria, así como también con los dispositivos de E/S, como por ejemplo los registros MAR (registro de direcciones de memoria) y MBR (registro de datos de memoria), buses internos y externos del CPU (buses con memoria y con los dispositivos de E/S).

En ensamblador se aborda el objetivo principal del tema: que el estudiante comprenda la secuencia exacta de pasos que debe realizar el CPU, dependiendo de su arquitectura y formato de instrucción (número de direcciones, CO, tipo de direccionamiento), y como este influye en la velocidad de ejecución de un programa.

En el desarrollo de las clases para entender el proceso que implica el ciclo de instrucción del computador se elaboran ejercicios que ponen en práctica los conceptos mencionados anteriormente, lo cual toma bastante tiempo dependiendo de la complejidad de ejercicio (aplicar la lógica de funcionamiento, además de la cantidad de instrucciones requeridas en el proceso a seguir) y la cantidad de diagramas a realizar. Por ejemplo, en el inicio del tema, para realizar el primer ejercicio en binario, la docente necesita dedicarle dos clases enteras buscando que los alumnos entiendan el proceso, lo cual implica que se gasta para ello un tercio del tiempo dedicado a esta parte del tema.

Al desarrollar ejercicios se emplea tiempo considerable, alrededor de 5 minutos por cada etapa, en la diagramación de las estructuras necesarias: registros, buses, posiciones de memoria, etc.

Existen otros factores que llevan a que al estudiante le tome más tiempo resolver un ejercicio, como haber cometido errores en el desarrollo del problema, como ser el confundir registros, errores de transcripción de un diagrama al siguiente o en el manejo de direcciones de memoria.

Un dato importante a considerar es que a las dos semanas de iniciado el tema, la docente toma una prueba periódica, en la cual todos los semestres aproximadamente el 50% de los estudiantes reprueban.

Cuando la docente asigna ejercicios para que el estudiante los resuelva en casa, el estudiante los resuelve por su cuenta, por lo que si desea cerciorarse de que realizó un trabajo correcto solo puede revisarlo el mismo hasta la siguiente clase.

En algunas ocasiones la resolución de ejercicios se torna algo mecánica, por lo que cuando se cambia la arquitectura del CPU o el formato de instrucción en los ejercicios en clase los estudiantes no saben cómo resolver la situación planteada.

### Antecedentes tecnológicos

Actualmente existen herramientas de ayuda para la comprensión del ciclo de instrucción del procesador, como por ejemplo:

**Microprocessor Tutorial:** es un tutorial en línea en el cual se muestra paso a paso el desarrollo del ciclo de instrucción, utilizando lenguaje ensamblador. Permite observar, de forma general, los pasos que sigue el procesador para ejecutar una instrucción, con ejemplos ya definidos. El tutorial a su vez hace una explicación tanto de este tema, como de otros relacionados, como por ejemplo la estructura del CPU (ALU, unidad de control, registros, bus del sistema), pipelining y arquitecturas de computadora. También cuenta con pequeñas pruebas al finalizar cada tema para asegurar la comprensión de cada uno.

**Pep/7 Simulator:** es un applet en Java que simula el ciclo de instrucción, manejando lógica de instrucciones en ensamblador. Este programa es utilizado para explicar el funcionamiento del procesador, contando con siete registros y cuatro bits de estado (flags: negative, zero, overflow y carry). Entre sus opciones permite ver los registros en binario, decimal y hexadecimal. Se enfoca principalmente en el manejo de registros del procesador y de la memoria principal. Así mismo, cabe recalcar que cuenta con un tutorial en línea disponible para los usuarios.

**Dan! 71 CPU Simulator:** este simulador muestra lo que sucede dentro una CPU durante el ciclo de instrucción. Permite crear, guardar y cargar programas que puede ejecutar. Maneja los datos en lenguaje maquina y está diseñado para ser manejado por usuarios que conocen a profundidad el tema. Muestra el contenido de todos sus registros en binario y permite ejecutar un programa paso por paso, sin mostrar el procedimiento a seguir en cada uno de ellos, o mostrar directamente el resultado. Así mismo, cabe recalcar que el simulador cuenta con un manual en línea disponible para que el usuario aprenda a utilizarlo.

El uso de simuladores aplicados al proceso de enseñanza-aprendizaje del ciclo de instrucción trae consigo una gran cantidad de ventajas, especialmente visuales, pero pone a los estudiantes en un estado pasivo al realizar todo el trabajo por ellos.

## Problema

### Situación Problemática

* Al realizar los esquemas de los registros de procesador y de memoria principal de manera entendible y ordenada se incrementa el tiempo para resolución del problema en sí, así mismo desvía la atención del estudiante, que en vez de enfocarse en la dependencia entre cambios arquitectónicos del procesador y la velocidad de ejecución de un programa, se dedica más a un trabajo mecánico de dibujo de componentes del CPU, de memoria y de sus formas de interconexión.
* Al realizar el estudiante los ejercicios de ciclo de instrucción realizando varios diagramas por separado no puede ver claramente lo que está pasando en el proceso de interacción entre procesador y memoria principal.
* La elaboración manual del proceso conlleva a que el estudiante cometa errores en la transcripción del contenido de los registros entre un diagrama y otro.
* Durante la elaboración de un ejercicio si se comete un error, el resto del flujo de datos e instrucciones es incorrecto y uno se da cuenta de eso solamente en la siguiente clase cuando se hace la revisión del mismo con el docente, lo cual es inadecuado para el proceso de auto-aprendizaje.
* Los simuladores existentes relacionadas al tema están dedicados a una sola arquitectura del CPU y a la vez no son fáciles de utilizar, lo cual exige tiempo extra para su aprendizaje, al mismo tiempo realizan todo el trabajo por el estudiante y disminuye el aprovechamiento deseado. Tal es el caso del Pep/7 Simulator y del Dan! 71 CPU Simulator.
* El Microprocessor Tutorial, es muy simple, y a pesar de ser un tutorial didáctico maneja datos pre-definidos para la explicación. A su vez, no se muestra de manera clara la instrucción que se está ejecutando en el procesador ya que utiliza una notación abreviada para ello, en lugar de utilizar el CO.

### Formulación del Problema

La elaboración manual de los diagramas y las limitaciones conceptuales y técnicas de las herramientas aplicadas al proceso de enseñanza-aprendizaje del ciclo de instrucción del procesador, dificultan1 el proceso de autoaprendizaje2 respecto a la dependencia entre las prestaciones del procesador y su diseño arquitectónico.

## Objetivo

### Objetivo General

Desarrollar una herramienta interactiva para el apoyo al proceso de enseñanza-aprendizaje del ciclo de instrucción del procesador.

1 Hacer difícil una cosa, introduciendo obstáculos o inconvenientes que antes no tenía.

2 es la forma de [aprender](http://definicion.de/aprendizaje/) por uno mismo. Se [trata](http://definicion.de/autoaprendizaje/) de un proceso de adquisición de conocimientos, habilidades, valores y actitudes, que la [persona](http://definicion.de/persona)  realiza por su cuenta ya sea mediante el estudio o la [experiencia](http://definicion.de/autoaprendizaje/).

### Objetivos Específicos y Acciones

**Tabla 1. Objetivos específicos y acciones.**

|  |  |
| --- | --- |
| **Objetivos Específicos** | **Acciones** |
| Diseñar componentes de interfaz que permitan visualizar arquitectura del CPU. | Elegir metodología de desarrollo de software. |
| Elegir arquitectura de desarrollo. |
| Elegir lenguaje y herramientas de programación. |
| Diseñar la estructura de la arquitectura del CPU basada en registros específico. |
| Diseñar la estructura de la arquitectura del CPU basada en registros de uso general. |
| Diseñar estructura de memoria principal y de buses externos del CPU. |
| Desarrollar módulo interactivo para trabajar con lenguaje binario. | Definir datos de entrada. |
| Implementar componente para la definición de formato de instrucción y códigos de operación. |
| Implementar componente para preparación de la RAM para el ejercicio por parte del estudiante. |
| Desarrollar componente de interacción con el estudiante, para desplazar el contenido de registros. |
| Desarrollar componente interactivo de verificación de errores. |
| Desarrollar módulo interactivo para trabajar con lenguaje hexadecimal. | Definir datos de entrada. |
| Implementar selección de arquitectura del CPU basada en registros. |
| Implementar componente para la selección de formato de instrucción y códigos de operación. |
| Implementar componente para preparación de la RAM para el ejercicio por parte del estudiante. |
| **Objetivos Específicos** | **Acciones** |
| Desarrollar módulo interactivo para trabajar con lenguaje hexadecimal. | Desarrollar componente de interacción con el estudiante, para el desplazamiento de contenido de registros. |
| Desarrollar componente interactivo de verificación de errores. |
| Desarrollar módulo interactivo para trabajar con lenguaje ensamblador. | Definir datos de entrada. |
| Implementar selección de arquitectura del CPU basada en registros. |
| Implementar componente para la selección de formato de instrucción y códigos de operación. |
| Implementar componente para preparación de la RAM para el ejercicio por parte del estudiante. |
| Desarrollar componente de interacción con el estudiante, para el desplazamiento de contenido de registros. |
| Desarrollar componente interactivo de verificación de errores. |

Fuente: Elaboración propia 2014.

## Alcances

**Alcances del proyecto:**

* La herramienta será un sistema desarrollado para un entorno web.
* Podrá ser utilizado desde cualquier sistema operativo.
* La realización de las pruebas de la herramienta se llevará a cabo con colaboración de los estudiantes de la UCB que cursaron la materia de Arquitectura de Computadoras como máximo hace dos años y con los que cursaran la misma en el semestre I-2015.
* La herramienta no será diseñada para ningún procesador real, ya que se trabajará solamente sobre los componentes básicos de una CPU para la comprensión del ciclo de instrucción.

**Alcances del producto:**

* El sistema trabajará con direccionamiento en lenguaje binario, hexadecimal y ensamblador.
* Será capaz de manejar arquitectura basada en registros específicos y en registros de uso general.
* Al realizar un ejercicio la herramienta controlará si los datos introducidos y secuencia de micro-operaciones son correctos, generando advertencias de error para hacer posible la detección y corrección de errores por parte del estudiante.
* El estudiante será capaz de definir el estado inicial del CPU y de la memoria principal.
* Visualizar la interacción entre CPU y Memoria Principal.
* Generar un registro de seguimiento de acciones realizadas satisfactoriamente por el estudiante (*log* de operaciones).

## Limitaciones

* La resolución de ejercicios de ciclo de instrucción en lenguaje binario y hexadecimal solo utilizaran formato de instrucción con una dirección en una arquitectura del CPU basada en registros específicos.
* La resolución de ejercicios ciclo de instrucción en lenguaje ensamblador utilizaran formato de instrucción con dos direcciones como máximo en una arquitectura del CPU basada en registros específicos y en registros de uso general.
* La longitud máxima de una instrucción en hexadecimal será de 64 bits y en binario será de 16 bits.
* La longitud del código de operación y de las direcciones será medida en bits y tendrán un valor que sea múltiplo de 4, esto debido a que de esta forma se podrá comparar un ejercicio en binario con uno en hexadecimal.

## Justificación

### Justificación Social

Al utilizar esta herramienta el estudiante no perderá más tiempo al realizar los esquemas y en la transcripción de sus datos lo cual le permitirá concentrarse directamente en la resolución de los ejercicios, al mismo tiempo el estudiante podrá visualizar claramente lo que pasa en el proceso de interacción entre CPU y memoria principal.

La lógica de detección de errores permitirá al estudiante estar consciente de cuando está cometiendo un error o cuando está haciendo un ejercicio de la manera correcta, enriqueciendo el proceso de enseñanza-aprendizaje. Sumado al punto anterior se verá reducido el tiempo de resolución de los ejercicios con lo que el estudiante podrá enfocarse en el objetivo principal del tema.

Con la aplicación de la herramienta en el avance en clases de la materia de Arquitectura de Computadoras, se espera que el estudiante haga un seguimiento con esta herramienta de enseñanza, facilitándose esta labor al encontrarse desarrollado en español y manejando diversas arquitecturas, además de permitir al estudiante no solo ver ejemplos definidos con animaciones a una velocidad adecuada, sino también resolver sus propios ejercicios con verificación en caso de que haya cometido algún error.

### Justificación Técnica

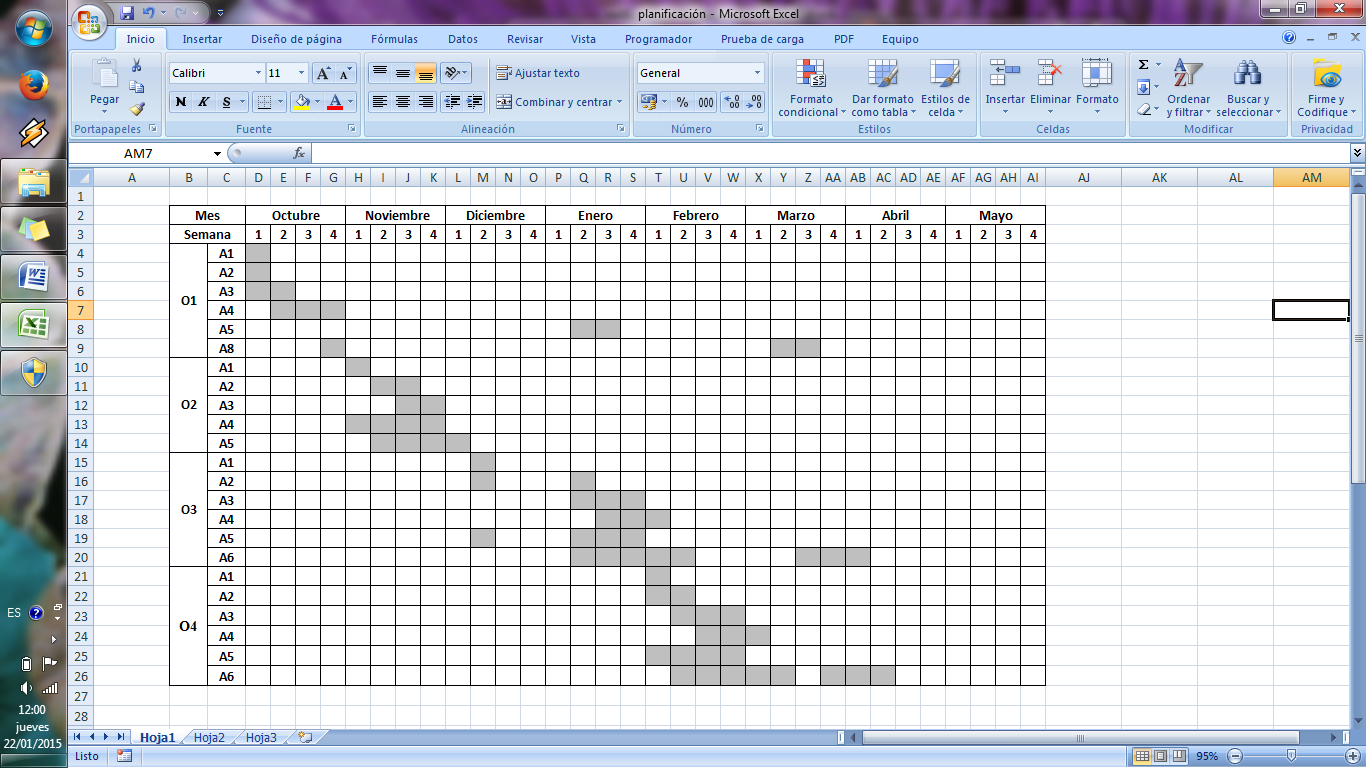
Uno de los principales objetivos de éste proyecto es que el sistema sea intuitivo, mostrando una interfaz de usuario amigable y fácil de entender.

Será una herramienta que le permitirá al estudiante elegir su propio formato de instrucción, así como elegir entre una arquitectura basada en registros específicos y en registros de uso general, por lo que podrá desarrollar sus propios ejercicios, al mismo tiempo de asegurarse que el proceso que está siguiendo es el correcto.

El estudiante no solo definirá la estructura del procesador, sino que también definirá el estado inicial de la memoria principal donde estará guardado el programa que desea ejecutarse.

## Cronograma

**Figura 1. Cronograma.**



Fuente: Elaboración propia 2014.

## Bibliografía

http://www.eastaughs.fsnet.co.uk/cpu/execution-index.html

http://www.rsu.edu/faculty/pmacpherson/programs/pep7.html

http://www.danyey.co.uk/dan71h.php