

PROGRAMACIÓN II

5to Año - IPS



AÑO 2016

- Materia:
 - **Programación II**
- Profesora:
 - **Pamela Viale**
- Horarios:
 - **Lunes de 9:40 a 12:40**
- Contacto:
 - **pamela@fceia.unr.edu.ar**



Teniendo una **idea o problema**, poder diseñar “*algo*” que nos dé una solución.

Algoritmo: secuencia de pasos finita orientada a la consecución de un objetivo.

Una vez que los pensamos/diseñamos:

- ¿Dónde los ejecutamos?
- ¿Cómo los escribimos?
- ¿Cómo los ejecutamos?



- ¿Dónde los ejecutamos?
- ¿Cómo los escribimos?
- ¿Cómo los ejecutamos?

Máquina electrónica, analógica o digital, dotada de una **memoria** de gran capacidad y de métodos de tratamiento de la información, capaz de resolver **problemas** matemáticos y lógicos mediante la utilización automática de programas informáticos. ¿Qué entienden las computadoras?



- ¿Dónde los ejecutamos?
- **¿Cómo los escribimos?**
- ¿Cómo los ejecutamos?

Máquina electrónica, analógica o digital, dotada de una **memoria** de gran capacidad y de métodos de tratamiento de la información, capaz de resolver **problemas** matemáticos y lógicos mediante la utilización automática de programas informáticos. **¿Qué entienden las computadoras?**



- ¿Dónde los ejecutamos?
- **¿Cómo los escribimos?**
- ¿Cómo los ejecutamos?

Código máquina: Conjunto de instrucciones sucesivas directamente interpretables por el hardware.

Al intentar leer, no es posible. Un programador no escribe de ésta manera.

Se obtiene a través de un proceso de compilación y linkeo de código fuente.



- ¿Dónde los ejecutamos?
- ¿Cómo los escribimos?
- ¿Cómo los ejecutamos?

Interpretación de código fuente: se lee un programa escrito en un lenguaje de programación instrucción a instrucción y, para cada una de ellas, efectúa una traducción a las instrucciones de código de máquina equivalentes y las ejecuta inmediatamente.

No todos los lenguajes tienen intérprete.



- ¿Dónde los ejecutamos?
- **¿Cómo los escribimos?**
- ¿Cómo los ejecutamos?

Paradigma: modelo o patrón en cualquier disciplina científica.

Paradigma de programación: representa un estilo de programación en cual se escriben soluciones a problemas en términos de algoritmos.



Paradigmas de Programación:

- Imperativo
- Funcional
- Lógico
- Orientado a Objetos
- Dirigido por Eventos, Concurrentes, etc.



Programación imperativa

Describe la programación en términos del estado del programa y sentencias o instrucciones que cambian dicho estado.

El estado se modifica mediante la asignación de variables.

Ejemplos: ASP, BASIC, C, C++, Fortran, Pascal, Java, Perl y PHP.



Programación Lógica

El objetivo es expresar programas utilizando la lógica matemática.

Se pueden expresar formalmente problemas complejos y resolverlos mediante la aplicación de reglas, hipótesis y teoremas.

Un programa lógico está formado por predicados. Utilizados en el área de inteligencia artificial.

No existe un estado global. Ejemplos: Prolog, Alice.



Programación Orientada a Objetos

El universo del problema se examina en términos de entidades y relaciones entre las entidades.

Las entidades son representadas en el programa como objetos. Los programas se expresan como un conjunto de objetos, que colaboran entre sí para realizar tareas.

Los objetos se agrupan en clases, las cuales definen el comportamiento de los objetos de esa clase.

Fáciles de mantener y reutilizar. Ejemplos: C++, Java.



Programación Funcional

Los programas se construyen a través de la aplicación y composición de funciones.

No existe un estado global.

No existen las asignaciones de variables ni las construcciones estructuradas como la secuencia o la iteración.



Tienen **transparencia referencial**:

“El significado de una expresión depende únicamente de los elementos que la constituyen.”

Están clasificados en puros e impuros.

Impuros: tienen algunos conceptos tomados de los lenguajes imperativos (como las secuencias de instrucciones o la asignación de variables). Ejemplos.: Standard ML, Scala, Ocaml.

Puros: conservan la transparencia referencial. Ejemplo: Haskell, Miranda



Conclusiones sobre los paradigmas

Ningún paradigma es mejor que otro, dependiendo de la situación un paradigma resulta más apropiado que otro.

Existen muchos lenguajes que mezclan características de varios paradigmas.

