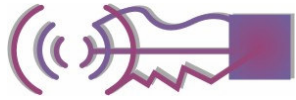




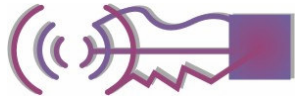
**Los lenguajes de Programación son
el corazón de la Ciencia de la
Computación.**

**Son herramientas que usamos no
solo para comunicarnos con las
máquinas sino también con las
personas.**



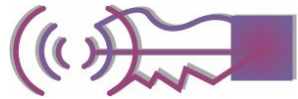
CUAL ES LA IDEA?

- **Introducimos, analizamos y evaluamos** los conceptos más importantes de los lenguajes de programación.
- “el valor de un lenguaje o de un concepto se debe juzgar según la forma en que **afecta la producción** de Software y a la facilidad con la que puede **integrarse** a otras herramientas”



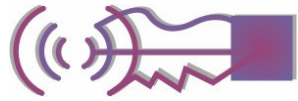
QUÉ CONSEGUIREMOS

- Adquirir habilidad de **apreciar y evaluar** lenguajes, identificando sus **límites y posibilidades**
- Habilidad para **elegir**, para **diseñar, implementar o utilizar** un lenguaje
- Enfatizar la **abstracción**



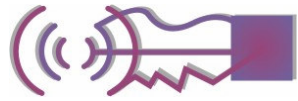
PARA QUÉ ESTUDIAR CONCEPTOS DE LENGUAJES

- Aumentar la capacidad para producir soft.
- Mejorar el uso del lenguaje
- Incrementar el vocabulario
- Elegir mejor un lenguaje
- Facilitar el aprendizaje de nuevos lenguajes
- Facilitar el diseño e implementación de lenguajes



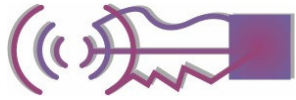
Aumentar la capacidad para producir software

Conocer profundamente las características de los lenguajes permite **aprovechar su potencia y evitar errores, minimizando esfuerzo**



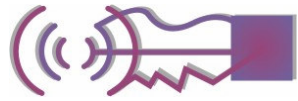
Mejorar el uso del lenguaje que ya se esta utilizando

- Si se entiende como se **implementa cada característica** se mejora la capacidad para escribir programas eficientes.



Incrementar el vocabulario

- El lenguaje
 - equivale a **comunicación**.
 - es una **ayuda** y una **limitación** al pensamiento.
 - permite **expresar y estructurar el pensamiento**
- **Incrementamos la capacidad para expresar ideas**

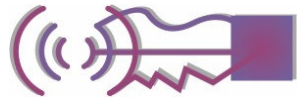


Elegir mejor un lenguaje

- Si se conocen mas a lenguajes, se facilita la elección del mas adecuada de acuerdo al área del problema y por lo tanto se **reduce el esfuerzo de codificación.**

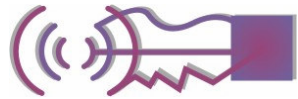
calculo —————→ ADA - Fortran

inteligencia aritificial → Prolog - ML



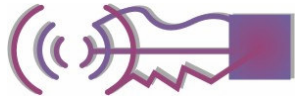
Facilitar el aprendizaje de nuevos lenguajes

- Un lingüista que conoce profundamente la estructura subyacente del lenguaje natural aprende muchísimo mas rápido un nuevo lenguaje (idioma) que una persona que no sabe nada del tema



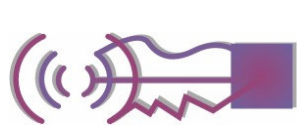
Facilitar el diseño e implementación de lenguajes

- El programador puede convertirse en diseñador o implementador de un lenguaje.



LENGUAJES Y

- LENGUAJES Y PROCESO DE DESARROLLO DE SOFTWARE
- LENGUAJES Y AMBIENTES DE DESARROLLO
- LENGUAJES Y METODOS DE DISEÑO
- LENGUAJES Y LA ARQUITECTURA DE LAS COMPUTADORAS

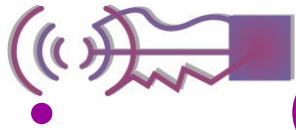


LENGUAJE DE PROGRAMACIÓN EN EL PROCESO DE DESARROLLO DE SOFTWARE

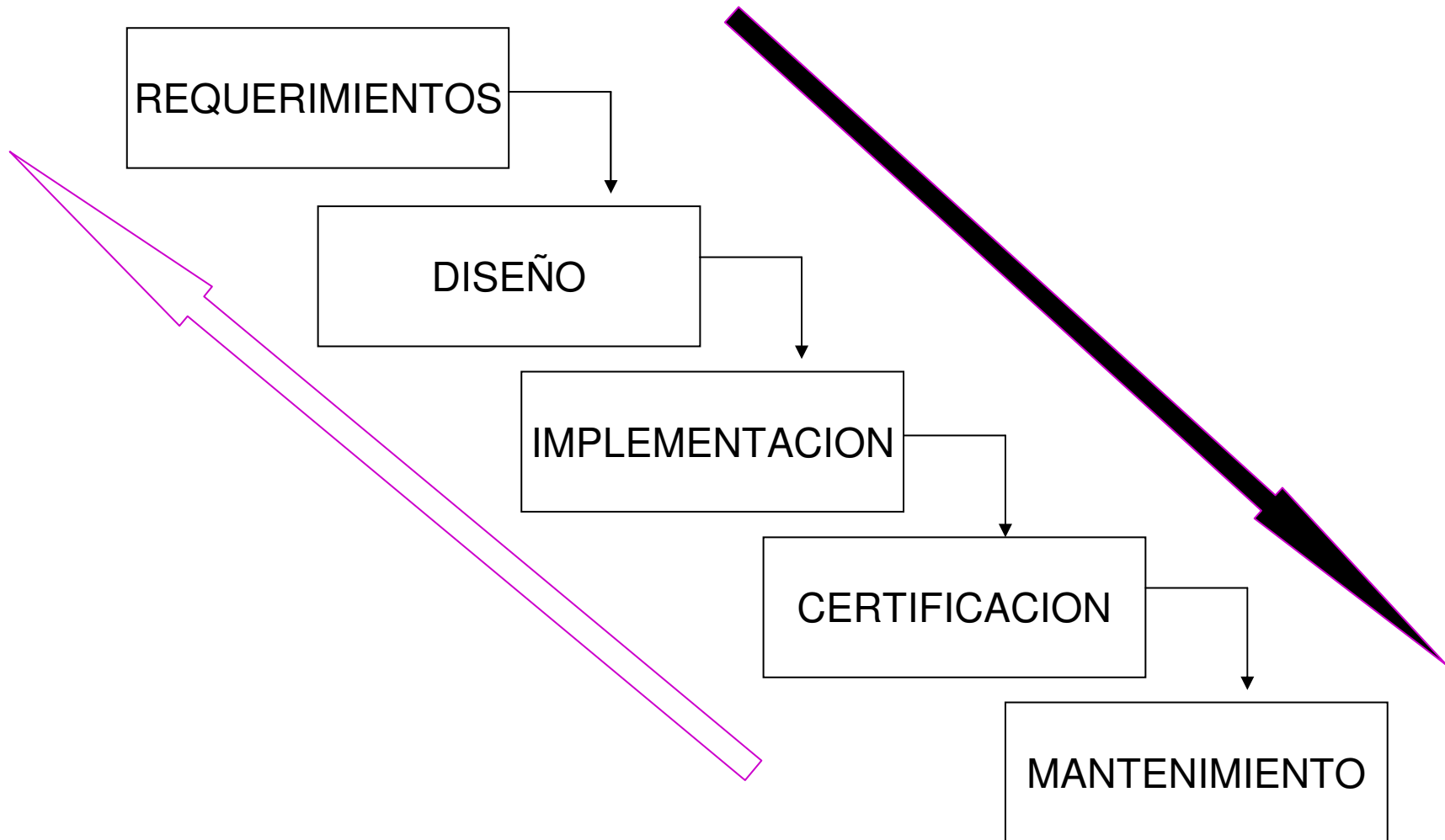
- Los lenguajes de programación son herramientas para producir software.

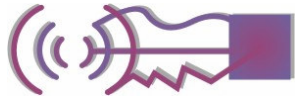
El modelo tradicional de Ciclo de vida del software que debe entregar cada paso al siguiente.

QUE y COMO



CICLO DE VIDA DE SOFTWARE

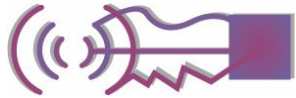




Análisis de requerimientos

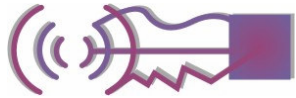
Se identifican y documentan los requerimientos del sistema.

- El resultado de esta etapa es un **documento de especificación** que indica **QUE** problema se debe resolver.



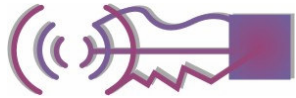
Diseño

- QUE: documento de especificación.
- COMO: **estrategia de solución.**
- El objetivo de esta etapa es una **descomposición en módulos y definición de interfases**
- **El método de diseño elegido impactará en la calidad el entendimiento y la modificabilidad de la aplicación resultante.**



Implementación

- Se codifica en **un lenguaje** el sistema diseñado. Se implementa la estrategia de solución.
- El QUE es ahora el diseño elegido y el COMO es su implementación. Se decide en este paso de que forma entre todas las posibles se codificara la solución diseñada.
- En esta etapa es en la que se utiliza en forma directa el lenguaje.
- El resultado es un sistema totalmente **implementado y documentado**.

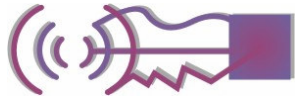


Certificación

(corrección o satisfacción de requerimientos, verificación, validación)

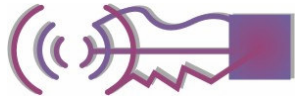
Se evalúa la calidad del producto.

- Los chequeos responden a dos preguntas
 - ¿Construimos el producto correcto? - contra el usuario
 - ¿Construimos el producto correctamente? - contra la especificación.



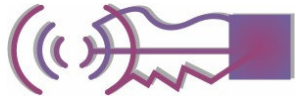
Mantenimiento

- Modificaciones en el sistema terminado:
 - detectar mal funcionamiento
 - modificación de algún requerimiento
- Costo al menos igual a la suma de todos los anteriores.
- En el tiempo será la etapa mas larga.



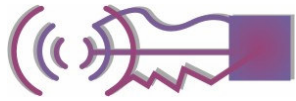
LENGUAJE DE PROGRAMACION Y AMBIENTE DE DESARROLLO

- **Conjunto integrado de herramientas y técnicas** para ser utilizados en todos los pasos del ciclo de vida.
- equipo de trabajo
- los editores, compiladores, simuladores, interpretes, linkeditores y debuggers.
- compatibilidad de herramientas

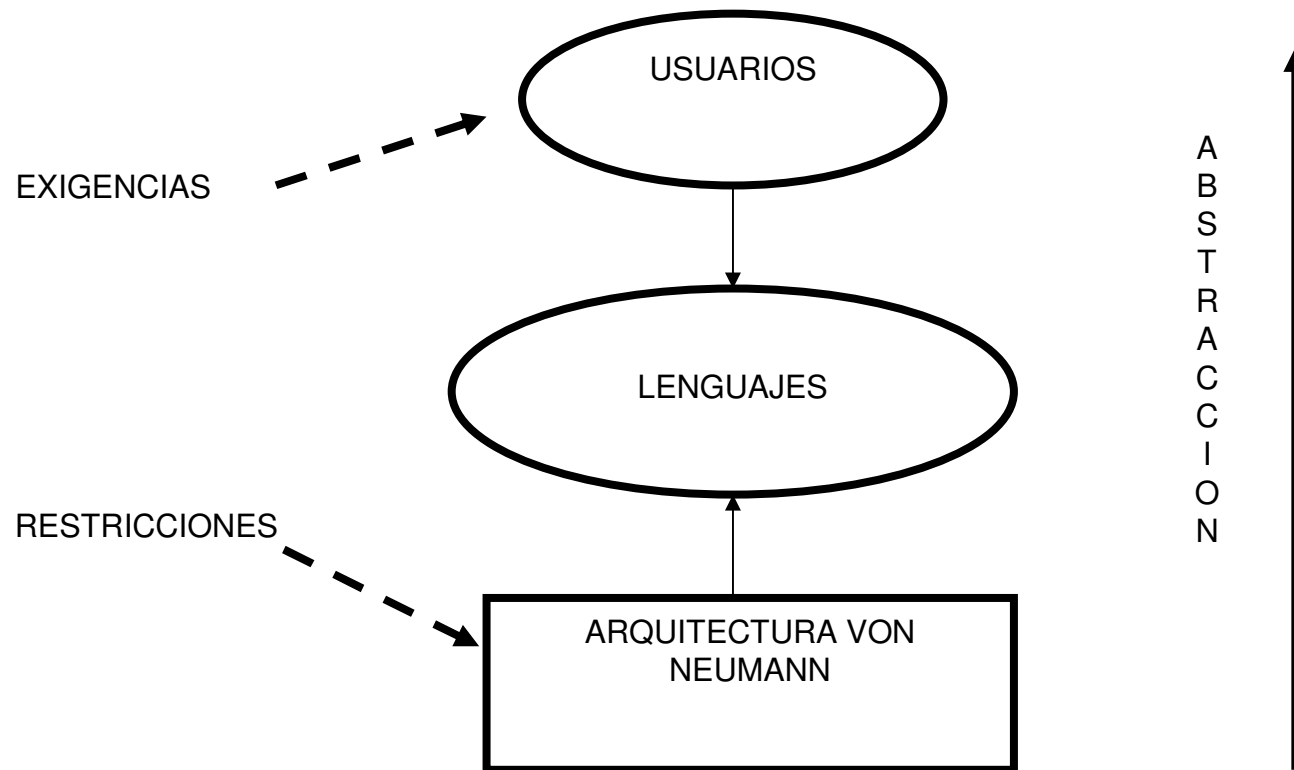


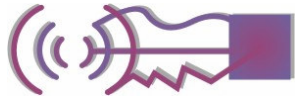
LENGUAJE DE PROGRAMACION Y METODO DE DISEÑO

- Paradigmas de lenguajes
 - Procedural: descomposición en pasos - rutinas
 - Funcional: expresiones y funciones - composición.
 - con TADs: reconocimientos de Tads
 - basada en módulos: grupo de entidades-interfaces
 - OO: clases – instancias - jerarquías
 - Programación genérica: módulos genéricos – instanciados en compilación o ejecución
 - Programación declarativa: no algorítmico, declaración del problema (especificación)

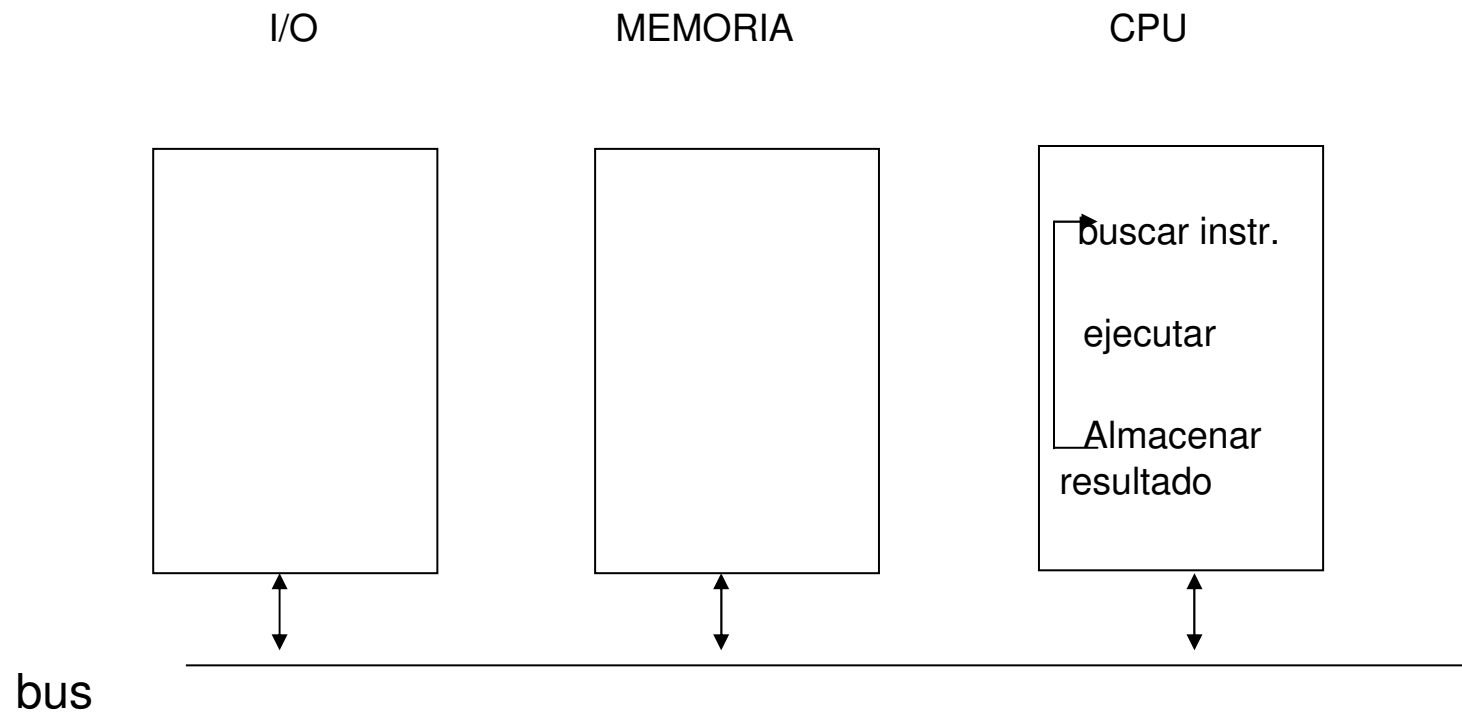


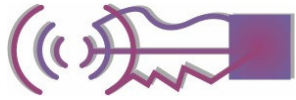
LENGUAJE DE PROGRAMACION Y ARQUITECTURA DE LA COMPUTADORA



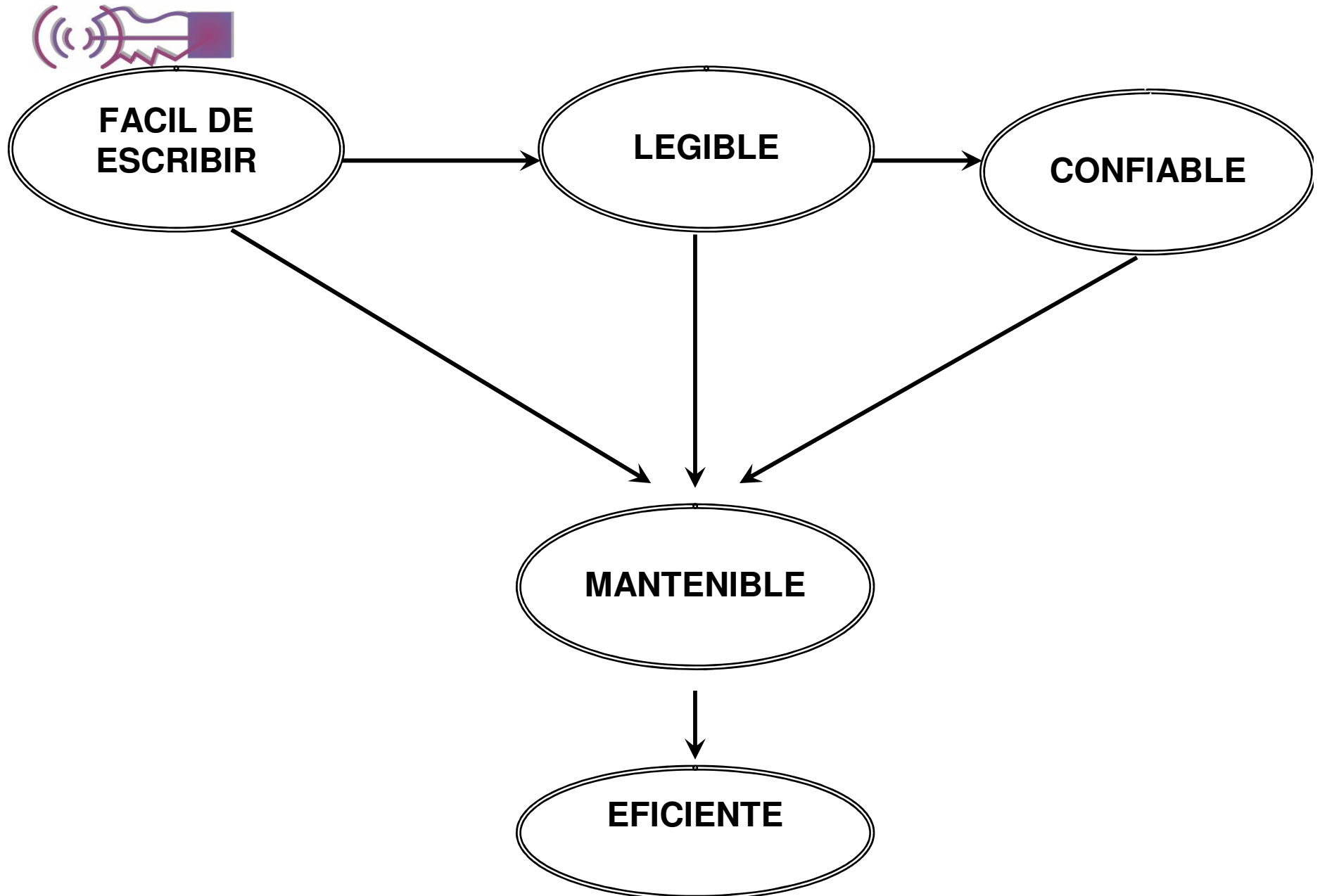


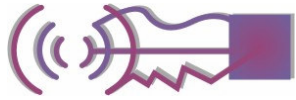
Arquitectura de Von Neumann





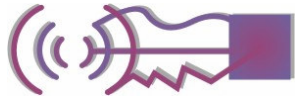
CRITERIOS PARA EVALUAR LOS LENGUAJES DE PROGRAMACION





CRITERIOS DE EVALUACION

- Los lenguajes de programación son herramientas para el desarrollo de software
- Impacto de los conceptos en la metodología de desarrollo de software y en el diseño de las computadoras
- Para ello necesitamos establecer criterios de evaluación



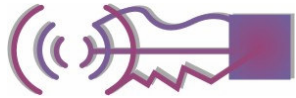
FACILIDAD DE ESCRITURA

- la **facilidad de escritura** es la medida de cuan fácil resulta usar un lenguaje para construir soluciones.

- Simplicidad
- Expresividad
- Ortogonalidad
- Soporte de Abstracción

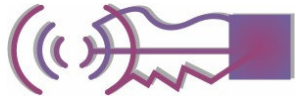


**FACIL DE
ESCRIBIR**



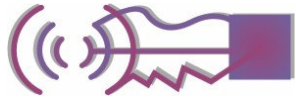
Simplicidad

- las características del lenguaje deben ser fáciles de capturar: entender y recordar
- atentan contra la simplicidad:
 - Muchas componentes elementales
 - Conocer subconjuntos de componentes
 - El mismo concepto semántico – distinta sintaxis
 - Distintos conceptos semánticos - la misma notación sintáctica
 - Abuso de operadores sobrecargados



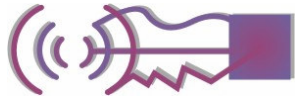
Expresividad

- expresar la naturaleza del problema:
alto/bajo nivel
- concentrar la atención en la resolución del problema
- es función de lo que se desea expresar
- distinto nivel de abstracción



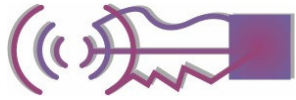
Ortogonalidad

- El efecto de la combinación de características debe entenderse fácilmente y ser predecible
- Proporciona un mayor grado de generalidad sin restricciones ni casos especiales.
- **El usuario comprende mejor si tiene un pequeño número de primitivas y un conjunto consistente de reglas de combinación.**



Soporte de abstracción

- Concepto clave para manejar la complejidad
- Abstracción de procesos y de datos
- Principio de diseño: **Ocultamiento de la información**



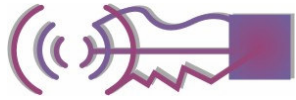
LEGIBILIDAD

- Facil de leer a las personas (equipo)

- SEMANTICA
- SINTAXIS
- DEFINICION
- ESTRUCTURAS DE DATOS
- ESTRUCTURAS DE CONTROL

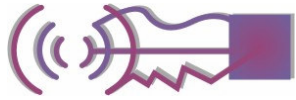


LEGIBILIDAD



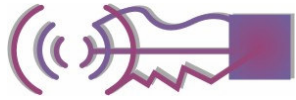
Semántica/Sintaxis

- Restricciones semánticas
- If-then-else anidados
- Separación de sentencias
- indentación



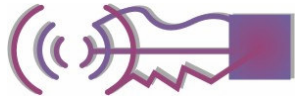
Sintáxis

- Documentación y comentarios
- Elección de nombres
- Uso de constantes
- Convenciones léxicas



Definición del lenguaje

- Precisión en la definición de la sintaxis y de la semántica
- ambigüedad
- Portabilidad
- Definiciones formales

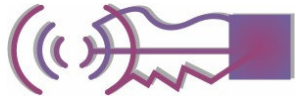


Estructuras de control

- Programación estructurada

Estructuras de datos

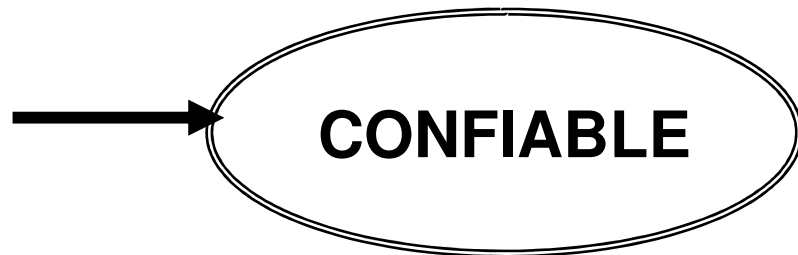
- Facilidades para expresar los datos del problema

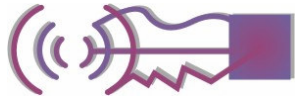


CONFIABILIDAD

- Correcto: cumple con las especificaciones
- Certificación:
- Verificación de programas: probar su corrección

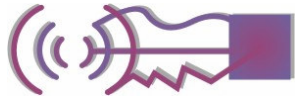
Chequeo de tipos
Restricción de alias
Robusto





Correcto vs confiable

- confiable pero no correcto
x,y: integer
a: char
if $x < 0$ then $y := a$ puede que nunca pase por
el then??
- correcto pero no confiable
las especificaciones no reflejan los
requerimientos



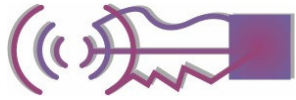
Chequeo de tipos

- Chequeos estáticos y dinámicos

Robusto

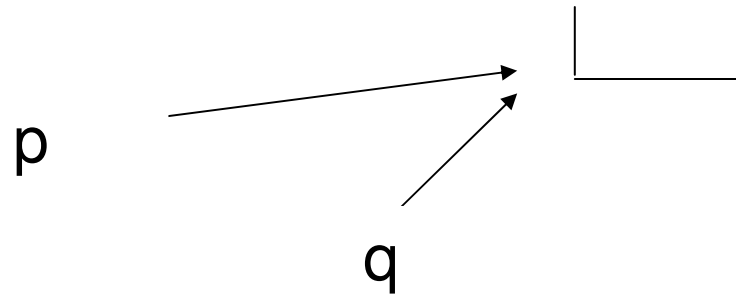
Capacidad de prevenir situaciones

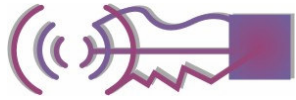
- Manejo de excepciones



Restricción de alias

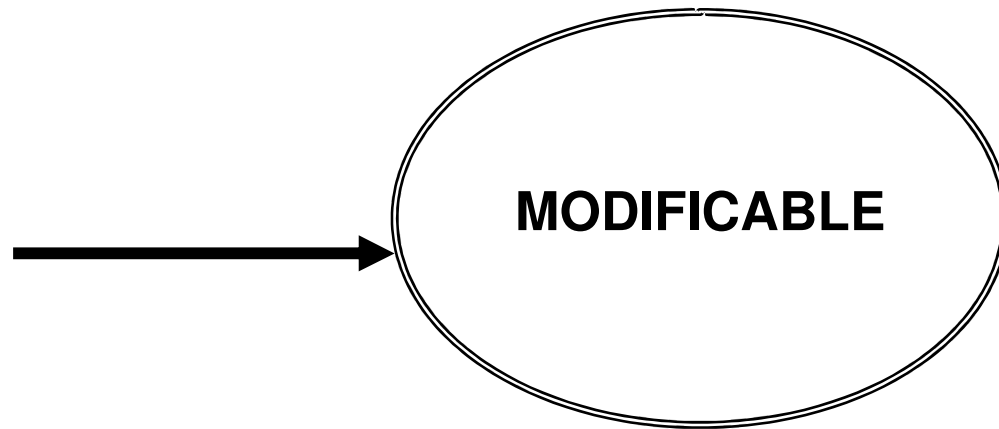
- Dos nombres que referencian la misma posición de memoria

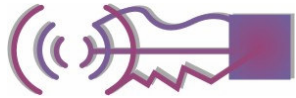




MANTENIBILIDAD

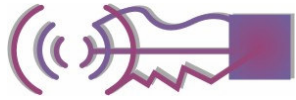
- Modificabilidad
- Factoreo
- Localidad





Mantenible

- Modificabilidad: facilidad de introducir cambios
- Factoreo: una sola vez y luego reusar
- Localidad: el efecto de una característica se restringirse a una porción local del programa



EFICIENCIA

- Tiempo y Espacio
- Esfuerzo humano
- Optimizable



EFICIENCIA