

Introducción

Tecnología de Programación



Adolfo Muñoz – Juan Magallón
Grado en Ingeniería Informática



Universidad
Zaragoza



Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza



Departamento de
Informática e Ingeniería
de Sistemas
Universidad Zaragoza

Programación



Ideas...



¿ Qué características son positivas para un programa ?

¿ En qué orden de prioridad las ponemos ?

Ideas...



¿ Qué características son positivas para un programa ?

¿ En qué orden de prioridad las ponemos ?

La principal, siempre:

- Que resuelva el problema para el que está diseñado en todos los casos.

Muchas de las características deseables tienen que ver con el tiempo (dependiendo del contexto):

- Tiempo de ejecución.
- Tiempo de compilación.
- Tiempo de uso hasta conseguir una tarea determinada (usabilidad).



Muchas de las características deseables tienen que ver con el tiempo (dependiendo del contexto):

- Tiempo de ejecución.
- Tiempo de compilación.
- Tiempo de uso hasta conseguir una tarea determinada (usabilidad).
- **Tiempo de programación:** ¿Cuánto tiempo cuesta al programador llegar a implementar la solución?
- **Mantenibilidad:** ¿Cuánto tiempo cuesta hacer modificaciones y/o corregir errores (*bugs*)?



En esta asignatura nos centraremos en tres características:

- Que el programa resuelva el problema para el que está diseñado.
- Que el tiempo de programador sea mínimo.
- Que el programa resultante sea legible / mantenible / escalable.



Principios



Diferentes desarrolladores han promocionado ciertos

principios del desarrollo de software

que son agnósticos a las tecnologías subyacentes.

Presentamos algunos de ellos.



KISS

"Keep it Simple, Stupid"



KISS

"Keep it Simple, Stupid"

El principio KISS establece que la mayoría de sistemas funcionan mejor si se mantienen simples que si se hacen complejos; por ello, la simplicidad debe ser mantenida como un objetivo clave del diseño, y cualquier complejidad innecesaria debe ser evitada.



***"Premature optimization
is the root of all evil."***

Donald Knuth

DRY

"Don't Repeat Yourself"



DRY

"Don't Repeat Yourself"

Según este principio toda "pieza de información" nunca debería ser duplicada debido a que la duplicación incrementa la dificultad en los cambios y evolución posterior, puede perjudicar la claridad y crear un espacio para posibles inconsistencias.

Los términos "pieza de información" son usados en un sentido amplio, abarcando:

- datos almacenados en una base de datos
- código fuente de un programa de software
- información textual o documentación



SRP

Single-responsibility principle



SRP

Single-responsibility principle

Cada módulo o clase debe tener responsabilidad sobre una sola parte de la funcionalidad proporcionada por el software y esta responsabilidad debe estar encapsulada en su totalidad por la clase. Todos sus servicios deben estar estrechamente alineados con esa responsabilidad.

Cada elemento del programa debería tener sólo una razón para cambiar.



YAGNI

"You Aren't Going to Need It"



YAGNI

"You Aren't Going to Need It"

No agregar nunca una funcionalidad hasta que no es necesaria:

- Se sacrifica tiempo que se destinaría para la funcionalidad básica.
- Nuevas características deben ser depuradas, documentadas y soportadas.
- Impone límites a lo que podría ser hecho en el futuro.



Refactorización en mantenimiento



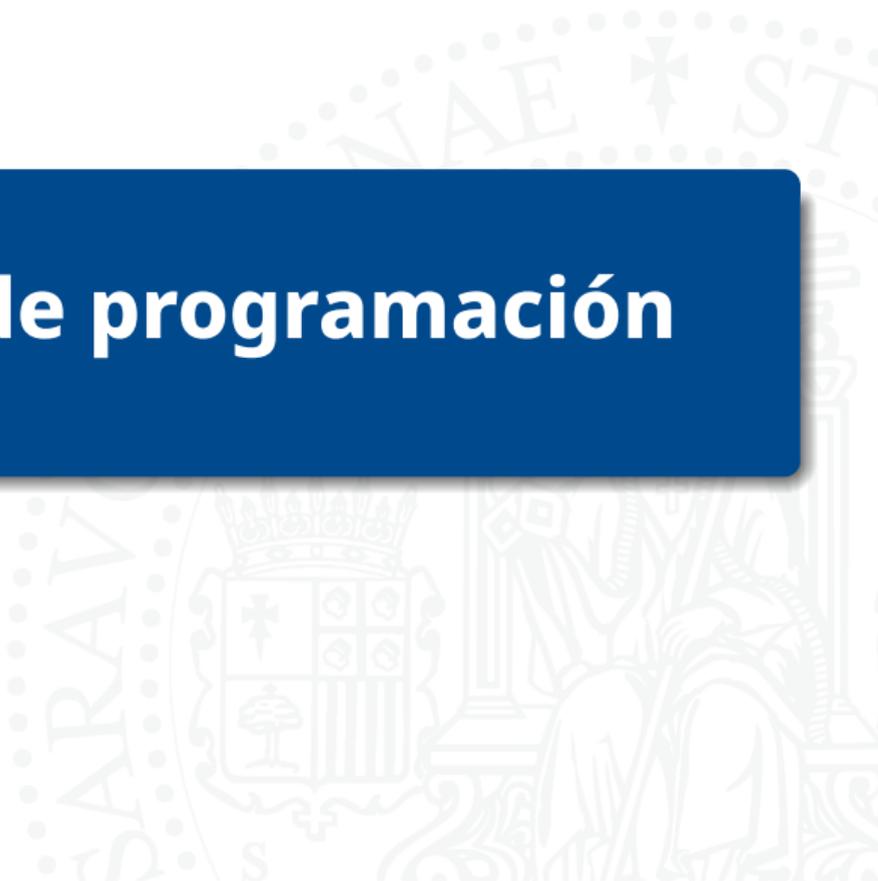
Refactorización en mantenimiento

El código raramente es perfecto la primera vez. Cada vez se aprenden más técnicas de programación y más sobre cada problema en cuestión.

Al **depurar (*debugging*)** se debe hacer también **refactorización (*refactoring*)** iterativo para asegurar no sólo que el código sea correcto sino también mantenible a largo plazo.



Lenguajes de programación

The background features a large, faint watermark of the University of Sarajevo seal. The seal is circular and contains the text 'SARAJEVO' at the top and 'UNIVERSITY OF SARAJEVO' at the bottom. In the center, there is a coat of arms with a crown on top, a cross on the left, and a shield with various symbols on the right. The seal is surrounded by a decorative border of dots.

Pregunta



Dado un problema que se resuelve con un programa...

¿ qué lenguaje de programación usarías para escribirlo ?



Pregunta



Dado un problema que se resuelve con un programa...

¿ qué lenguaje de programación usarías para escribirlo ?

Depende del problema: Elige el lenguaje óptimo para cada problema.



Ideas...



¿ Qué criterios puedes seguir para elegir el lenguaje *adecuado* para un problema concreto ?

TIOBE

Jan 2023	Jan 2022	Change	Programming Language	Ratings	Change
1	1		 Python	16.36%	+2.78%
2	2		 C	16.26%	+3.82%
3	4	▲	 C++	12.91%	+4.62%
4	3	▼	 Java	12.21%	+1.55%
5	5		 C#	5.73%	+0.05%
6	6		 Visual Basic	4.64%	-0.10%
7	7		 JavaScript	2.87%	+0.78%
8	9	▲	 SQL	2.50%	+0.70%
9	8	▼	 Assembly language	1.60%	-0.25%
10	11	▲	 PHP	1.39%	-0.00%

PYPL

Rank	Change	Language	Share	Trend
1		Python	27.93 %	-0.9 %
2		Java	16.78 %	-1.3 %
3		JavaScript	9.63 %	+0.5 %
4	↑	C#	6.99 %	-0.3 %
5	↓	C/C++	6.9 %	-0.5 %
6		PHP	5.29 %	-0.8 %
7		R	4.03 %	-0.2 %
8	↑↑↑	TypeScript	2.79 %	+1.0 %
9		Swift	2.23 %	+0.3 %
10	↓↓	Objective-C	2.2 %	-0.1 %



GitHub

# Ranking	Programming Language	Percentage (YoY Change)	YoY Trend
1	Python	17.916% (-0.128%)	
2	Java	11.281% (-1.008%)	^
3	JavaScript	9.875% (-4.276%)	v
4	C++	9.704% (+2.990%)	^
5	Go	9.435% (+1.220%)	
6	TypeScript	8.307% (-0.222%)	v
7	PHP	5.270% (-0.017%)	^
8	Ruby	4.636% (-1.570%)	v
9	C	4.241% (+1.070%)	^
10	C#	3.270% (-0.124%)	v



***"The most important thing in
the programming language is
the name.***

***A language will not succeed
without a good name.
I have recently invented a very
good name and now I am
looking for a suitable
language."***

Donald Knuth

	TIOBE 2023	PYPL 2023	GitHut 2022
C++	3	5	4
Java	4	2	2
Haskell	45	23	27



Ideas:

- Elegir un lenguaje por su popularidad es un error.



Ideas:

- Elegir un lenguaje por su popularidad es un error.
- Cuantos más lenguajes de programación se conocen, más sencillo es aprender uno nuevo.



Ideas:

- Elegir un lenguaje por su popularidad es un error.
- Cuantos más lenguajes de programación se conocen, más sencillo es aprender uno nuevo.
- Lo importante es aprender **conceptos** de programación, los lenguajes vienen después.



Ideas:

- Elegir un lenguaje por su popularidad es un error.
- Cuantos más lenguajes de programación se conocen, más sencillo es aprender uno nuevo.
- Lo importante es aprender **conceptos** de programación, los lenguajes vienen después.
- Elegir el lenguaje adecuado para la tarea adecuada (por ejemplo, enseñar).





***"Instead of imagining that
our main task is to instruct a
computer what to do,
let us concentrate rather on
explaining to human beings
what we want
a computer to do."***

Donald Knuth

Introducción

Tecnología de Programación



Adolfo Muñoz – Juan Magallón
Grado en Ingeniería Informática



Universidad
Zaragoza



Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza



Departamento de
Informática e Ingeniería
de Sistemas
Universidad Zaragoza