

Introducción a la programación III

Martín San José de Vicente

Director de Contenidos de Imagina Formación

"El presente documento es para uso exclusivo de los alumnos inscritos al curso AGILE MANAGEMENT de EDEM y en todo caso dentro del entorno del Campus y/o de cualesquiera otras herramientas que EDEM le haya facilitado acceso para el desarrollo del curso, por lo que no se permite la descarga del mismo. Asimismo, queda prohibida la difusión, distribución o divulgación de la presente grabación y/o de cualesquiera otras y/o documentos facilitados en el desarrollo del curso y, particularmente, su difusión a través de redes sociales, plataformas de vídeo, así como entornos web cuya finalidad sea la de compartir vídeos y/o documentos formativos (apuntes, etc.). El incumplimiento de esta prohibición generará las correspondientes responsabilidades a exigir por EDEM y/o por el profesor titular de la propiedad intelectual de los documentos formativos, así como de los titulares de los datos de carácter personal cuya imagen/voz y/u otros datos aparezcan en los mismos."

OBJETIVO

En estas diapositivas vamos a analizar cómo las bases de la programación actual heredan de descubrimientos de hace décadas.

Además, comentaremos y debatiremos acerca de la importancia de escoger correctamente el lenguaje de programación con el que desarrollar.

ÍNDICE

- Los 3 paradigmas de programación
- Análisis y comparativa de los lenguajes de programación más potentes en la actualidad.
- La importancia de escoger correctamente.

Los 3 Paradigmas de la Programación

Los 3 paradigmas que heredamos hace décadas y aún siguen vigentes.

> Los 3 Paradigmas de la Programación

Algo de historia

A pesar de que a lo largo de los años los avances en el mundo de la informática han sido espectaculares, lo cierto es que **la esencia en los lenguajes no ha cambiado tanto desde los años 40.**

> Los 3 Paradigmas de la Programación

Algo de historia

1938

Alan Turing establece las **bases** de lo que hoy entendemos como la programación **informática**.

Turing no fue el primero en crear una máquina programable, pero sí que **fue el primero en comprender que los programas** eran y son, simplemente, **información**.

> Los 3 Paradigmas de la Programación

Algo de historia

1945

Turing **ya escribía programas que un desarrollador actual podría reconocer**. Programas que usaban **bucles, pilas, subrutinas, asignaciones, ramas** y otras estructuras familiares. El lenguaje de Turing era **binario**.

Desde aquella época ha habido diversas revoluciones en el mundo de la programación.

> Los 3 Paradigmas de la Programación

Algo de historia

1949

La **primera revolución** fue a **finales de los 40**, con la llegada de los **ensambladores**.

Estos “lenguajes” evitaron que los programadores tuvieran que estar traduciendo sus programas a binario.

> Los 3 Paradigmas de la Programación

Algo de historia

1951

Se inventa el **primer compilador**.

Su nombre era **A0** y su creador fue **Grace Hopper**. Es más, él fue el que acuñó la palabra compilador.

> Los 3 Paradigmas de la Programación

Algo de historia

1953

Se crea el lenguaje de programación **Fortran**.

Inspiración para que más tarde llegaran en tromba el resto que conocemos a día de hoy: **COBOL**, **PL/1**, **SNOBOL**, **C**, **Pascal**, **C++**, **Java** y así hasta **Javascript** y demás lenguajes interpretados actuales.

> Los 3 Paradigmas de la Programación

Los paradigmas en el mundo de la programación

Una de las últimas revoluciones en el mundo de la programación (sin contar el Machine Learning y las redes neuronales) llegó con los **paradigmas**.

Los paradigmas son **formas de programación, relativamente poco relacionadas con los lenguajes de programación.**

> Los 3 Paradigmas de la Programación

Los paradigmas en el mundo de la programación

Un paradigma **se encarga de indicar qué estructuras de programación utilizar y cuándo utilizarlas.**

Hasta el día de hoy ha habido **3 paradigmas principales** y es probable que no haya más.

> Los 3 Paradigmas de la Programación

Los 3 grandes pilares

Los 3 paradigmas que llevan existiendo décadas son:

- > Programación **Estructurada**
- > Programación **Orientada a Objetos**
- > Programación **Funcional**

> Los 3 Paradigmas de la Programación

1. Programación Estructurada

1968

Edsger Wybe Dijkstra descubre que las instrucciones **goto** (servían para saltar, sin límites, de un bloque a otro dentro de un programa) **eran dañinas para la estructura de los programas.**

Cambió los saltos por las operaciones condicionales y bucles (**IF THEN, ELSE, DO WHILE** o **UNTIL**).

> Los 3 Paradigmas de la Programación

1. Programación Estructurada

Este fue **uno de los mayores impactos** que ha tenido la programación **en la historia**.

Es más, **actualmente todos los desarrolladores somos desarrolladores estructurados** gracias a que Dijkstra consiguió ganar la batalla intelectual de su época. Lo que vino después fue aún más importante

Se empezó a poder **estructurar un programa** y permitió **descomponer recurrentemente** en funciones.

> Los 3 Paradigmas de la Programación

2. Programación Orientada a Objetos

1966

Dos años antes que el paradigma anterior, los científicos **Ole Johan Dahl** y **Kristen Nygaard** descubren cómo **encapsular variables y funciones** en algo que se llamaría con el tiempo **clases**.

Esto más tarde se tradujo en **herencia, polimorfismo y encapsulación**. Tres conceptos clave de todo lenguaje moderno y que veremos a lo largo del curso.

> Los 3 Paradigmas de la Programación

3. Programación Funcional

1958

Años antes de los dos paradigmas, **John McCarthy** inventó el lenguaje **LISP** que estaba basado en el concepto de cálculo **Lambda** del matemático **Alonzo Church**.

El descubrimiento de la posibilidad de usar la **inmutabilidad** y la **ausencia de asignación**, hicieron posible la aparición de muchos lenguajes (Python, entre ellos) y herramientas de control de versiones como GIT, SVN, etc.

Análisis y comparativa de lenguajes de programación

¿Cuáles son los lenguajes de programación más potentes en la actualidad?

> **Análisis y comparativa** de lenguajes de programación

¿Cómo saber qué lenguaje se debe escoger?

Escribir y leer esta sección hace unos años habría sido una experiencia totalmente distinta.

La evolución de los lenguajes de programación ha sido vertiginosa durante los últimos años.

Los lenguajes de programación funcionan como productos y por lo tanto, existe una oferta y demanda que con el paso de los años es cada vez más competitiva.

> **Análisis y comparativa** de lenguajes de programación

¿Cómo saber qué lenguaje se debe escoger?

Existen comunidades de desarrollo para los lenguajes que hacen que se mantengan vivos en el tiempo.

Cuando hablamos de un lenguaje vivo, queremos decir que el lenguaje evoluciona con el tiempo. Es decir, que cada cierto tiempo (semanas, meses, años, etc.) se crean nuevas versiones del mismo aportando novedades o resolviendo problemas.

Al escoger un lenguaje tenemos que buscar **que no quede obsoleto** y que la comunidad sea **grande y activa**.

> **Análisis y comparativa** de lenguajes de programación

¿Cómo saber qué lenguaje se debe escoger?

Este concepto de tener una comunidad grande y activa es esencial, pues querrá decir que cuando desarrollemos en ese lenguaje, tendremos mayores facilidades para encontrar soluciones a errores que podamos encontrarnos por el camino.

En el mundo de la programación es bien conocido el **índice Tiobe** que trata de ilustrar cuáles son los lenguajes con mayor popularidad.

> Análisis y comparativa de lenguajes de programación

¿Cómo saber qué lenguaje se debe escoger?

Si nos guiásemos por este índice y su última actualización, encontraríamos que el lenguaje más popular es **C**, seguido de **Java**, **Python**, **C++** y **C#**:

Jun 2020	Jun 2019	Change	Programming Language	Ratings	Change
1	2	⬆	C	17.19%	+3.89%
2	1	⬇	Java	16.10%	+1.10%
3	3		Python	8.36%	-0.16%
4	4		C++	5.95%	-1.43%
5	6	⬆	C#	4.73%	+0.24%
6	5	⬇	Visual Basic	4.69%	+0.07%
7	7		JavaScript	2.27%	-0.44%
8	8		PHP	2.26%	-0.30%
9	22	⬆	R	2.19%	+1.27%
10	9	⬇	SQL	1.73%	-0.50%

> **Análisis y comparativa** de lenguajes de programación

¿Cómo saber qué lenguaje se debe escoger?

Pero **este índice comete un error** realizar estas conclusiones, ya que cuenta el número de **webs que contengan el nombre del idioma** como resumen del interés generado por parte de la comunidad de desarrolladores.

Por ejemplo, el lenguaje obsoleto de Apple **Objective-C** cuenta con cerca de 20 millones de webs, mientras que otros lenguajes muy importantes como **C** tienen únicamente 11 millones. Explicando así por qué **lenguajes que a penas despiertan interés se colocan en tan alta posición del ranking Tiobe.**

> **Análisis y comparativa** de lenguajes de programación

¿Cómo saber qué lenguaje se debe escoger?

En cambio, el **ranking PYPL** se centra en la **cuota de mercado de cada lenguaje** y es bastante más justo a la hora de definir el interés de la comunidad.

Se basan en la clave de búsqueda **tutorial** dentro de **Google Trends** y apoyado en las estadísticas de **StackOverflow** permitiendo así ajustar mucho más lo que la comunidad demanda.

Como desventaja, **el índice PYPL está limitado a 28 lenguajes.**

> **Análisis y comparativa** de lenguajes de programación

¿Cómo saber qué lenguaje se debe escoger?

Si nos basamos en **PYPL**, veremos que **Python** es el **lenguaje con más share del mercado**:

Worldwide, Jun 2020 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	31.6 %	+4.3 %
2		Java	17.67 %	-2.4 %
3		Javascript	8.02 %	-0.2 %
4		C#	6.87 %	-0.4 %
5		PHP	6.02 %	-0.9 %
6		C/C++	5.69 %	-0.2 %
7		R	3.86 %	-0.1 %
8		Objective-C	2.5 %	-0.3 %
9		Swift	2.24 %	-0.1 %
10	↑	TypeScript	1.86 %	+0.2 %

> **Análisis y comparativa** de lenguajes de programación

¿Cómo saber qué lenguaje se debe escoger?

A parte de contar con un gran share, hay ciertos valores positivos con los que un lenguaje debe contar:

- > Que sea **multiparadigma**
- > Que sea **legible, flexible y simple** para agilizar el desarrollo, además de mantenerlo durante el tiempo.
- > Que sea de **propósito general**, es decir que valga para crear aplicaciones y programas de todo tipo.
- > Que sea **multiplataforma**
- > Que sea **código abierto** y que la comunidad pueda participar para mejorar y mantener el lenguaje.
- > Que sea **seguro**

> **Análisis y comparativa** de lenguajes de programación

¿Cómo saber qué lenguaje se debe escoger?

Por ejemplo, veamos algunas características de los lenguajes principales a día de hoy:

- **Java:** Es uno de los lenguajes más utilizado en el mundo y hasta hace pocos años era el lenguaje más importante dentro de la educación universitaria. Destacan las siguientes características:
 - Legible y simple
 - Multiparadigma
 - Multiplataforma
 - Propósito general
 - Código Abierto
 - Alto Nivel

> **Análisis y comparativa** de lenguajes de programación

¿Cómo saber qué lenguaje se debe escoger?

- **C:** Es uno de los lenguajes más famosos que existen. No es de tan alto nivel como otros, pero es el más empleado y apreciado por desarrolladores de sistemas operativos. Es un lenguaje robusto, pero mucho más complejo:
 - Es de propósito general
 - Multiplataforma
 - Seguro
 - Medio Nivel

> **Análisis y comparativa** de lenguajes de programación

¿Cómo saber qué lenguaje se debe escoger?

- **C ++:** Fue una continuación y ampliación del lenguaje C que incorporaba el paradigma de Orientación a Objetos:
 - Es de propósito general
 - Multiplataforma
 - Multiparadigma
 - Legible, pero su sintaxis está netamente heredada del lenguaje C.
 - Seguro
 - Alto Nivel

> **Análisis y comparativa** de lenguajes de programación

¿Cómo saber qué lenguaje se debe escoger?

- **C#:** Es uno de los lenguajes modernos que está teniendo más éxito gracias a las nuevas tecnologías estandarizadas de Microsoft :
 - Es de propósito general
 - Multiplataforma
 - Multiparadigma
 - Legible, sencillo y flexible
 - Seguro
 - Alto Nivel

> **Análisis y comparativa** de lenguajes de programación

¿Cómo saber qué lenguaje se debe escoger?

- **Python:** Es el lenguaje moderno que más importancia ha cobrado en los últimos años. Ha superado a otros grandes lenguajes asentados gracias a sus características:
 - Es de propósito general
 - Legible, muy flexible y similar al idioma inglés en su sintaxis.
 - Multiparadigma
 - Multiplataforma
 - Seguro
 - Alto Nivel

> **Análisis y comparativa** de lenguajes de programación

¿Cómo saber qué lenguaje se debe escoger?

En este curso hemos optado por aprender a programar en Python debido a que es un lenguaje que cumple con todo lo que se debe valorar positivamente en este contexto.

Tiene una comunidad de desarrolladores enormes y se trata de un lenguaje en constante evolución.

Didácticamente hablando, Python es una opción fantástica que permite sumergirse en el mundo del desarrollo y plantea de forma sencilla, si fuera necesario, el paso a otros nuevos lenguajes.

A large blue geometric shape, resembling a stylized 'L' or a corner, occupies the left side of the slide. It has a diagonal cut at the top right corner.

La importancia de escoger correctamente

Decidir y decidir correctamente no es lo mismo.

> La importancia de escoger bien

La toma de decisiones

A lo largo de la creación de un programa, se toman constantemente nuevas decisiones que suponen un nuevo camino de desarrollo.

Sin lugar a dudas, la decisión más importante es la definición de la **Arquitectura** inicial.

> La importancia de escoger bien

La toma de decisiones

Entre las muchas decisiones que hay que tomar a la hora de decidirse a crear software, **escoger el lenguaje de programación es una de las vitales.**

Decidirse por una opción u otra puede ser relativamente sencillo si no pensamos a largo plazo o en el mantenimiento y escalabilidad de nuestros proyectos.

Una decisión poco madurada ahora, puede convertirse en un verdadero dolor de cabeza en un futuro (ya sea cercano o lejano).

> La importancia de escoger bien

La toma de decisiones

A día de hoy **existen muchos lenguajes de programación de alto nivel** muy potentes y **puede parecer indiferente escoger** entre unos u otros, pero eso está **muy lejos de la realidad**.



> La importancia de escoger bien

La toma de decisiones

Vuelve a observar los datos de **Popularity Programming Language**.

<https://pypl.github.io/PYPL.html>

> La importancia de escoger bien

La toma de decisiones

Como comentamos antes, esta web se encarga de realizar comparaciones de **share de mercado** de los distingos lenguajes de programación que existen.

Al igual que en el mercado de valores, el **interés que generan los lenguajes de programación sube y baja** a lo largo de los años.

> La importancia de escoger bien

La toma de decisiones

Esto tiene unas implicaciones enormes en los desarrollos del presente y del futuro.

Lenguajes que **hace años** parecían **robustos** e **ideales**, **actualmente** se ven como **obsoletos y sin a penas comunidad activa**.

El mundo de la banca es un claro ejemplo de apostar todos los desarrollos a lenguajes que actualmente no se encuentran si quiera entre los 20 primeros.

> La importancia de escoger bien

La toma de decisiones

Ese anclaje en el tiempo ha hecho que muchos equipos hayan tenido que realizar un grandísimo esfuerzo para adaptarse a lenguajes más modernos y aprovechar todas sus ventajas.

Incluso apostar por un lenguaje que no perdure en el tiempo, puede suponer que el número de desarrolladores que conozcan ese lenguaje se vea cada vez más reducido y el coste de obtener un buen equipo sea cada vez mayor.

> La importancia de escoger bien

Python

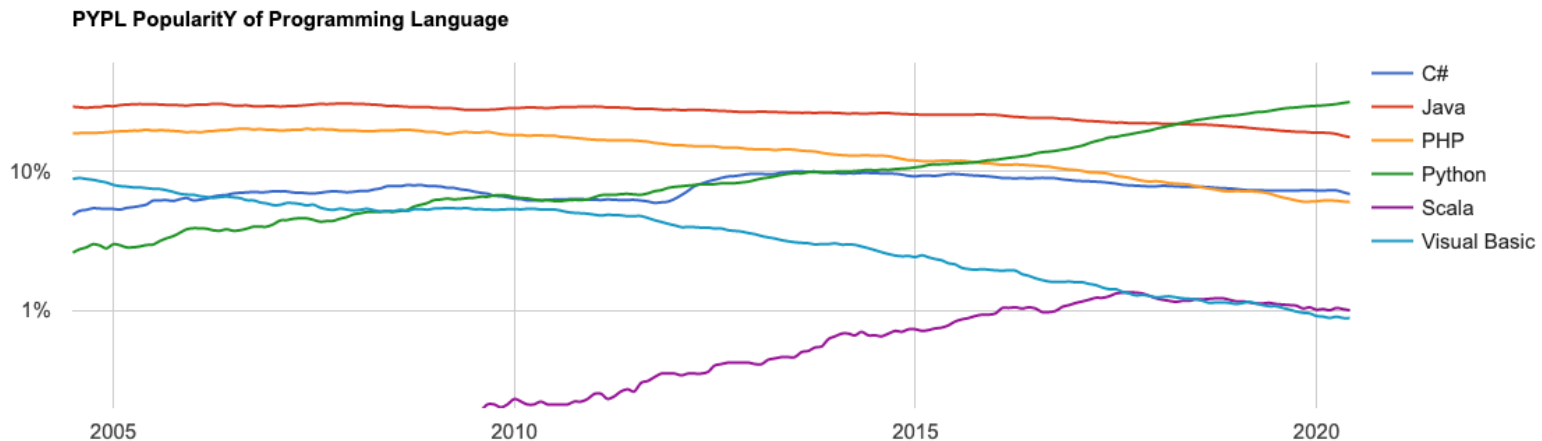
A día de hoy, Python es el **lenguaje de programación** con mayor **share de mercado**.

Se debe especialmente a que es un lenguaje moderno flexible y en el cuál la comunidad ha apostado por el desarrollo de modelos de **machine learning y deep learning**, los dos grandes mundos de la programación presente y futura.

> La importancia de escoger bien

Python

Python ha ido ganando interés y popularidad a medida que han ido pasando los años, superando a grande lenguajes asentados en la parte superior de la tabla durante años:



> La importancia de escoger bien

Lenguajes que ya no son lo que eran

Hay muchos lenguajes que fueron imprescindibles hace años y que a día de hoy han quedado muy atrás frente a opciones como Python:

