Tecnicatura universitaria en programación (TUP)

Metodología de la investigación

**Historia y evolución de la programación**

**Docentes:**

Lupani, María Cecilia

Palermo, Lorena

Spini, Natalia

**Alumnos:**

Chaves, Enzo Julián Leonel – Leg. 27993

Olguin, Alejo – Leg. 28106

Tola, Lucas – Leg. 27901

Acosta, Ramon Agustín – Leg. 28959

Scollo, Sebastián Ezequiel – Leg. 27942

Santiago Jorge Aguirre – Leg. 26910



**ABSTRACT**

En lo que a informática y computación se refiere, la programación desempeña un papel fundamental y de constante transformación. En este informe, nos adentraremos en la increíble historia y evolución de la programación, explorando los momentos clave, personas destacadas y visionarias que con sus innovaciones han dado forma a esta disciplina tan fundamental en la sociedad digital de hoy en día.

El objetivo de este trabajo es, por un lado, poder concretar un recorrido a lo largo de la historia de la programación, desde el telar mecánico diseñado por Joseph Marie Jacquard hasta la programación moderna de hoy en día. También, por el otro lado, buscaremos proporcionar una visión profunda de como la programación ha evolucionado en respuesta a los avances tecnológicos y como ha impactado en la transformación de la industria.

En cuanto a la metodología de este informe, a través de la información obtenida de diferentes fuentes, está orientado a informar, particularmente, conocer los inicios, desarrollo y evolución actual de la programación para tener una visión completa de cómo surgió y a que camino se dirige.

Finalmente se da a entender que la programación es fundamental en la sociedad actual, ya que impulsa la innovación tecnológica y la resolución de problemas en una amplia gama de industrias. Su trayectoria desde sus inicios hasta la actualidad ha sido marcada por avances significativos, como la creación de lenguajes de programación, el surgimiento de la informática moderna y la inteligencia artificial. En la actualidad, la programación se puede considera como una habilidad esencial para el éxito en un mundo cada vez más impulsado por la tecnología, y su importancia solo continúa creciendo a medida que avanzamos hacia el futuro.

**Palabras claves**

Programación, resolución de problemas, informática, computación, historia de la programación, evolución de la programación, innovación tecnológica, sociedad digital,

**INTRODUCCIÓN**

La programación se puede definir como el proceso a través del cual se emplean una serie de códigos y cifras para introducir algoritmos necesarios para la realización de un proyecto o también se podría decir que su objetivo es la de resolver una problemática que nos planteen o se nos presente.

El primer hallazgo de algo relacionado con la programación lo encontramos en 1801 de la mano de Joseph Marie Jacquard y su telar programable. Con él se da a luz al primer sistema de programación, integrado en un telar que leía el código introducido y automatizaba los procesos.

Sin embargo, la programación tal y como la conocemos nace en 1957 con el conocido científico John W. Backus, cuando creó el primer lenguaje de programación de alto nivel, **Fortran**. Su finalidad era clarificar y facilitar la comprensión (ante los lenguajes erráticos e indescifrables que había en la época), acercándolo a una notación matemática normal.

**Origen de la programación**

La programación o el verbo programar es definido en el diccionario de Oxford obtenido de Google como “Dar las instrucciones necesarias a una máquina o aparato para que realice su función de manera automática. “

A partir de esta definición, es necesario que retroceder en la historia hasta 1801, momento en el cual Joseph Marie Jacquard invento el primer sistema de programación: un programa introducido en un telar a través de tarjetas perforadas y el mismo leía el código permitiendo tejer determinadas figuras. Luego de 30 años más tardes, Charles Babbage crea calculadora mecánica capaz de resolver no solo las operaciones elementales sino también ecuaciones, y la desarrollo utilizando como base las tarjetas perforadas de Jacquard.

**El papel de IBM en la evolución de la programación:**



La empresa IBM desempeño un papel significativo en la evolución de la programación. IBM fue una de las pioneras en el uso de tarjetas perforadas como medio para ingresar datos en las primeras computadoras, a fines de la década de 1920 y durante la década de 1930.   
  
Con el paso del tiempo, en 1957 John W. Backus creo Fortran, que significa “Formula Translation”. Backus tuvo una conexión importante con IBM durante gran parte de su carrera. Fue el primer lenguaje de programación conocido que tenía como fin realizar cálculos numéricos y científicos de manera eficiente, este fue el primer lenguaje de alto nivel. A pesar del paso del tiempo, este lenguaje se sigue utilizando en la industria automovilística, gubernamental, aeroespacial y en la investigación. En resumen, Fortran se utilizó para simplificar y mejorar la escritura de programas, haciéndolos más eficientes y legibles para realizar cálculos numéricos y análisis de datos.

Los eventos mencionados fueron los principales proyectos que sirvieron como desencadenantes de los posteriores desarrollos de lenguajes que comúnmente conocemos hasta el día de hoy.

**COBOL el lenguaje de negocios en la era de la informática**

En 1959 con la principal necesidad de un mismo lenguaje para todas las operaciones de negocios del sector corporativo, nació Cobol (Common Business Oriented Language). Fue desarrollado por Grace Hopper, es un lenguaje uniforme para transacciones empresariales, su objetivo era lograr la compatibilidad y funcionamiento en todos los ordenadores. Se encuentra detrás de la mayoría de las gestiones operacionales del sector empresarial, como los procesos de tarjeta de crédito, telefonía, etc. Su diseño se encuentra orientado a los archivos y aplicaciones, si bien sigue siendo hoy en día uno de los lenguajes predominantes en la informática del área mencionada debido a la gran capacidad de procesamiento de datos por lotes, su popularidad se vio disminuida por la falta de programadores experimentados que dominen el lenguaje, es por esto por lo que las empresas están comenzando a migrar sus procesos o reescribirlos en lenguajes más modernos. La mayor parte de la programación en COBOL ahora es para mantener las aplicaciones existentes.

**BASIC – La simplicidad que popularizó la programación.**   
En 1964 nació **BASIC**, familia de lenguajes de programación que surgió como una herramienta de apoyo enfocada a la enseñanza como un medio para facilitar la programación en ordenadores a estudiantes (y profesores) que no fueran de ciencias, pero que acabo adquiriendo una relevancia sorprendente, hasta el punto de que hoy, sigue siendo utilizado en programas como “Gambas” o “Visual Basic”.

Los ocho principios que rigieron el diseño de BASIC fueron:

* Ser fácil de usar para los principiantes.
* Ser un lenguaje de propósito general (no orientado).
* Permitir a los expertos añadir características avanzadas, conservando simple el lenguaje para los principiantes.
* Ser interactivo en todos los casos.
* Proveer mensajes de errores claros y amigables.
* Responder rápido en los [programas](https://es.wikipedia.org/wiki/Programa_(inform%C3%A1tica)) pequeños en general.
* No requerir un conocimiento del hardware de la computadora.
* Proteger al usuario del [sistema operativo](https://es.wikipedia.org/wiki/Sistema_operativo).

Una vez comenzada la década de los 70’s importantes desarrollos fueron llevados a cabo los cuales determinaron la creación de varios lenguajes de programación que hoy en día son muy populares.

**Inicios del Paradigma Orientado a Objetos**

En 1967 Kristen Nygaard y Ole-Johan Dahl de la Universidad de Oslo, desarrollaban sistemas informáticos que simulaban sistemas mecánicos. En estas simulaciones encontraron distintas dificultades: El programa era muy complejo y debía ser modificado constantemente.

Como solución, diseñaron el programa para que funcionara paralelamente al objeto físico, con la misma cantidad de componentes, dividiendo al programa en distintos módulos o unidades, uno por cada pieza física, esto lo llevaron a la práctica creando un lenguaje de programación llamado Simula 67 o simplemente Simula.

Es a partir de Simula que se introducen los conceptos de clases, objetos, herencia, procedimientos virtuales y referencia a objetos.

En los años 70 en California, Estados Unidos, el científico Alan Kay tuvo la oportunidad de examinar el lenguaje Simula, el cual aprovechó para llevar a cabo las ideas propuestas en su tesis doctoral, y junto con Adele Goldberg y Daniel Ingalls, crearon el primer lenguaje orientado a objetos, SmallTalk.

La programación en ese entonces tenía una falla grave, las variables eran visibles y modificables desde cualquier parte del código. David Parnas propuso un diseño modular, en el cual se oculta la información y se condensa cada una de las variables globales en un solo módulo con sus operaciones asociadas, de manera que solo fueran accesibles desde ese módulo u objeto, o de manera indirecta.

**C Y C++: La fundación de la programación de alto nivel**



**La década de 1970** fue una época importante en la historia de la programación, con varios desarrollos significativos que influyeron en la forma en que se programaba y en la tecnología de la información en general.  
Una de las contribuciones más significativas de la década de 1970 fue la creación del lenguaje de programación C por Dennis Ritchie en los laboratorios Bell de AT&T, este surge de la necesidad de un lenguaje de programación de alto nivel que fuera más portátil y eficiente para el desarrollo de sistemas y sistemas operativos. Su diseño sencillo y su capacidad para trabajar a bajo nivel con hardware permitieron su éxito y su influencia en la informática moderna. C es un lenguaje fundamental que ha influido en el desarrollo de numerosos lenguajes de programación posteriores y sigue siendo ampliamente utilizado en la actualidad. También fue muy importante el lanzamiento del sistema operativo Unix en los laboratorios Bell, este sentó las bases para muchos sistemas operativos modernos.

La característica que tuvo esta época fue el énfasis que se hizo sobre la programación estructurada como un enfoque para escribir software de una manera más clara y organizada. Lenguajes como Pascal y Ada promovieron este enfoque, lo que llevó a la creación de estructuras de control más legibles. Un lenguaje de programación también surgido en esta época fue Pascal, donde este tenía un enfoque en la claridad y legibilidad del código, después este fue opacado por C y C++, aunque sentó las bases para próximos lenguajes.  
Otras dos particularidades de esta época fueron, una que las microcomputadoras comenzaron a ganar popularidad, lo que llevó a un aumento en la demanda de programadores, su auge lo tuvo en la década de 1980. Y la otra particularidad fue el desarrollo de lenguajes específicos para bases de datos, ejemplo SQL, que se convirtió en una fija para la gestión de datos.

**La década de 1980**

La influencia del lenguaje C continuó en la década de 1980. Unix, un sistema operativo desarrollado en C, ganó popularidad en esta década y contribuyó a la difusión de C como un lenguaje de programación importante.  
 Se desarrollaron aplicaciones de software de productividad, como procesadores de texto y hojas de cálculo, lo que aumentó la demanda de programadores.

En esta década, a nuestro entender, solo hubo dos lenguajes de programación muy relevantes, uno es Ada, este es un lenguaje de programación de alto nivel desarrollado por el Departamento de Defensa de los Estados Unidos. Se diseñó para aplicaciones de seguridad, como sistemas de armas y sistemas de control de tráfico aéreo.

Y el más importante de la década fue C++, la extensión del lenguaje C. Surge antes la necesidad de un lenguaje que permitiera programar de manera más eficiente y orientada a objetos. Se desarrolló con un enfoque en la eficiencia y el control de hardware, lo que lo hace adecuado para programar sistemas y aplicaciones que requieren un alto rendimiento. Se esforzó por ser un lenguaje más fácil de usar y comprender y hasta el día de hoy es uno de los lenguajes más utilizados.

A partir de los años 80 surgen nuevos lenguajes orientados a objetos basados en Lisp, como Flavors, Ceyx y Loops, los cuales fueron fusionados en el primer lenguaje de programación orientado a objetos con un estándar ANSI: Common Lisp Object System, o CLOS.

En 1984 se plantea crear un sucesor al lenguaje C con orientación a objetos, y de esta forma Bjarne Stroustrup creó C++, incorporando ideas de Smalltalk y Simula. El lenguaje Modula fue diseñado por Niklaus Wirth en 1976 aunque no se desarrolló ningún compilador. Wirth continúo desarrollándolo, diseñando a mediados de los años 1980 un nuevo lenguaje, denominado Modula-2, basado en Pascal y en Modula.

Ada es el más largo y caro esfuerzo de diseño de un lenguaje de programación. El desarrollo fue encargado por el Departamento de Defensa de Estados Unidos, el cual se comenzó a mediados de los años 70 y termino en 1983.

**La década de 1990**

El rápido crecimiento de Internet en la década de 1990 fue el siguiente gran acontecimiento histórico para los lenguajes de programación. Con la apertura de una plataforma totalmente nueva para los sistemas informáticos, Internet creó una oportunidad adoptar nuevos lenguajes. En particular, el lenguaje de programación JavaScript se hizo popular debido a su pronta integración con el navegador web Netscape Navigator, y varios lenguajes de scripting alcanzaron un amplio uso en el desarrollo de aplicaciones personalizadas para servidores web. La década de 1990 no vio ninguna novedad fundamental en los lenguajes imperativos, pero sí mucha recombinación y la maduración de viejas ideas. Esta era comenzó la difusión de los lenguajes funcionales. Una filosofía de conducción grande era la productividad del programador. Surgieron muchos lenguajes de “aplicaciones de desarrollo rápido” (RAD), los cuales usualmente venían con un IDE, recolector de basura, y eran descendientes de lenguajes anteriores. Todos estos lenguajes eran orientados a objeto. Estos incluían Object Pascal, Visual Basic y Java. Java, en particular, recibió mucha atención. Pero más radicales e innovadores que los lenguajes de RAD eran los nuevos lenguajes de script. Estos no descendían directamente de otros lenguajes y ofrecieron nuevas sintaxis e incorporación más liberal de otras características. Muchos consideran estos lenguajes de script más productivos que los lenguajes de RAD, aunque esto se debe a menudo a que es más difícil escribir y mantener largos programas que pequeños programas simples. Sin embargo, no es menos cierto que los programas de script llegaron para convertirse en los más prominentes en la conexión con la Web.  
**Paradigmas de la programación**

Se denominan paradigma de programación a las formas de clasificar los lenguajes en función de sus características, los mismos, pueden estar clasificados en más de una categoría.

**Programación Imperativa**: En este tipo de desarrollo, el programador le da indicaciones a la maquina como cambiar su estado.

**Programación Procedimental:** Agrupa las instrucciones en procedimientos.

**Programación Orientada a Objetos:** Agrupan las instrucciones con la parte del estado en el que operan.

**Programación Declarativa:** El programador declara las propiedades del resultado deseado, pero no como calcularlo.

**Programación funcional:** El resultado deseado se declara el valor de una seria de aplicaciones de función.

**Programación lógica:** El resultado deseado se declara la respuesta a una pregunta sobre un sistema de hechos y reglas.

**Programación Matemática:** El resultado deseado se declara la solución de un problema de optimización.

**Lenguajes de programación procedurales**

El desarrollo de lenguajes procedurales dio lugar a lenguajes sirven de vocabulario relacionado con el problema a resolver. Ejemplos de estos lenguajes son:

* [COBOL](https://es.wikipedia.org/wiki/COBOL) : hace servir términos como archivo, mover y copiar.
* [FORTRAN](https://es.wikipedia.org/wiki/Fortran): hace servir terminología del lenguaje matemático.
* [ALGOL](https://es.wikipedia.org/wiki/ALGOL): enfocado a ser un lenguaje apropiado para definir algoritmos, también puede servir de lenguaje matemático.
* [BASIC](https://es.wikipedia.org/wiki/BASIC) se desarrolló para permitir escribir programas a más personas.
* [C](https://es.wikipedia.org/wiki/Lenguaje_C) - un lenguaje de propósito general.

Todos estos lenguajes siguen un paradigma procedural. Describen, paso a paso, exactamente el proceso que debe seguirse para resolver un problema específico. La [eficacia](https://es.wikipedia.org/wiki/Eficacia) y [eficiencia](https://es.wikipedia.org/wiki/Eficiencia_algor%C3%ADtmica) de cada solución dependen de la experiencia, la inventiva y la habilidad del programador.

**En la Actualidad**

Estos son los lenguajes de programación más populares según el [índice PYPL](https://pypl.github.io/PYPL.html) (mide cual es el nivel de búsqueda de tutoriales en Google y la [encuesta a programadores de HYPERLINK "https://survey.stackoverflow.co/2023/"Stack HYPERLINK "https://survey.stackoverflow.co/2023/" HYPERLINK "https://survey.stackoverflow.co/2023/"Overflow HYPERLINK "https://survey.stackoverflow.co/2023/" 2023](https://survey.stackoverflow.co/2023/).

|  |  |  |  |
| --- | --- | --- | --- |
| **Posición** | **Ranking PYPL de septiembre de 2023** | **Encuesta a programadores de Stack Overflow 2023** |  |
| #1 | Python | JavaScript |  |
| #2 | Java | HTML/CSS |  |
| #3 | JavaScript | Python |  |
| #4 | C# | SQL |  |
| #5 | C/C++ | TypeScript |  |
| #6 | PHP | Bash/Shell |  |
| #7 | R | Java |  |
| #8 | TypeScript | C# |  |
| #9 | Swift | C++ |  |

A continuación hablaremos del Top 3 del ranking actual.



**JavaScript, Java y Python: Tres gigantes de la programación  
Java** fue creado por James Gosling y su equipo en Sun Microsystems a mediados de la década de 1990. Nace como una respuesta a la necesidad de un lenguaje de programación que pudiera ejecutarse en múltiples plataformas, lo que llevó al desarrollo de la plataforma Java.  
 Su característica más destacada es la portabilidad, ya que el código escrito en Java puede ejecutarse en cualquier máquina que tenga una máquina virtual. Esto significa que las aplicaciones Java son independientes de la plataforma, lo que las hace adecuadas para desarrollo de aplicaciones de servidor, aplicaciones móviles, aplicaciones empresariales y más.  
  
**JavaScript** fue creado por Brendan Eich en 1995, es un lenguaje de programación diseñado principalmente para la programación del lado del cliente en aplicaciones web. Se utiliza para crear interactividad en páginas web, como validación de formularios, animaciones, manipulación del y comunicación con servidores.  
 A pesar de sus nombres similares, los dos anteriores, son lenguajes diferentes con propósitos y características distintas.  
  
Y por último **Python**, creado en 1980 con el objetivo de crear un lenguaje de programación que fuera fácil de leer, escribir y entender. Para esto el creador, Guido van Rossum busco palmar legitimidad en el código, expresar las ideas de manera clara y concisa, productividad, escribiendo menos líneas de código y versatilidad para que sea utilizado en amplia variedad de aplicaciones. Otra característica importante de Python es que está orientado a objetos lo que permite su creación y manipulación de manera sencilla.  
Este es utilizado en una amplia variedad de campos, desde desarrollo de software hasta ciencia de datos e inteligencia artificial.  
  
  
  
**Bibliografía**

Historia y evolución de los lenguajes de programación. Fecha de consulta 08/09/2023

* [Aranda, V. T. (2004). Historia y evolución de los lenguajes de programación HYPERLINK "https://www.acta.es/medios/articulos/informatica\_y\_computacion/034083.pdf".](https://www.acta.es/medios/articulos/informatica_y_computacion/034083.pdf)

Evolución de los sistemas de lenguajes de programación a través de la historia. Fecha de consulta 08/09/2023

* Alay, J. I. G., & Sevillano, R. P. C. (2022). Evolución de los sistemas de lenguaje de programación a lo largo de la historia. E-IDEA Journal of Engineering Science, 4(10), 14-26.

Historia de la programación. Fecha de consulta 10/09/2023

* Espinosa, J. R. (2012). Historia de la Programación. Obtenido de Historia de la Programación: [http://www](http://www/). it. uc3m. es/jvillena/irc/practicas/estudios/Lenguajes\_de\_Programacion. pdf.

Los videos juegos y su evolución desde la programación. Fecha de consulta 12/09/2023

* [Calderón Espitia, S. E. (2020). Los videojuegos y su evolución desde la programación (Doctoral HYPERLINK "http://caoba.sanmateo.edu.co/jspui/bitstream/123456789/241/1/TG-LOS%20VIDEOJUEGOS%20Y%20SU%20EVOLUCION%20DESDE%20LA%20PROGRAMACION%20%281%29.pdf"dissertation HYPERLINK "http://caoba.sanmateo.edu.co/jspui/bitstream/123456789/241/1/TG-LOS%20VIDEOJUEGOS%20Y%20SU%20EVOLUCION%20DESDE%20LA%20PROGRAMACION%20%281%29.pdf", Editorial Universitaria San Mateo).](http://caoba.sanmateo.edu.co/jspui/bitstream/123456789/241/1/TG-LOS%20VIDEOJUEGOS%20Y%20SU%20EVOLUCION%20DESDE%20LA%20PROGRAMACION%20%281%29.pdf)

Wikipedia – COBOL. Fecha de consulta 26/10/2023

* <https://es.wikipedia.org/wiki/COBOL>

Wikipedia – COBOL. Fecha de consulta 26/10/2023

* <https://es.wikipedia.org/wiki/BASIC>

Historia de la programación y Conceptos Claves. Fecha de Consulta 26/10/2023

* <https://syltec.es/blog/2021/10/05/programacion-un-poco-de-historia-y-conceptos-clave/>

Ranking de lenguajes de programación 2023

* <https://www.stackscale.com/es/blog/lenguajes-programacion-mas-populares/#Top_10_lenguajes_de_programacion_del_2023>