**Ada Lovelace creó el primer algoritmo informático**

Con Babbage mantuvo no solo una buena relación profesional en torno a la computación digital, sino también una larga amistad. En 1845, Ada tradujo al inglés un documento explicativo sobre el excepcional trabajo de Babbage al que fue añadiendo notas explicativas que ampliaron su extensión a casi el doble del original. Este traducción y explicación fue publicada en una revista francesa.

Esta investigación, que tenía su origen en el trabajo de Babbage, sirvió a la joven matemática para introducir algunas ideas sobre programación muy avanzadas para la época. Entre ellas, un sistema de tarjetas perforadas que serían esenciales para programar los primeros ordenadores a mediados del siglo XX. Estas aportaciones de Lovelace supusieron la creación del primer algoritmo destinado a ser procesado por una máquina.

Además de estas aportaciones, Ada Lovelace tomó como referencia el número de Bernoulli para desarrollar lo que hoy conocemos como **algoritmo informático**. Sin embargo, uno de los grandes avances que desarrolló fue la introducción de este algoritmo en la máquina de Babbage. Este hecho le permitió definir lo que actualmente en informática se conoce como **«bucle»:** ejecutar una serie de acciones de forma repetida.

A su vez, **Ada Lovelace** diseñó la idea de una máquina que fuera capaz de programarse y reprogramarse para ejecutar funciones que le llevarán a la realización de una serie de tareas que fueran más allá de llevar a cabo simples cálculos y que quedarán expresadas mediante símbolos.

**Lenguajes de programación más antiguos:**

* **Ensamblador (Assembly):** Surgió en la década de 1940 y fue uno de los primeros lenguajes de programación de bajo nivel.
* **Fortran (1957):** Uno de los primeros lenguajes de alto nivel, diseñado principalmente para cálculos científicos y matemáticos.
* **Lisp (1958):** Un lenguaje funcional que se utiliza principalmente en inteligencia artificial y procesamiento simbólico.
* **COBOL (1959):** Lenguaje de programación diseñado para la gestión de datos y aplicaciones comerciales.
* **Algol (1958):** Fue influyente en la evolución de lenguajes de programación, especialmente en la teoría de los lenguajes de programación.

**Lenguajes intermedios (décadas de 1970-1990):**

* **C (1972):** Uno de los lenguajes más influyentes y populares, utilizado en sistemas operativos y desarrollo de software en general.
* **Pascal (1970):** Usado principalmente para la enseñanza de conceptos de programación estructurada.
* **Smalltalk (1972):** Un lenguaje orientado a objetos, que influyó en el diseño de otros lenguajes como Python y Ruby.
* **C++ (1985):** Una extensión de C que introduce la programación orientada a objetos.
* **Java (1995):** Lenguaje multiplataforma ampliamente utilizado en aplicaciones empresariales y móviles.
* **JavaScript (1995):** Lenguaje de programación para desarrollo web.

**Lenguajes más nuevos (2000 hasta la actualidad):**

* **Python (1991, pero se popularizó en los 2000s):** Se destaca por su simplicidad y versatilidad, utilizado en ciencia de datos, inteligencia artificial, desarrollo web y automatización.
* **Ruby (1995):** Conocido por su facilidad de uso, especialmente en el marco web Ruby on Rails.
* **Go (2009):** Desarrollado por Google, es conocido por su simplicidad y eficiencia en la creación de software de alto rendimiento.
* **Swift (2014):** Lenguaje desarrollado por Apple para el desarrollo de aplicaciones iOS y macOS.
* **Rust (2015):** Un lenguaje de sistemas diseñado para ser seguro, rápido y concurrente, que ha ganado popularidad rápidamente.
* **Kotlin (2011, popularizado en 2017):** Un lenguaje que se ejecuta en la JVM y es utilizado principalmente para desarrollo Android.

**1. Python**

* **Uso:** Python es uno de los lenguajes más populares en la actualidad, conocido por su simplicidad y legibilidad. Se utiliza en una amplia gama de aplicaciones:
  + **Ciencia de Datos e Inteligencia Artificial (IA):** Python es la lengua franca para proyectos de Machine Learning (ML), procesamiento de datos y análisis, con bibliotecas como pandas, NumPy, scikit-learn, TensorFlow, y Keras.
  + **Desarrollo Web:** Frameworks como Django y Flask permiten desarrollar aplicaciones web de manera rápida y eficiente.
  + **Automatización y Scripting:** Python es ideal para escribir scripts que automatizan tareas repetitivas o sistemas de administración.
  + **Desarrollo de Software:** También se usa para aplicaciones de escritorio o software empresarial, con bibliotecas como Tkinter o PyQt.
* **Por qué se usa:** Es fácil de aprender, tiene una sintaxis limpia, y tiene una enorme comunidad de soporte.

**2. JavaScript**

* **Uso:** JavaScript es el lenguaje principal de desarrollo web del lado del cliente (frontend), pero también se usa del lado del servidor con Node.js.
  + **Desarrollo Web (Frontend):** JavaScript es fundamental para la creación de páginas web dinámicas e interactivas. Trabaja junto con HTML y CSS para hacer sitios web funcionales.
  + **Desarrollo de Aplicaciones Web (Backend):** Usando Node.js, JavaScript puede ser usado en el servidor, lo que permite a los desarrolladores usar un solo lenguaje para todo el stack de desarrollo.
  + **Desarrollo Móvil y de Escritorio:** Frameworks como React Native permiten crear aplicaciones móviles, y Electron permite crear aplicaciones de escritorio.
* **Por qué se usa:** Todos los navegadores web lo soportan, lo que lo hace indispensable para desarrollo web. Además, con el auge de Node.js, ahora también se usa mucho en el backend.

**3. Java**

* **Uso:** Java ha sido durante mucho tiempo uno de los lenguajes más utilizados, especialmente en aplicaciones empresariales y móviles.
  + **Desarrollo Empresarial:** Java se usa para construir aplicaciones empresariales grandes y complejas. Plataformas como Spring y Hibernate son muy populares en el desarrollo de aplicaciones Java.
  + **Desarrollo Móvil (Android):** Java fue el lenguaje principal para el desarrollo de aplicaciones Android (aunque Kotlin ha ganado terreno).
  + **Aplicaciones en la Nube y Big Data:** Con plataformas como Apache Hadoop, Java también es usado en el análisis de grandes volúmenes de datos.
* **Por qué se usa:** Es un lenguaje robusto, escalable y seguro, con una gran comunidad y una gran cantidad de herramientas y bibliotecas.

**4. C/C++**

* **Uso:** Aunque se considera más bajo nivel, C y C++ siguen siendo fundamentales para ciertas aplicaciones que requieren rendimiento extremo y control sobre el hardware.
  + **Sistemas Operativos y Software de Bajo Nivel:** Ambos lenguajes se usan para escribir sistemas operativos, controladores de dispositivos y otros software de bajo nivel.
  + **Juegos:** C++ es ampliamente utilizado en el desarrollo de videojuegos, ya que permite un control preciso sobre los recursos del sistema y una ejecución eficiente.
  + **Aplicaciones de Alto Rendimiento:** C se usa en situaciones donde el rendimiento es crucial, como en aplicaciones científicas y de ingeniería.
* **Por qué se usa:** Son extremadamente eficientes y permiten acceso directo a la memoria, lo que los hace ideales para sistemas donde el rendimiento es clave.

**5. C# (C-Sharp)**

* **Uso:** C# es un lenguaje desarrollado por Microsoft que se utiliza principalmente para el desarrollo de aplicaciones en la plataforma .NET.
  + **Desarrollo de Aplicaciones de Escritorio:** Utilizado en el desarrollo de aplicaciones de escritorio para Windows, especialmente con frameworks como WPF (Windows Presentation Foundation).
  + **Desarrollo de Juegos:** C# es ampliamente usado en el desarrollo de videojuegos, especialmente con el motor de juego Unity.
  + **Aplicaciones Web:** Con ASP.NET, C# también se usa para construir aplicaciones web dinámicas.
* **Por qué se usa:** C# ofrece una sintaxis moderna, es fácil de aprender, y tiene un fuerte ecosistema respaldado por Microsoft.

**6. Ruby**

* **Uso:** Ruby es conocido principalmente por su framework web, **Ruby on Rails**, que es una herramienta poderosa para construir aplicaciones web rápidamente.
  + **Desarrollo Web (Backend):** Ruby on Rails hace que la creación de aplicaciones web sea más fácil, enfocándose en la simplicidad y el diseño "convención sobre configuración".
  + **Automatización y Scripting:** Ruby también se usa en el desarrollo de scripts y herramientas de automatización.
* **Por qué se usa:** Ruby es fácil de aprender y permite a los desarrolladores crear aplicaciones de manera rápida y eficiente. Su enfoque en la productividad lo ha hecho popular entre las startups.

**7. PHP**

* **Uso:** PHP sigue siendo uno de los lenguajes más utilizados para desarrollo web, especialmente para aplicaciones dinámicas y sitios web.
  + **Desarrollo Web (Backend):** Es ampliamente usado para crear aplicaciones web, especialmente con sistemas de gestión de contenido como WordPress, Joomla y Drupal.
  + **Aplicaciones de Servidor Web:** PHP se ejecuta del lado del servidor y se usa para procesar formularios, interactuar con bases de datos y generar contenido dinámico.
* **Por qué se usa:** PHP es fácil de aprender y está integrado en muchos servidores web, lo que lo hace muy accesible para desarrolladores web. Además, tiene una gran comunidad de soporte.

**8. Swift**

* **Uso:** Swift es el lenguaje principal para desarrollar aplicaciones para los sistemas operativos de Apple (iOS, macOS, watchOS, tvOS).
  + **Desarrollo Móvil (iOS):** Swift es usado para crear aplicaciones nativas para iPhone y iPad.
  + **Desarrollo de Software de Apple:** Se usa también para aplicaciones en el ecosistema macOS y otros dispositivos Apple.
* **Por qué se usa:** Swift es rápido, moderno y fácil de aprender. Apple lo desarrolló específicamente para ser más seguro y eficiente que su antecesor, Objective-C.

**9. Kotlin**

* **Uso:** Kotlin ha ganado una gran popularidad como lenguaje de desarrollo para aplicaciones Android.
  + **Desarrollo Móvil (Android):** Kotlin es ahora el lenguaje preferido para Android, especialmente debido a su sintaxis más limpia y sus características modernas.
  + **Desarrollo Web y Backend:** Kotlin también se usa para aplicaciones backend, especialmente con el framework Ktor.
* **Por qué se usa:** Es completamente interoperable con Java, pero ofrece mejoras significativas en términos de concisión, seguridad y facilidad de uso.