La programación de computadoras: ayer y hoy

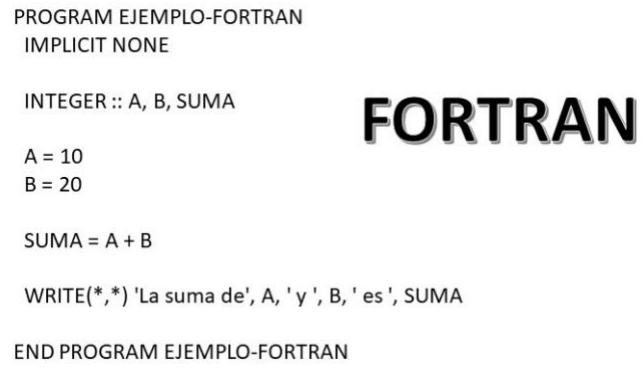
Francisco Vargas Navarro, Miembro Senior IEEE, vicepresidente CPIC y catedrático de la Universidad INVENIO

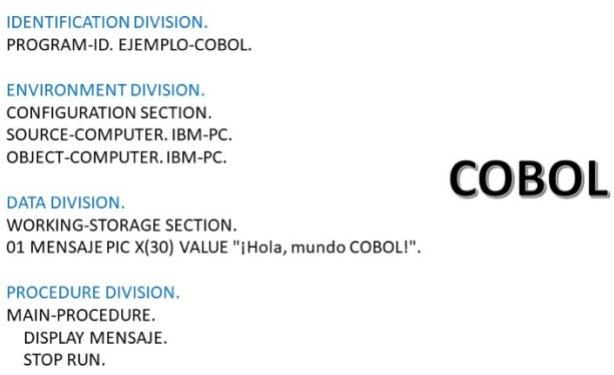
INTRODUCCIÓN

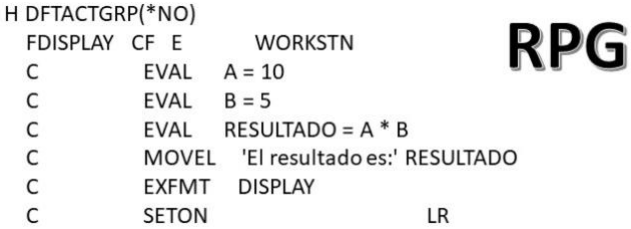
A través de los años el proceso de programación de las computadoras ha ido cambiando, así como cambió la tecnología del procesamiento electrónico de datos como tal. Ante el avance en la fabricación de las computadoras, los lenguajes de programación se han ido adaptando para sacar el máximo provecho de las características innovadoras que llegan con los cambios tecnológicos. Revisemos como han cambiado los lenguajes de programación, a la luz de estos cambios.

EL PASADO

Para los años 60's y parte de los 70's la electrónica usada para el procesamiento electrónico de datos estaba en el proceso del reemplazo de los tubos al vacío por transistores y luego los circuitos integrados. Desde el punto de vista de sistemas de información se exploraba la posibilidad de terminales de procesamiento para los usuarios versus el típico proceso centralizado tipo "lotes". La programación para esos procesos "batch" era basada en lectura de archivos, típico maestro-detalle. Lenguajes como COBOL (Common Business Oriented Language) estaban de primeros en la lista de programación, pero también algunos fabricantes como IBM impulsaban RPG (Report Program Generator). Hay que recordar que ya FORTRAN estaba en el ambiente para casos de procesamiento numérico. La programación por lotes tenía la característica de que el programador no tenía que preocuparse por pantallas de interacción con el usuario si no en garantizar un adecuado procesamiento de la información. Por otro lado, los sistemas de base de datos empezaban a ganar popularidad como opción a los sistemas de archivos VSAM (Virtual Storage Access Method) e ISAM (Indexed Sequential Access Method), rutinas integradas al sistema operativo que se encargaban de proveer acceso vía llaves específicas. Los compiladores de los lenguajes también eran algo muy diferente a lo actual pues también aplicaban el proceso "batch". Un programa COBOL se escribía en un editor de texto y una vez terminado y grabado en el disco, servía de entrada a un programa llamado compilador que lo procesaba, detectando errores y de no existir estos, se generaba un objeto para ser ejecutado en el sistema operativo. Los errores se mostraban al final del proceso y dado que COBOL es un lenguaje totalmente estructurado, un error en las primeras líneas generaba una cascada de errores considerable. Un programa pequeño, 100 líneas, podía generar 800 errores y más. No existían los interpretadores o programas de procesamiento en línea como los conocemos hoy.







Ante la presión de los avances tecnológicos y los cambios en las teorías de desarrollo de sistemas, los sistemas operativos se tuvieron que adaptar al manejo de terminales, interacción con los usuarios punto a punto con cada uno de ellos. El ambiente con múltiples usuarios (Multiuser) y múltiples tareas por usuario (Multitasking) se abría paso. Ya los usuarios podían disfrutar el

ISSN: 2015-5392

procesamiento de datos en terminales desde su puesto de trabajo y hacer uso de programas tipo "On-Line" donde ellos mismos enviaban información y recibían respuesta inmediata.

En esos tiempos un lenguaje tomó terreno, se llamó BASIC (Beginners' All-purpose Symbolic Instruction Code). Este lenguaje era interpretado y no requería generar un objeto. Solo se requería que el sistema operativo tuviese las librerías para el soporte de este lenguaje. Los programas eran línea a línea, cada fabricante tenía su propio estilo de implementación y alguno de ellos lo tomaron como lenguaje base, por ejemplo, Data General o WANG. Nuevamente los avances en tecnología marcaron el avance de los lenguajes de programación, se presentó al mercado el PC (Personal Computer) por la empresa IBM (International Business Machines Corporation) y con este, una empresa llamada MICROSOFT escribió un sistema operativo que finalmente tomo el nombre de MS-DOS (Microsoft Disk Operating System).

La salida del PC al mercado mundial creó un antes y un después en el procesamiento electrónico de datos. Antes de esto, los usuarios tenían terminales, pero ahora disfrutaban del procesamiento local. Ellos tenían en sus manos un equipo con procesador, memoria y disco, la PC. Los fabricantes de programas empezaron a sacar al mercado productos compatibles con el sistema operativo MS-DOS. "Compatible" fue un término que se pudo de moda y mucho de ello fue por la salida de un gigante, un insigne en temas de sistemas operativos, UNIX. Este sistema operativo se originó como AT&T UNIX inició en 1969 y presentó características importantes como ser multiusuario y multitarea. En ese momento teníamos productos para UNIX y productos de MS-DOS. Los fabricantes, de programas y equipos, se inclinaban a uno o el otro. La competencia está presente hasta hoy en día.

A nivel de procesadores surgió una competencia por la arquitectura de estos entre los defensores del modelo CISC (Complex Instruction Set Computer) y RISC (Reduced Instruction Set Computer). Los sistemas operativos eran diferentes y los compiladores también, aunque el cambio a nivel de lenguajes de programación no fue tan sensible.

Solo como comentario al margen, las redes de datos se volvieron un tema a considerar, pero eso es asunto para otro artículo.

EL PRESENTE

La conectividad de los modelos de redes locales sumado a los servicios de la red de redes INTERNET más los avances tecnológicos en el procesamiento de datos, provocaron cambios sensibles en los lenguajes de programación. Atrás quedó COBOL y su rigidez, pasando con aquellas versiones de PASCAL y en su lugar surgen lenguajes como C, C#, C++, PYTHON.

```
a = 10
b = 20

suma = a + b

print("La suma de", a, "y", b, "es", suma)
```

PYTHON

La gente de SUN MicroSystem (SUN), ahora ORACLE, sacó su muy popular JAVA.

```
public class EjemploJava {
   public static void main(String[] args) {
      int a = 10;
      int b = 20;

      int suma = a + b;

      System.out.println("La suma de " + a + " y " + b + " es " + suma);
   }
}
```

JAVA

Estos lenguajes salen acompañados de su IDE (integrated development environment). Fabricantes de software pusieron en el mercado IDEs alternativos, dando al usuario múltiples y muy buenas opciones para configurar su ambiente de la mejor manera.

El aspecto del video local evolucionó totalmente incluyendo el uso de color y alta resolución HD. Esto implicó que los lenguajes de programación deben poder controlar este tipo de dispositivos. Apareció el término programación visual. El modelo más conocido en la presentación de información en un monitor es el usado por Microsoft llamado WINDOW que a su vez se ha convertido en el nombre de su sistema operativo WINDOWS. APPLE por su lado ya tenía su entorno de ventanas propio desde el principio y también, su propio sistema operativo basado en UNIX.

Estos modelos provocaron un cambio en el paradigma de la programación pasando de un estilo entrada-proceso-salida a una programación por eventos sobre ciclos de la CPU en estado "Wait-On-a-Event" o espera por evento/cambio.

En ese contexto Microsoft sacó su entorno IDE al cuál llamó Visual Code y dio la funcionalidad de interactuar con diferentes lenguajes compilados. Sin embargo, varios fabricantes de programas presentan opciones muy interesantes de entornos de desarrollo integrados específicos para cada lenguaje.

Al tomar forma la gestión de almacenamiento secundario usando bases de datos, los lenguajes se ven en la obligación de incluir en su repertorio la conectividad con los diferentes productos de bases de datos. Así que los lenguajes actuales deben incluir bibliotecas o módulos para la interacción con la base de datos. Así mismo, es necesaria la interacción con el sistema operativo mediante la creación de ventanas y un entorno de

ISSN: 2015-5392

interfaz gráfica de usuario (GUI) donde el uso de iconos se volvió una prioridad. La posibilidad de procesamiento local genera posibilidades ilimitadas en términos de programación.

INTERNET también presentó en un desafío a los lenguajes de programación. El uso del protocolo HTTP (Hypertext Transfer Protocol) para el intercambio de información entre un proveedor de datos y el cliente local que presentará los datos al usuario, hizo que fuese necesario presentar una nueva familia de lenguajes como soporte para el modelo que opera sobre HTTP al cuál se le llamó HTML (HyperText Markup Language), de paso algunos lo consideran un lenguaje y otros no. Lo que es un hecho es que aparecieron en el mercado nuevos nombres como: PHP, PERL, ASP, ASPX y el ahora muy famoso JAVASCRIPT. Todos ellos con la funcionalidad de interactuar con HTML para mejorar su capacidad de procesamiento. De modo que se presentó una combinación entre HTML, CSS (Cascading Style Sheets) a los cuales se les asignó la tarea de formatear la presentación al usuario y JAVASCRIPT como responsable por la programación de tareas locales.

JAVASCRIPT

```
<!DOCTYPE html>
<html>
 <head>
  <meta charset="utf-8">
  <title>Suma con JavaScript</title>
 </head>
 <body>
  <label for="a">Primer número:</label>
  <input type="number" id="a" name="a"><br>

  <label for="b">Segundo número:</label>
  <input type="number" id="b" name="b"><br>

  <button onclick="sumar()">Sumar</button><br>

  <p id="resultado"></p>

  <script>
   function sumar() {
    let a = parseInt(document.getElementById("a").value);
    let b = parseInt(document.getElementById("b").value);

    let suma = a + b;

    document.getElementById("resultado").textContent = `La suma de ${a} y
${b} es ${suma}`;
   }
  </script>
 </body>
</html>
```

Todo esto ejecutado desde un navegador, lo que dio paso a un estilo de programación tipo WEB (World Wide Web) y a la designación de los programas resultantes como tipo WEB ENABLE.

En la actualidad tenemos dos arquitecturas sobre las cuales se está programado: una de ellas es la tradicional con objetos ejecutados por el sistema operativo y generados a través de un compilador o bien la ejecución de lenguaje vía un intérprete. La otra arquitectura es la WEB ENABLE donde los programas viajaran por INTERNET o por los menos usando los protocolos de esta red y esto implica HTML como base.

Ya desde los 80's teníamos un lenguaje especializado en el área de BASES DE DATOS llamado SQL (Structured Query Language). En los PCs productos como DBASE, CLIPPER, FOXBASE dieron paso a sistemas más robustos como POSTGRESQL, MYSQL, ORACLE, MS-SQL. Todos estos motores de bases de datos cumplen con una condición en común y es la utilización del lenguaje SQL. Un programador puede especializarse en SQL y dedicarse a la construcción de los diferentes procesos de inclusión, borrado, actualización, consulta, que permiten a una aplicación obtener los servicios de almacenamiento secundario.

SQL

```
SELECT clientes.nombre, COUNT(pedidos.id_pedido) as total_pedidos
FROM clientes
INNER JOIN pedidos ON clientes.id_cliente = pedidos.id_cliente
GROUP BY clientes.id_cliente
HAVING COUNT(pedidos.id_pedido) > 0;
```

EL FUTURO

La programación por eventos toma su lugar y por ende los lenguajes deben incorporar un esquema WINDOW-EVENT mediante el cuál interactúen con el usuario a través de los equipos sofisticados en graficación, sonido, interacción tipo "touch-screen", etc.

C++

```
#include <QApplication>
#include <QPushButton>

int main(int argc, char *argv[]) {
  QApplication app(argc, argv);

  QPushButton button("Haz clic");
  button.resize(200, 100);
  button.show();

  QObject::connect(&button, &QPushButton::clicked, [&]() {
    button.setText("Haz clic de nuevo");
  });

  return app.exec();
}
```

El incremento en el uso de INTERNET en el mundo así como el uso masivo de teléfonos inteligentes, incluso la conectividad de diferentes dispositivos IoT (Internet Of Things) nos está forzando a realizar nuevos cambios de las funcionalidades de los lenguajes de programación. Lenguajes como PYTHON y C++ enfrentan el reto de mantenerse vigentes mediante la creación de nuevas

librerías. El teléfono celular implicó un cambio del ambiente final para nuestra programación. IDEs especializados para ambiente de celular con las librerías necesarias están disponibles en el mercado. Mejoras en el procesamiento electrónico de datos nuevamente obligan a cambios en los lenguajes de programación, el auge de controladores como ARDUINO y similares, hacen que se tengan versiones especializadas para generar objetos compatibles con estos equipos o programas que puedan ser interpretados por los procesadores mediante su sistema operativo. Los nuevos dispositivos y sus capacidades retan todos los días a los programadores.

Otro fenómeno para tomar en cuenta son las llamadas redes sociales. Estas generan millones de BYTES (Unidad binaria usada para referirse a los datos) pero su formato ya no corresponde al procesamiento de datos tradicional, sino que ahora tenemos: Imágenes, Audio, Video. Los lenguajes de programación deben presentar librerías que permitan la manipulación de este tipo de información. Incluso a nivel de bases de datos ya tenemos productos especializados en estos nuevos formatos no estructurados.

Lenguajes orientados a propósitos específicos como R, desarrollado para computación estadística, así como lenguajes desarrollados por empresas a nivel mundial como GO de la empresa GOOGLE, terminan de completar nuestra visión de los que tenemos y hacia donde vamos.

El desafío para cualquier programador programador será mantenerse al día y aprender a usar los nuevos lenguajes que van saliendo al mercado. Ardua tarea.

C#

```
using System;
using System.Data.SqlClient;

namespace EjemploBD
{
  class Program
  {
    static void Main(string[] args)
    {

      string connectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial
Catalog=EjemploBD;Integrated Security=True";

        string createTableQuery = "CREATE TABLE Usuarios (ID INT PRIMARY
KEY IDENTITY, Nombre NVARCHAR(50), Edad INT)";

      string insertQuery = "INSERT INTO Usuarios (Nombre, Edad) VALUES
('Juan', 30), ('Pedro', 25)";

        string selectQuery = "SELECT * FROM Usuarios";

        using (SqlConnection connection = new
SqlConnection(connectionString))
      {
        connection.Open();

        using (SqlCommand command = new SqlCommand(createTableQuery,
connection))
        {
          command.ExecuteNonQuery();
          Console.WriteLine("Tabla creada");
        }

        using (SqlCommand command = new SqlCommand(insertQuery,
connection))
        {
          int rowsAffected = command.ExecuteNonQuery();
          Console.WriteLine($"{rowsAffected} filas insertadas");
        }
        using (SqlCommand command = new SqlCommand(selectQuery,
connection))
        {
          SqlDataReader reader = command.ExecuteReader();
          while (reader.Read())
          {
            int id = reader.GetInt32(0);
            string nombre = reader.GetString(1);
            int edad = reader.GetInt32(2);
            Console.WriteLine($"ID: {id}, Nombre: {nombre}, Edad: {edad}");
          }
          reader.Close();
        }
      }

      Console.ReadKey();
    }
  }
}
```

R

```
x <- c(1, 2, 3, 4, 5)

mean_x <- mean(x)
sd_x <- sd(x)

cat("La media de x es:", mean_x, "\n")
cat("La desviación estándar de x es:", sd_x, "\n")
```


ISSN: 2015-5392

```
package main

import "fmt"

func main() {

        x := 10

        fmt.Println("El valor de x es:", x)

        y := x + 5

        fmt.Println("El resultado de x + 5 es:", y)
}
```

GO

GLOSARIO

COBOL, RPG, FORTRAN, TURBO PASCAL, C, C#, C++, PYTHON, JAVA, JAVASCRIPT, PHP, ASP, ASPX, PERL, R, GO: lenguajes de programación de alto nivel.

ISAM, VSAM: mecanismos para la manipulación de archivos secuenciales indexados.

IBM, Data General, Microsoft, WANG, SUN: empresas dedicadas a la fabricación de "software" y/o "hardware" dedicados al procesamiento electrónico de datos.

MS-DOS, UNIX, WINDOWS: sistemas operativos multiusuario, multitarea.

HTTP, HTML, CSS: componentes parte del esquema WEB ENABLE en INTERNET.

SQL: Lenguaje para la interacción con Bases de datos.

DBASE, FOXBASE, CLIPPER, POSTGRESQL, MYSQL, ORACLE, MS-SQL: motores (DBMS) de bases de datos.

RISK, CISC: arquitecturas de procesadores.

ARDUINO: controladores tipo "single-board".

REFERENCIAS

Banzi, M.; Mellis, D.; Cuartielles, D. (2005) Arduino Project. https://www.arduino.cc/

Ellison, L.; Miner, B.; Oates, Ed (1977) Oracle Corporation. https://www.oracle.com/

Flint, C. (1911) International Business Machines Corporation IBM. https://www.ibm.com

Gates, B. (1975) Microsoft Corporation. https://www.microsoft.com

OpenAI. (2021). ChatGPT. https://openai.com/research/

Page, L.; Brin, S. (1998) Google LLC https://www.google.com/

Wales, J.; Sanger, L (2001) Wikipedia. https://www.wikipedia.org/

Wang, A.; Chu, G. (1951) Wang Labs.