

# Introducción a los Lenguajes de Cuarta Generación (4GL)

**Alcides A. Solano Mata**

Universidad de Costa Rica, Fac. de Ingeniería,  
San José, Costa Rica  
[alon063@gmail.com](mailto:alon063@gmail.com)

**Gustavo A. Yong Morales**

Universidad de Costa Rica, Fac. de Ingeniería,  
San José, Costa Rica  
[tavoyong@gmail.com](mailto:tavoyong@gmail.com)

**Andrés S. Camacho Brenes**

Universidad de Costa Rica, Fac. de Ingeniería,  
San José, Costa Rica  
[acamcx@gmail.com](mailto:acamcx@gmail.com)

## **Abstract:**

The fourth generation languages (4GL) are tools that optimize the software development, automating the creation of software. It have been used to generate GUI code and in the implementation of programs that make things easy for the developers and clients to resolve problems. The 4GL base consist in getting information about what the user need not how to do it, this can be from a data base report generation, to solve a mathematic problem.

**Key words:** 4GL, automatization , software, development

## **Resumen:**

Los lenguajes de cuarta generación o bien 4GL son herramientas encargadas de optimizar el desarrollo de software automatizando la creación de este. Se han utilizado principalmente en la generación de código para GUI y además en la implementación de programas que facilitan las tareas de los desarrolladores y clientes. La base de los 4GL es que el usuario no dice como se harán las cosas, solo brinda parámetros y el programa se encargara de generar los solicitado por el usuario, sea esto desde generación de reportes de bases de datos a solucionar un problema matemático.

**Palabras clave:** 4GL, automatización, software, desarrollo

## 1. Introducción:

En el mundo actual donde la tecnología tiene cada vez un papel más importante junto con la computación, es importante conocer un poco de los lenguajes y herramientas que utilizamos constantemente los que nos vemos envueltos en este mundo. Es de suma importancia conocer sus características y las ventajas que tienen sobre otros tal vez más viejos, esto para saber hacer una elección correcta de un lenguaje para una aplicación que deseamos crear, la cual se encargara de facilitar o bien automatizar las necesidades de un cliente.

Cuando pensamos en automatizar, en lo que pensamos es en un programa implementado por algún desarrollador de software para ayudar en las tareas de un cliente. Los 4GL son la herramienta destinada a automatizar el proceso de desarrollo de software para crear otro software. En resumen una herramienta que aumenta el rendimiento y velocidad del desarrollo del software.

Como presentaremos más adelante la evolución que han sufrido los lenguajes de programación ha sido grandiosa y cada vez existen mas herramientas que proporcionan mayor facilidad a la hora de realizar una aplicación.

## 2. Historia de los lenguajes de programación

La evolución de los lenguajes de programación ha sido constante dentro del siglo XX pasando por varias etapas bien definidas llamadas generaciones.

La primera etapa de los lenguajes de programaron consiste en el uso de código binario y tarjetas perforadas que funcionaban con gran rapidez pero eran difíciles de entender y de poca portabilidad a otros sistemas diferentes al original. [4]

La segunda generación contempla el lenguaje ensamblador y uso de etiquetas, estos lenguajes a pesar de agregar el uso de abstracción en su minina capacidad y de ser muy eficientes, dependían en gran medida de la arquitectura sobre la que se desarrollaban. [4]

La tercera etapa de los lenguajes de programación viene a ampliar más el uso del término abstracción, eran de programación estructurada y todavía basados en el como hacer las cosas, agregan mayor funcionalidad pero bajan el rendimiento. [4]

El término “Lenguajes de Cuarta Generación” (4GL) fue primeramente usado para describir a los lenguajes de especificación, que son lenguajes de alto nivel no procedurales (el usuario define qué se debe hacer, no cómo debe hacerse [1]), la denotación 4GL fue utilizada por primera vez por James Martín en 1982[2]. Los lenguajes RPG de IBM e Informatics de IBM y MARK-IV respectivamente pueden ser considerados los primeros lenguajes de este tipo. [1]

### Ventajas Principales

Lenguajes de Tercera Generación	Lenguajes de Cuarta Generación
<b>Estandarización</b>	<b>Flexibilidad</b>
<b>Actualizaciones Continuas</b>	<b>Nuevas Aplicaciones</b>
<b>Volumen de Código</b>	<b>Conversión de Código</b>
<b>Rendimiento de Ejecución</b>	<b>Mayor Productividad</b>

Figura1. Ventajas de los 4GL[8]

## 3. Lenguajes de Cuarta Generación

Los lenguajes de cuarta generación contemplan un gran grado de abstracción que viene a crear una verdadera caja negra pues el programador no dice como se realizaran las aplicaciones que se desee crear, nada más dará una serie de parámetros que aunque dependen del programa que se quiera crear siempre van a estar muy alejados de la máquina o hardware, esta ha sido la tendencia a seguir con cada generación de los lenguajes de programación. [8]

Para mostrar lo anterior basta contemplar que para un mismo programa mientras hay cientos de líneas de código en un lenguaje de tercera generación, en un 4GL la cantidad de código necesario pasa a ser de unas pocas líneas nada más. Esto por que los 4GL vienen acompañados de poderosas herramientas y se acercan en gran medida al lenguaje humano, siendo más concretos y fáciles de entender.

Hay que recordar que la potencia de los 4GL se basa en la posibilidad de generar código automáticamente y que sin embargo la funcionalidad adquirida hasta hoy ha sido bastante reducida.

Los primeros 4GL han basado toda su potencia de generación automática de código en el prototipaje de GUI, dejando un déficit en la generación de código orientado al manejo de datos y los procesos funcionales[8].

Se debe considerar que los 4GL han cambiado mucho desde los primeros lenguajes incluidos dentro de este término, la evolución se ha visto marcada por varios aspectos, uno de ellos es las restricciones del sistema operativo y del mismo hardware, que han marcado en gran medida la pauta a seguir. [3]

Uno de los cambios sustanciales que han tenido los lenguajes de cuarta generación ha sido su relación con las bases de datos cambiando el uso de técnicas y recursos en comparación con los 4GL más viejos. Además del mayor manejo de interfaces por el usuario. [3]

También los lenguajes de cuarta generación se han visto influenciados por las tendencias de cambio que ha sufrido la computación con el pasar de los años, como la tendencia a programar con orientación a objetos, la incorporación de la arquitectura cliente /servidor, la naciente ingeniería de software y la tendencia a trabajar en equipo. [8]

La programación orientada a objetos y la disposición de los desarrolladores están logrando poner en el mercado librerías de objetos que complementan el funcionamiento de los 4GL[8]. Entre las librerías más comunes se encuentran objetos para el acceso a múltiples bases de datos relacionales, objetos para el acceso a sistemas de mensajería electrónica, objetos para el acceso a sistemas de Workgroup, objetos para el acceso a bases de datos jerárquicas, objetos para el uso avanzado de GUIs, entre otras.

Sin embargo, aun se necesita mejorar mucho los modelos que incluyen los 4GL, existen pocos procesos modelizados y la dificultad radica en encontrar aquellos procesos que el usuario más necesita. Mas adelante discutiremos algunos de los tipos de 4GL que hay, al igual que ejemplificaremos el tema con casos concretos de productos 4GL.

#### **4. Tipos de lenguajes de cuarta generación**

Existen diferentes tipos de lenguajes de cuarta generación, cada uno con una función en particular. Entre ellos están los generadores de reportes, los generadores de “forms”, ambientes de cuarta generación, administradores de datos y los generadores de aplicaciones.

Los generadores de reportes (Report Generators) toman una descripción del formato y el reporte a generar, y de ahí pueden generar el reporte de forma directa, o bien un programa que genere el reporte[3]. Existen unos generadores de reportes utilizados para generar informes a partir de una base de datos. Se parece a un lenguaje de consultas en que permite al usuario hacer preguntas sobre la base de datos y obtener información de ella para un informe. Sin embargo, en el generador de informes se tiene un mayor control sobre el aspecto de la salida. Se puede dejar que el generador determine automáticamente el aspecto de la salida o se puede diseñar ésta para que tenga el aspecto que desee el usuario final[1].

Los generadores de “Forms” (Forms Generators), en algunos casos manejan iteraciones online con los usuarios de la aplicación o bien crean un programa para que manejen esta iteración[3]. Permiten que el usuario defina el aspecto de la pantalla, qué información se debe visualizar y en qué lugar de la pantalla debe visualizarse. Algunos generadores de formularios permiten la creación de atributos derivados utilizando operadores aritméticos y también permiten especificar controles para la validación de los datos de entrada[3].

Los Ambientes de Cuarta Generación (Fourth Generation Environments) se encargan de generar todo un sistema a partir de las salidas de herramientas CASE, pantallas, reportes y además posiblemente de la especificaron de algún proceso lógico adicional[3].

Los Administradores de Datos (Data Management) proveen comandos para la manipulación de datos, selecciones de casos y documentación de datos en la preparación de las información para análisis estadísticos y reportes. Entre ellos se encuentran SAS, SPSS y Stata[3].

Los generadores de aplicaciones son una herramienta para crear programas que hagan de interfase entre el usuario y la base de datos. El uso de un generador de aplicaciones puede reducir el tiempo que se necesita para diseñar un programa de aplicación. Los generadores de aplicaciones constan de procedimientos que realizan las funciones fundamentales que se utilizan en la mayoría de los programas. Estos procedimientos están escritos en un lenguaje de programación de alto nivel y forman una librería de funciones entre las que escoger. El usuario especifica qué debe hacer el programa y el generador de aplicaciones es quien determina cómo realizar la tarea[3].

En la próxima sección hablaremos de algunos ejemplos de 4GL específicos para algunas de estas categorías.

## **5. Algunos lenguajes de cuarta generación**

Existen muchos lenguajes de cuarta generación hoy en día, unos más usados que otros, ya sea por su portabilidad, facilidad de uso o eficiencia.

Algunos de ellos son: SheerPower4GL, PowerBuilder, WinDev, Focus, Natural, Progress4GL, SQL, Oracle Reports, PostScript, Matemática, MatLab, Ramis, entre otros.

En este apartado del artículo se presentarán algunos lenguajes de cuarta generación y se describirán las características de algunos de ellos.

### **5.1 Progress 4GL**

Progress 4GL o Progress OpenEdge como se han llamado sus últimas versiones, es un lenguaje muy utilizado pues es potable y muy confiable. Es una plataforma diseñada para ayudar a los desarrolladores en la construcción de aplicaciones empresariales de forma rápida, esto ayuda a recuperar la inversión de manera más rápida. Tiene la facilidad de fácilmente conectarse e integrarse con clientes, con otras aplicaciones y con distintas bases de datos. [9]

Esta plataforma es muy potente. “Gracias a ella los desarrolladores pueden olvidarse de las complejidades de los entornos informáticos de hoy en día, permitiéndoles concentrarse en lo que realmente importa, crear la lógica de negocio de sus aplicaciones. Los sistemas que componen la plataforma OpenEdge son compatibles, fiables y escalables, sin embargo requieren menos recursos administrativos que otras plataformas.” [10]

OpenEdge es una plataforma abierta y flexible. “Un desarrollador de OpenEdge puede crear APIs para clientes de .NET, clientes de Java™, mensajes de Java, hasta el esquema de de base de datos en formatos XML, eso, sin nunca dejar la eficiencia de Progress 4GL y de las herramientas de OpenEdge.” Traducido de [11]

Progress incluye herramientas de programación y de bases de datos, y recursos de servidores todo en su ambiente, permitiendo fácil acceso a todos los recursos para crear aplicaciones.

Progress garantiza a los usuarios disponibilidad y accesibilidad en sus bases de datos, ya que el tamaño de las tablas puede ser excesivamente grande sin problema alguno. “Además, las pruebas realizadas han mostrado que la plataforma OpenEdge tiene una escalabilidad aproximada de más de 30,000 usuarios simultáneos.” [10]

Esta garantía de calidad que da Progress, hace de este un programa muy usado en el momento, con una gran funcionalidad. Además es un programa muy simple de utilizar en comparación a otros como SQL, donde se tardaría mucho más tiempo creando una base de datos.

## 5.2 SQL (Structured Query Language)

SQL (lenguaje de consultas estructurado) es un lenguaje de acceso a bases de datos relacionales con el cual se pueden crear y manipular las mismas. SQL “es un conjunto de objetos eficientemente almacenados”[12]. La información de las bases de datos se guarda en tablas.

Una sentencia SQL es una frase con la que decimos lo que queremos obtener y de que tablas deseamos obtenerla. Una sentencia tiene un formato de cómo realizarse y empieza con una palabra reservada que indica la acción que se quiere realizar, luego siguen lo que uno quiere hacer respecto a la acción definida. SQL permite realizar consultas mediante sentencias SQL con el fin de poder desplegar información importante que se quiera de alguna o varias tablas.

SQL posee muchos servicios, entre ellos: servicio de duplicación, servicio de notificación, servicio de integración, etc. El servicio de duplicación es para mantener varias copias de la base de datos o de alguno de sus objetos. EL servicio de notificación es para enviar distintas notificaciones a uno o más dispositivos. El servicio de integración sirve para la creación de paquetes con diferentes tipos de datos. [13]

“Cabe destacar que la mayoría de los servicios se han creado teniendo presente la estrategia de Business Intelligence de Microsoft. Así que el nuevo SQL Server, más que un servidor de bases de datos, debería considerarse una plataforma completa de Windows Intelligence.” [13]

## 5.3 WinDev

WinDev es una herramienta de desarrollo desarrollada por la empresa francesa PC SOFT. Tiene la facilidad para desarrollar aplicaciones en Windows, .Net, Internet y Java. Permite el desarrollo de interfaz gráfica.

Se pueden realizar muchos tipos de aplicaciones, entre ellas: “Gestión, industriales, médicas, de terreno, de ERP, CRM, EDI, enseñanza asistida por ordenador, integración de aplicaciones empresariales, Back y Font Office, .NET, Web Services, catálogos. No hay límites.” [14]

En WinDev la calidad de las aplicaciones dependen menos del equipo de desarrollo que con otras herramientas, esto debido a que trae un conjunto de funciones avanzadas sin la necesidad de que alguien las programe, por ejemplo, puede ser que el entorno detecte que mejoras para aumentar el rendimiento y la velocidad del sistema y este mismo las sugiere y las realiza automáticamente, además, posee una herramienta generadora de reportes automática. [14]

WinDev tiene muchas herramientas para usarse en su IDE, entre ellas están: editor de reportes, editor de componentes, editor de pruebas, editor de ayuda en línea, editor de ventanas, editor de consultas, editor de código, etc. Su IDE es muy completo. [14]

WinDev además de permitir el desarrollo de código y programas en distintos lenguajes, permite leer y escribir en todas las bases de datos. Es un programa muy completo porque permite desarrollar cualquier aplicación.

Su crecimiento debe a su gran portabilidad y su facilidad de uso para los usuarios.

## 5.4 PowerBuilder

PowerBuilder es una herramienta de desarrollo desarrollada por la empresa Sybase. Es un entorno gráfico de programación orientado a objetos para el desarrollo de aplicaciones cliente/servidor, distribuidas y web. Incluye herramientas para generar reportes, acceder bases de datos y para crear interfaz gráfica.

El lenguaje de escritura es PowerScript. Una escritura consiste en realizar las declaraciones que realizan el proceso en respuesta a un evento. [15]

## 5.5 Informix 4GL

Es el líder para manejar aplicaciones cliente que acceden a las bases de datos de Informix en el ambiente de Unix. Es una estructura basada en ventanas y caracteres orientados que permite a desarrolladores preparar o directamente incrustar en un código fuente de un lenguaje de cuarta generación determinado la estructura estándar de consultas SQL. Desde el código fuente del 4GL el pre-procesador del Informix genera pseudo código (p-código) de máquina o código C dependiendo de cual haya sido el 4GL elegido en tiempo de compilación. [5]

Para producir las aplicaciones se utilizan tres diferentes productos: el Compilador, el Sistema Rápido de Desarrollo y el Depurador Interactivo. El compilador genera código C y código ESQL/C que puede ser compilado en el código objeto, el sistema rápido de desarrollo reduce en gran manera el tiempo de compilación creando el pseudo código y el depurador ayuda a la detección y rápida corrección de errores dentro de la aplicación.

La gran utilidad del Informix lo han colocado como líder dentro del mercado para el desarrollo de aplicaciones para bases de datos, pero debido a que su lanzamiento fue en 1986 muchas aplicaciones creadas utilizándolo tal vez ya son muy viejas y necesitan adaptarse a las nuevas necesidades como la compatibilidad y uso de interfaces gráficas para el usuario. [5]

Por lo anterior es que se ha desarrollado una nueva versión de Informix denominado Informix Dynamic 4GL que permite migrar hacia las nuevas necesidades del mercado con un simple recompilación del programa lo que facilita en gran medida el proceso de adaptación a las nuevas tecnologías. Lo anterior permite que los costos se reduzcan y retiran la necesidad de un rediseño o reestructuración de un programa, ya que es tan fácil como presionar un botón y así migrar a diversas plataformas.

La nueva versión de Informix también combina otras ventajas como, un consumo menor de almacenamiento reduciendo el costo en hardware, también contempla una gran flexibilidad, una reducción en los requerimientos de memoria (en tipo de ejecución se cargan solo los módulos de p-código que se están usando actualmente) [5] y una portabilidad muy grande ya que puede montarse sobre muchas plataformas.

## 5.6 Mathematica

Es un ambiente general de computación que vino empaquetar en un solo producto una serie aplicaciones de tareas de computación técnica que aunque ya habían sido implementadas nunca se habían contemplado para formar un gran paquete de computación. [7]

En Mathematica se contemplan muchos de los aspectos técnicos de la computación como el manejo numérico, la conversión de datos, la visualización y la creación de interfaces para el usuario. El avance intelectual que hizo posible el desarrollo de un paquete tan completo fue la invención de un lenguaje que fuera capaz de manipular la gran cantidad de objetos que alberga la computación técnica.

Por su completitud es un paquete que a pesar de inicialmente ser usado por técnicos ha pasado a ser un ambiente manejado por gran cantidad de personas que han aprendido desplegar todas las utilidades que el programa ofrece como por ejemplo los estudiantes a los que les permite aprender de manera interactiva.[6]

El crecimiento de esta herramienta se debe también a que permite diseñar y analizar diferentes comportamientos que pueden tener ciertos procedimientos, ya que con su interfaz hace muy fácil la contemplación de detalles que tal vez no serian tan claros de otra manera.

Su crecimiento se debe también a su gran portabilidad y nivel para interactuar con otros programas como Excel, Matlab y los más importantes lenguajes de programación. Su funcionalidad es muy grande debido al manejo tan exacto de los conjuntos numéricos y su capacidad de generar gráficos que hacen el análisis. [6]

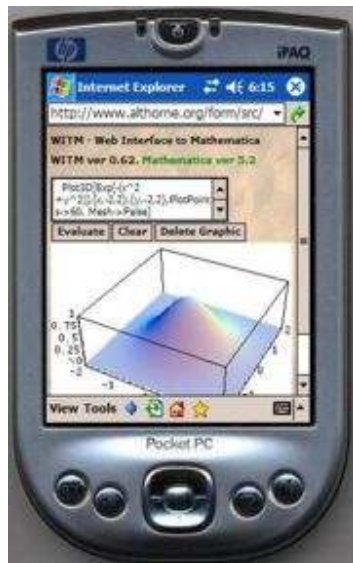


Figura2. Imagen de un PDA HP con el kernel de Mathematica[7]

## 6. Conclusiones:

La evolución de los lenguajes tiende cada vez más a alejarnos de la máquina o hardware, creando una mayor abstracción de los problemas a resolver, esto es beneficioso pues genera un ahorro significativo de recursos como el tiempo que es tan valioso actualmente.

Los Lenguajes de Cuarta Generación tienden a ser muy compatibles entre sus mismas evoluciones lo que nos permite crear aplicaciones con la confianza de que el trabajo realizado no será desechado más adelante, tal es el caso de Informix.

Paquetes tan poderosos como Mathematica hacen posible que las técnicas de computación mejoren constantemente pues brindan una mayor facilidad para el análisis y diseño de nuevas herramientas, mientras también ayudan a áreas tan importantes como la educación, todo esto empleando la misma herramienta.

Es importante resaltar que para utilizar un 4GL se debe tener claro que si se desea manipular para sacarle un mayor rendimiento, se debe de hacer cambiando la forma normal de hacer software. Para esto, los programadores deben de volverse analistas, deben dominar técnicas estructuradas, conceptos de diseño de interfaz gráfica, conceptos de arquitectura, conceptos de orientación a objetos y de principios de diseño. Y todo esto para poder obtener una mayor productividad, una mayor facilidad al dar mantenimiento y además una mejor apariencia de la aplicación.

## 7. Bibliografía:

- [1] [Zamora Cortina, Luis. "Conceptos Básicos sobre: CASE Lenguajes de Cuarta Generación", 2003](#)
- [2] [9 Fourth Generation Environments.htm](#)
- [3] [http://en.wikipedia.org/wiki/Fourth-generation\\_programming\\_language](http://en.wikipedia.org/wiki/Fourth-generation_programming_language)
- [4] [PDF lenguajes de Cuarta generación](#)
- [5] [http://www-07.ibm.com/software/kr/data/informix/product/pdf/d4gl\\_wp.pdf](http://www-07.ibm.com/software/kr/data/informix/product/pdf/d4gl_wp.pdf)
- [6] <http://www.wolfram.com/products/mathematica/history.html>
- [7] <http://en.wikipedia.org/wiki/Mathematica>
- [8] <http://www.ie.inf.uc3m.es/grupo/Investigacion/LineasInvestigacion/Articulos/cw94-4gl.doc>
- [9] <http://www.progress.com/openedge/index.ssp>
- [10] [http://www.progress.com/progress\\_software/worldwide\\_sites/es/docs/sp\\_openedge10\\_0bplatform.pdf](http://www.progress.com/progress_software/worldwide_sites/es/docs/sp_openedge10_0bplatform.pdf)
- [11] [http://www.progress.com/progress\\_software/products/develop/docs/oe10\\_4gl\\_development.pdf](http://www.progress.com/progress_software/products/develop/docs/oe10_4gl_development.pdf)
- [12] <http://www.formaselect.com/curso/experto-en-sql-server-2000/Introduccion-a-SQL-Server%202000.pdf>
- [13] <http://www.danysoft.com/free/03457sql.pdf>
- [14] <http://www.pcsoft-windev-webdev.com/WD11folleto.pdf>
- [15] <http://alfa.facyt.uc.edu.ve/computacion/pensum/cs0347/Laborat/ManualPB.pdf>
- [16] <http://www.ie.inf.uc3m.es/grupo/Investigacion/LineasInvestigacion/Articulos/cw94-4gl.doc> -