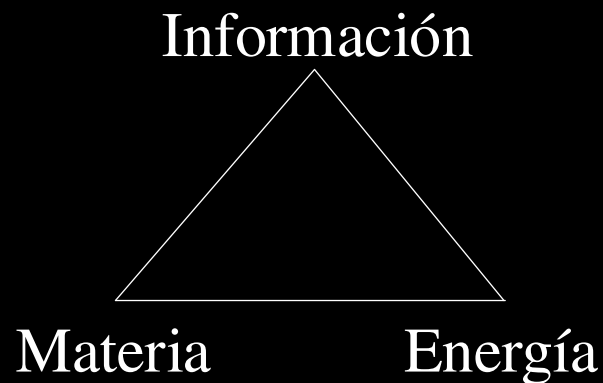


De los Generadores de Programas a los Sistemas Evolutivos



REDI

Fernando Galindo Soria

fgalindo@ipn.mx

www.fgalindosoria.com

Abril del 2012, Junio del 2014

MÉXICO

En este trabajo se presenta una introducción a los generadores de programas, a los generadores de sistemas y su relación con los sistemas evolutivos.

Incluyendo su desarrollo histórico desde 1975.

Y específicamente se muestra como construir generadores de programas, generadores de sistemas y otras herramientas.

Y como desarrollar productos basados en esas herramientas.

Cuando se hace algo varias veces, llega un momento que se detecta que no tiene sentido seguir repitiendo lo mismo

Si se hace algo varias veces conviene delegar la actividad (se les explica a otros como hacerlo y se les delega)

o se hace un programa que lo haga

O se hace un generador de programas



Generador de Programas

Fernando Galindo Soria 1975

En 1975 en El Colegio de México se desarrolló un proyecto que requería encontrar la relación entre aproximadamente 1000 variables (se tenía que encontrar la relación entre cada variable con las otras 1000),

por el tipo de herramienta (paquete estadístico) que se usaba, se tenía que hacer un programa para encontrar la relación entre la primera variable y la segunda,

otro para encontrar la relación entre la primera variable y la tercera, otro para la primera y la cuarta, y así sucesivamente para la primera y todas las demás (o sea alrededor de 1000 programas),

pero también se requería hacer 1000 programas para encontrar la relación entre la segunda variable y cada una de las otras,

y lo mismo para la tercera variable y la cuarta y para todas las otras, o sea que en total se requería hacer alrededor de 1000,000 (un millón) de programas,

claramente era muy poco factible hacer esa cantidad de programas, por lo que se construyó un Generador de Programas, que tomaba cada variable y la combinaba con todas las otras, generando todos los programas y mandándolos ejecutar.

*Básicamente el **Generador de Programas** tenía en un archivo una copia del programa a generar (que era el mismo para el millón de programas que se necesitaban) y en una tabla (Diccionario de Datos) la lista de las mil variables que se requerían combinar, incluyendo la posición de cada variable en el archivo donde estaban almacenadas.*

*El **Generador de Programas** tomaba cada una de las variable junto con su posición dentro del archivo, las incrustaba en el programa a generar generaba cada programa en el lenguaje del paquete estadístico,*

como esta información se generaba en el lenguaje del paquete estadístico, se tenía que llamar al paquete estadístico para que se ejecutara y poder obtener los resultados,

*pero dado que no se quería que una persona tuviera que mandar a ejecutar cada programa a mano, se realizó un **Ejecutor de Programas** que tomaba el nombre del archivo generado por el generador de programas y lo mandaba ejecutar,*

*finalmente se realizó un **Generador de Sistemas** que mientras existían variables a combinar en la tabla de variables llamaba al generador de programas, que generaba un programa en el lenguaje del paquete estadístico, luego llamaba al **Ejecutor de Programas**, que mandaba ejecutar el programa y obtenía los resultados.*

*Para realizar lo anterior y dado que no era simple en esa época mandar ejecutar programas en forma directa, sin que alguien lo hiciera a mano, el **Generador de Sistemas** llamaba y coordinaba en forma automática varias herramientas para resolver el problema, incluyendo: lenguajes de alto nivel, paquetes estadísticos, editores de texto, lenguajes ensambladores y lenguajes de administración de sistemas operativos, entre otros.*

*tabla de variables
(Diccionario de Datos)*

Generador de Sistemas

Generador de Programas

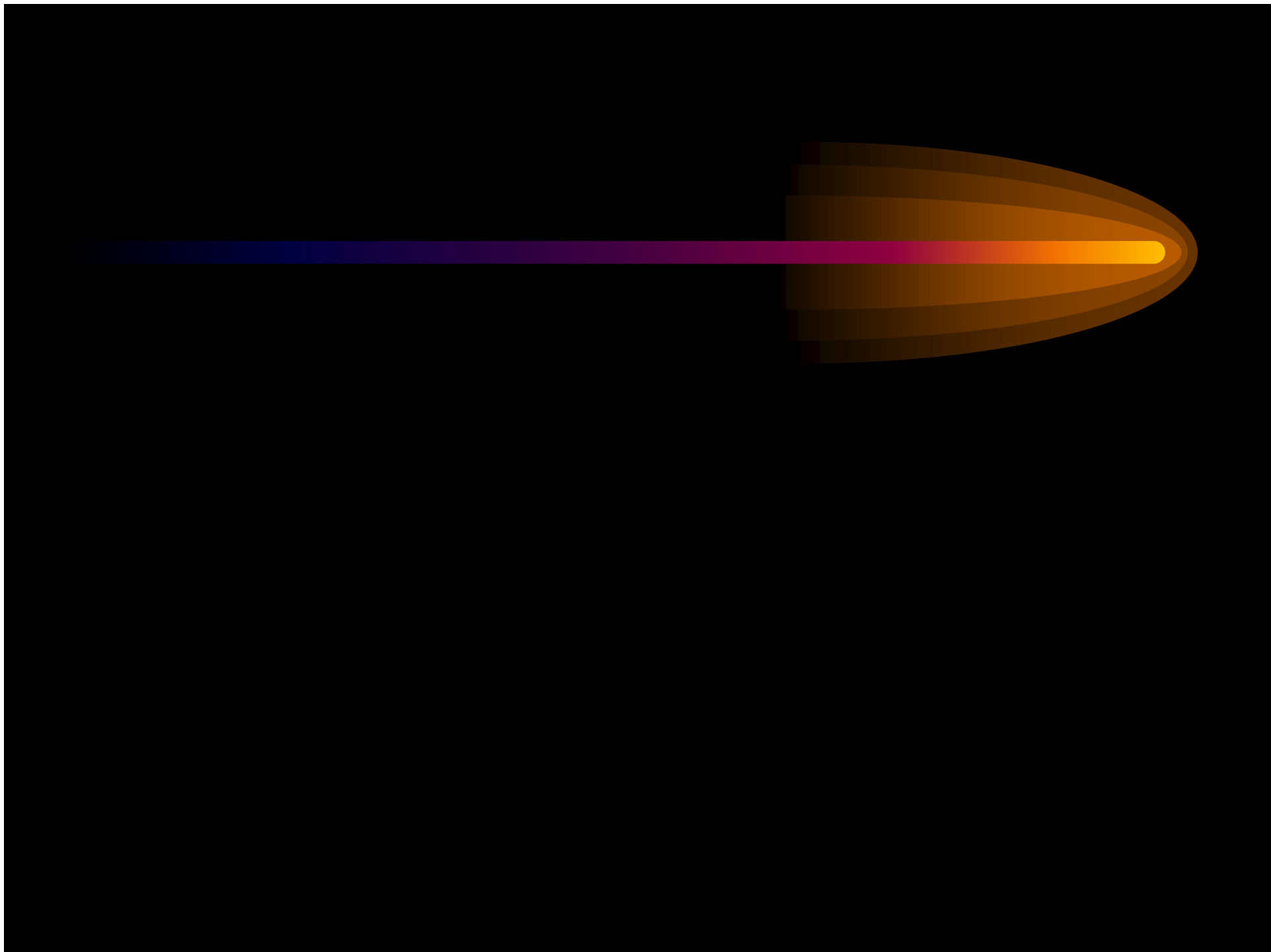
*programa generado
en el lenguaje del
paquete estadístico,*

Ejecutor de Programas

Paquete Estadístico

*Resultado
relación entre variables*





Entre los años 70's y principios de los 80's un equipo integrado por Vicente López Trueba, Christian Zempoaltecatl Ibarra, Ricardo García, Juan Martín González Vázquez y otros, desarrollaron una serie de herramientas incluyendo: Generadores de Programas, Generadores de Sistemas, Documentadores Automáticos, Migradores de Programas entre diferentes plataformas, Sistemas Evolutivos, etc.,

estas herramientas se desarrollaron, usaron y aun se siguen usando para resolver problemas a nivel de Secretarías de Estado, ya que en esa época ellos eran los responsables del desarrollo de los sistemas de diferentes Secretarías.

Ya para principios de los 90's, Vicente López Trueba desarrollaba e instalaba Sistemas Evolutivos basados en estas herramientas en toda América y dentro de proyectos del Banco Mundial y de Naciones Unidas.


Actualmente seguimos desarrollando y usando estas herramientas, ya que, lograron llegar a conceptos muy sencillos para realizar herramientas muy poderosas.

A continuación se mostrara como se pueden construir algunas de estas herramientas, para lo cual y en general, solo se requiere conocer algún lenguaje de programación y como se manejan caracteres, cadenas de caracteres y archivos en ese lenguaje.



Generadores de Programas

Vicente López Trueba Finales 70's



Cuando se hace algo varias veces, llega un momento que se detecta que existen cosas que se repiten y cosas que cambian de una versión a la siguiente

Se toma un programa, por ejemplo el de la suma

Lee a, b

c=a+b

Escribe c

Se toma otro programa también de la suma

Lee x, y

z=x+y

Escribe z

Se comparan los programas y se detecta lo que cambia

*Lee **a**, **b***

c**=**a**+**b

*Escribe **c***

*Lee **x**, **y***

z**=**x**+**y

*Escribe **z***

Se comparan los programas y se detecta lo que cambia

*Lee **a**, **b** Lee **x**, **y***

c**=**a**+**b** **z**=**x**+**y

*Escribe **c** Escribe **z***

*Se sustituyen los elementos que cambian en el programa por una etiqueta **\$1**, **\$2**, **S3***

*Lee **\$1**, **\$2***

S3**=**\$1**+**\$2

*Escribe **\$3***

*A los elementos que cambian se les llama **parámetros***

Estamos usando el símbolo \$ para indicar parámetro, pero se puede usar cualquier simbolo que no se use dentro del sistema (como #, “, etc.)

Se comparan los programas y se detecta lo que cambia

*Lee **a**, **b** Lee **x**, **y***

c**=**a**+**b** **z**=**x**+**y

*Escribe **c** Escribe **z***

*Se sustituyen los elementos que cambian en el programa por una etiqueta **\$1**, **\$2**, **S3***

*Lee **\$1**, **\$2***

S3**=**\$1**+**\$2

*Escribe **\$3***

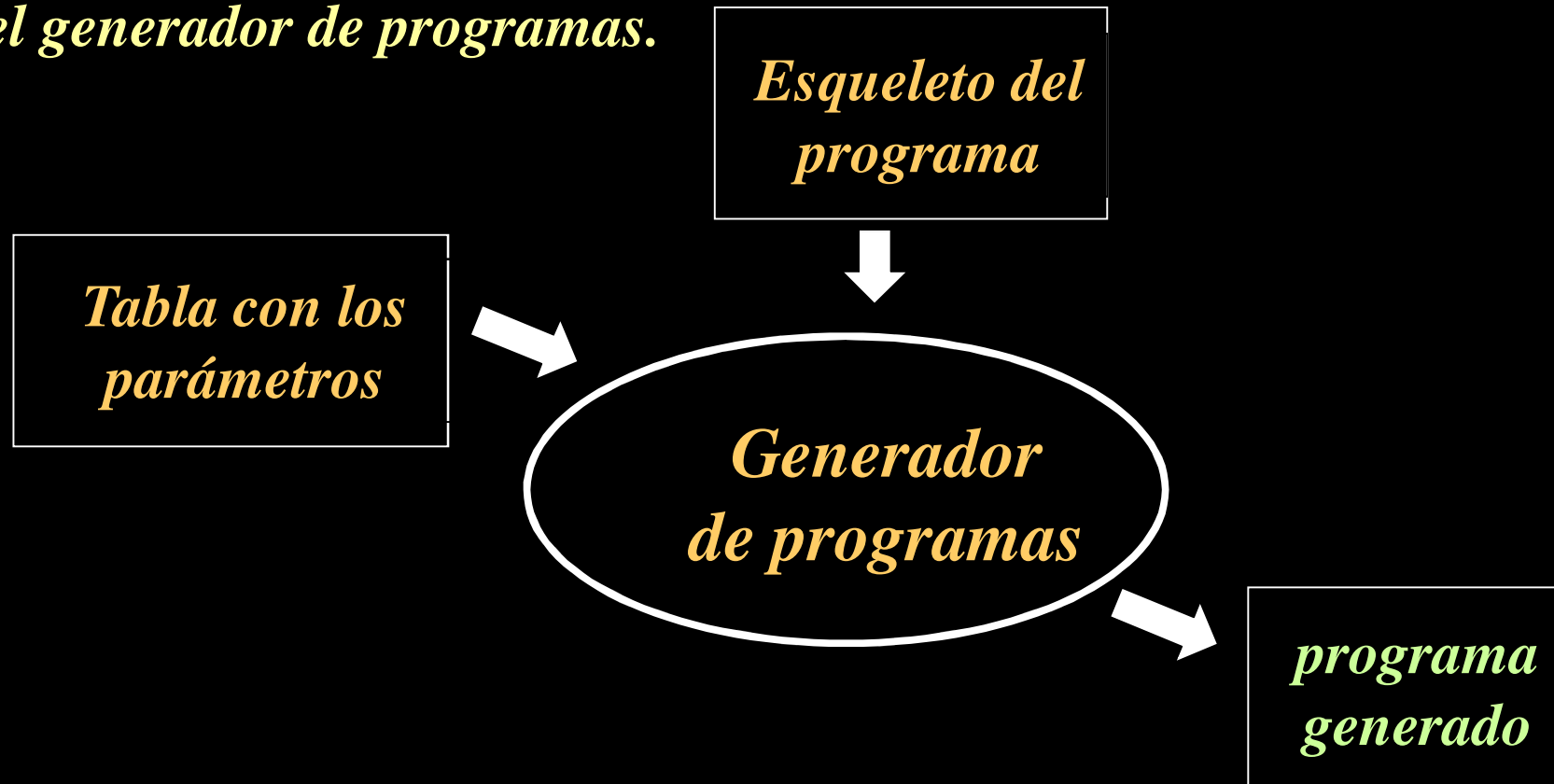
esqueleto

A los elementos que cambian se les llama parámetros

*al programa con las etiquetas se le llama esqueleto y a este proceso se le llama **Parametrización***

Generador de Programas

El sistema generador de programas consta de 4 componentes: el esqueleto del programa que se quiere generar, una tabla con los parámetros que se van a usar en este caso particular, un archivo donde se almacena el programa generado, y el generador de programas.



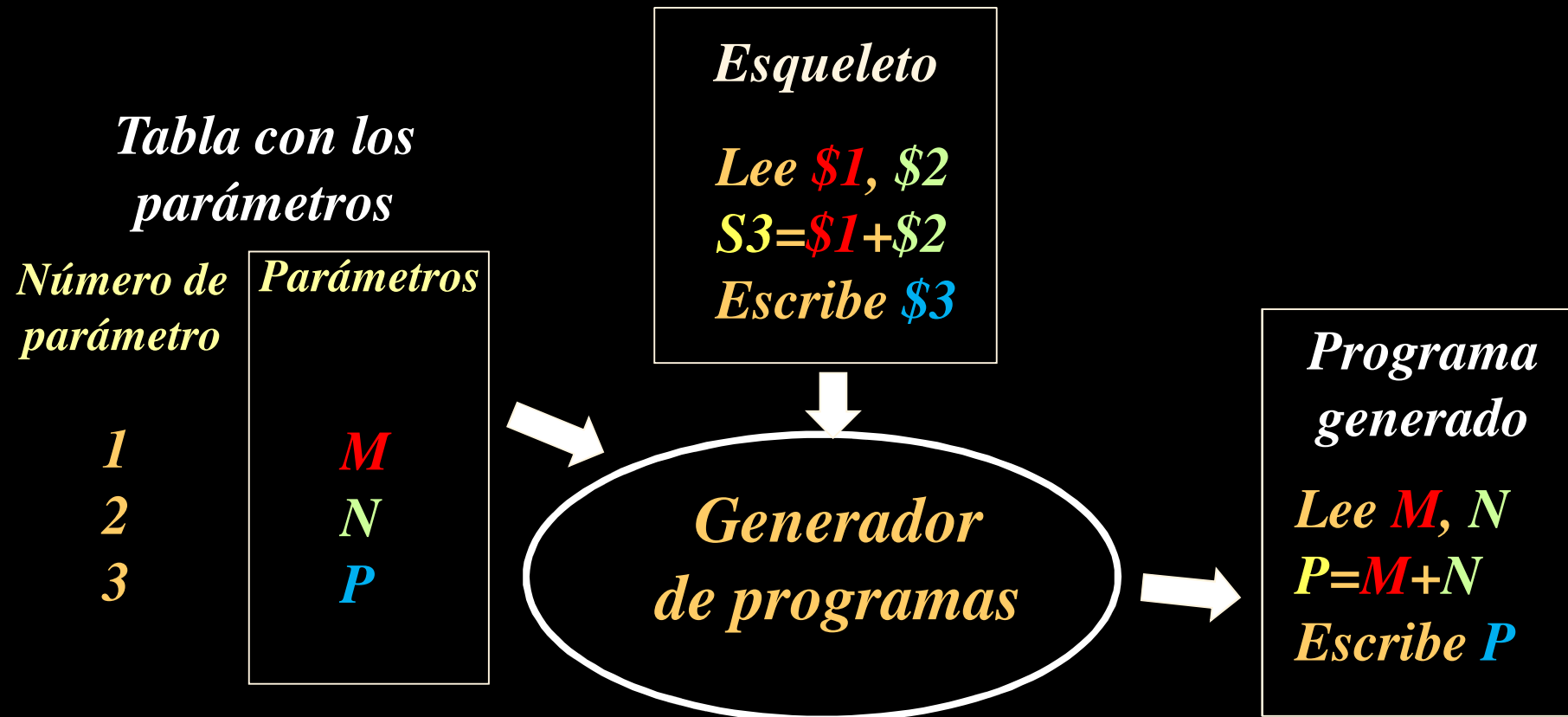
Generador de Programas

Por ejemplo si se quiere hacer un programa que sume M con N y ponga el resultado en P :

se toma el esqueleto de la suma;

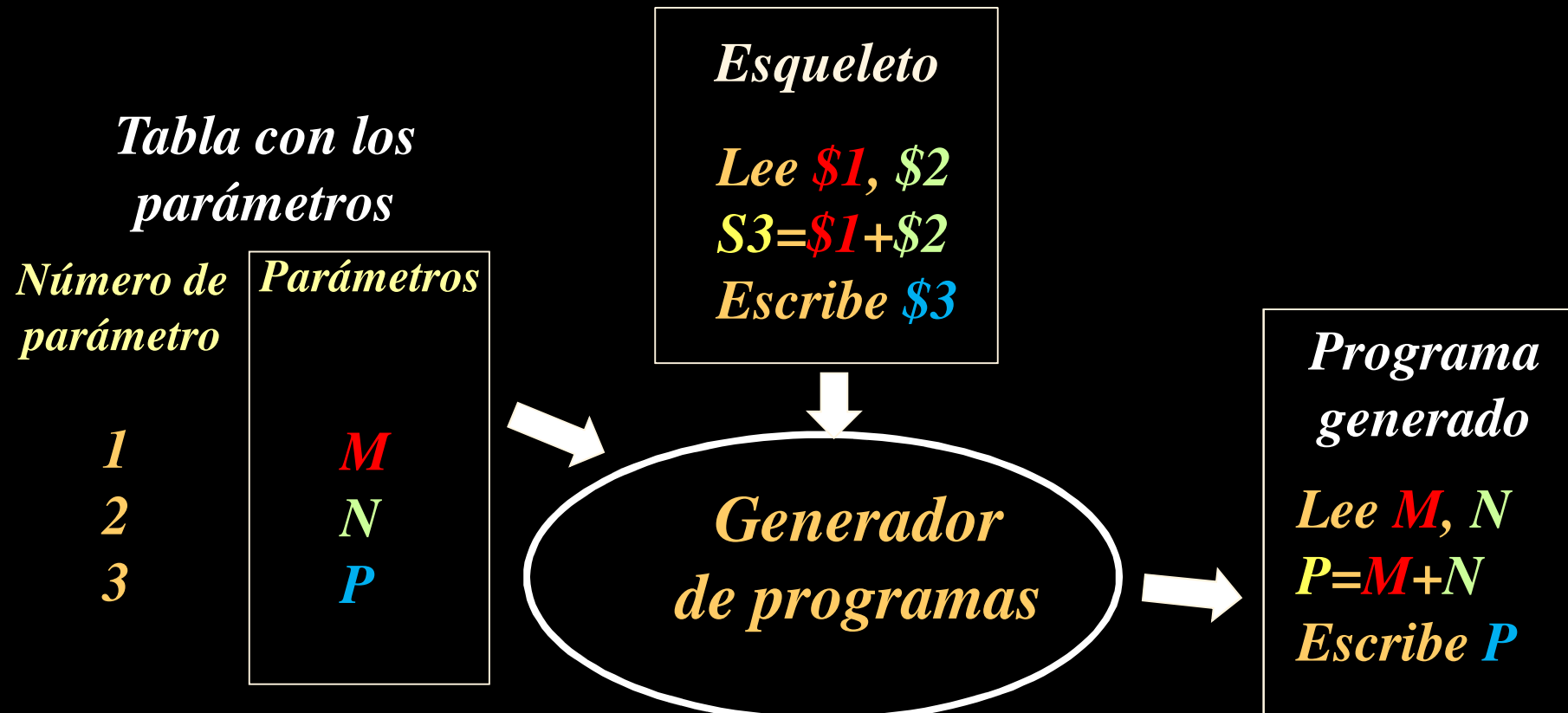
se colocan M , N y P en la tabla de parámetros;

y se llama al generador de programas.



Generador de Programas

El generador de programas comienza a leer el código que aparece en el esqueleto y si no es un parámetro lo reescribe en la tabla del Programa generado, pero si es un parámetro (o sea que tiene el carácter \$) entonces lee el número del parámetro, va a la tabla y toma ese parámetro y lo escribe en el programa generado)



Generador de Programas

Esqueleto

Lee \$1, \$2
S3=\$1+\$2
Escribe \$3

Tabla con los parámetros

Número
de
parámetro

1
2
3

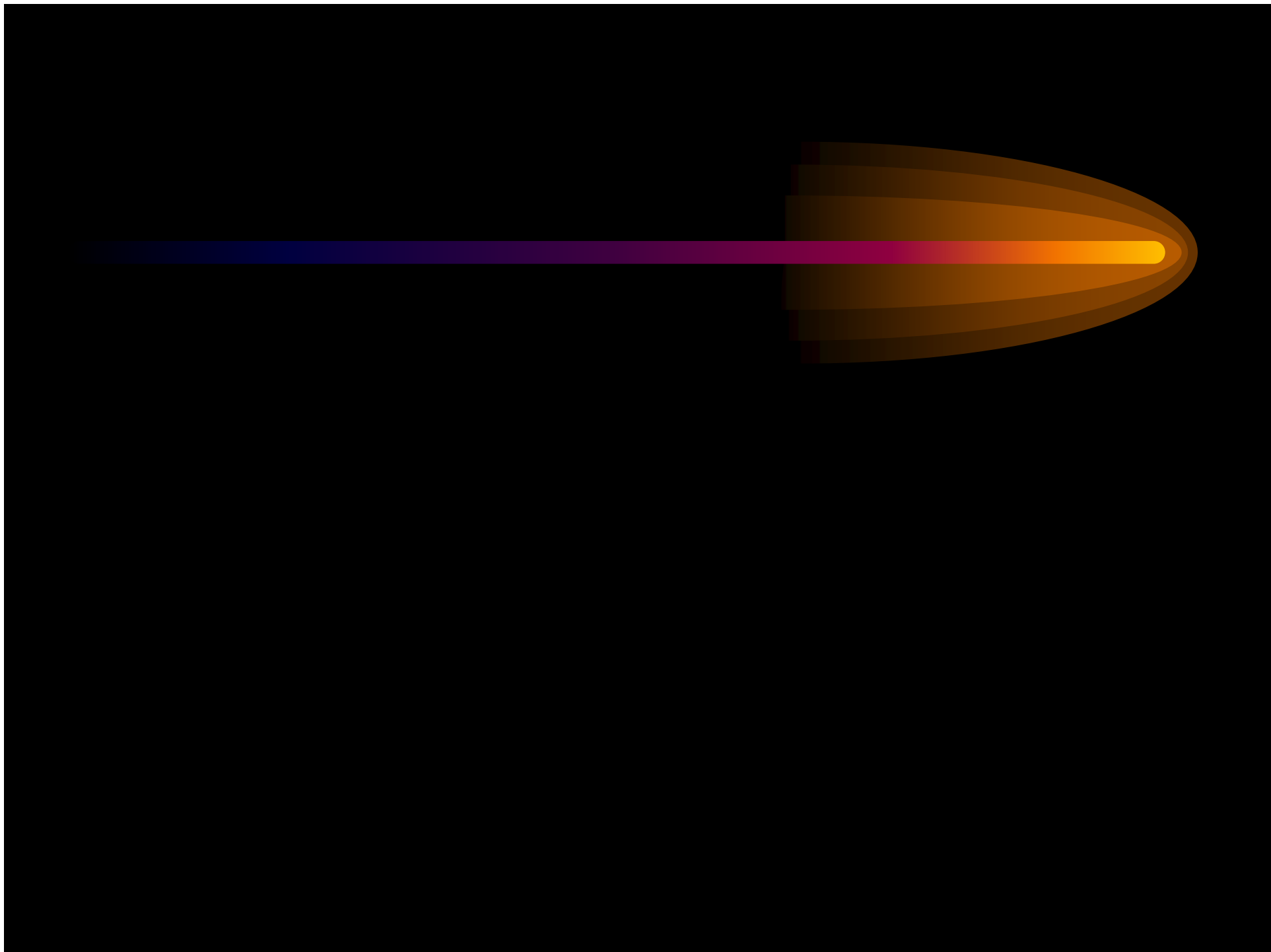
Parámetros

M
N
P

Lee siguiente carácter del Esqueleto
Si no es parámetro (si no es el carácter \$)
Escribe el carácter en Programa Generado
Si es parámetro (si es el carácter \$)
Lee el número de parámetro del Esqueleto
Toma el parámetro de la Tabla de parámetros
Escribe el parámetro en Programa Generado
Mientras no fin de archivo Repite el ciclo

Programa generado

Lee M, N
P=M+N
Escribe P





Generadores de programas

Generadores de sistemas

Sistemas evolutivos



Generadores de Programas

Generadores de Sistemas

Vicente López Trueba Finales 70's

A partir de un Generador de Programas es relativamente fácil construir un Generador de Sistemas

En principio podemos ver un sistema como un conjunto de programas interrelacionados,

por lo que para construir un generador de sistemas, lo primero que se hace es detectar los programas que forman el sistema

cada programa se parametriza y se obtiene su esqueleto,

luego mientras existan programas a generar, el generador de sistemas llama al generador de programas que llama a cada programa y le asocia los datos particulares del sistema que se quiere generar

El único detalle que falta es como indicarle al generador de sistemas cual es el siguiente programa a generar





Generadores de Programas

Generadores de Sistemas

Vectores Teóricos

Vicente López Trueba Finales 70's

Vectores Teóricos

En una versión muy preliminar, un vector teórico es una lista de los programas que forman el sistema en el orden en que se deben generar

Por ejemplo si el sistema esta formado por los programas

A (pantalla de captura),

B (validación de datos),

C (calculo de resultados),

D (presentación de resultados)

El Vector Teórico quedaría como la siguiente tabla

A	B	C	D
---	---	---	---

Vector Teórico

A

B

C

D

Esqueleto

Esqueleto

Lee \$1, \$2

S3=\$1+\$2

Escribe \$3

Generador de Sistemas

Mientras existan programas en el vector teórico

Llama al generador de programas

Generador de Programas

Lee siguiente carácter del Esqueleto

Si no es parámetro (si no es el carácter \$)

Escribe el carácter en Programa Generado

Si es parámetro (si es el carácter \$)

Lee el número de parámetro del Esqueleto

Toma el parámetro de la Tabla de parámetros

Escribe el parámetro en Programa Generado

Mientras no fin de archivo Repite el ciclo

*Tabla con
los
parámetros*

1 M

2 N

3 P

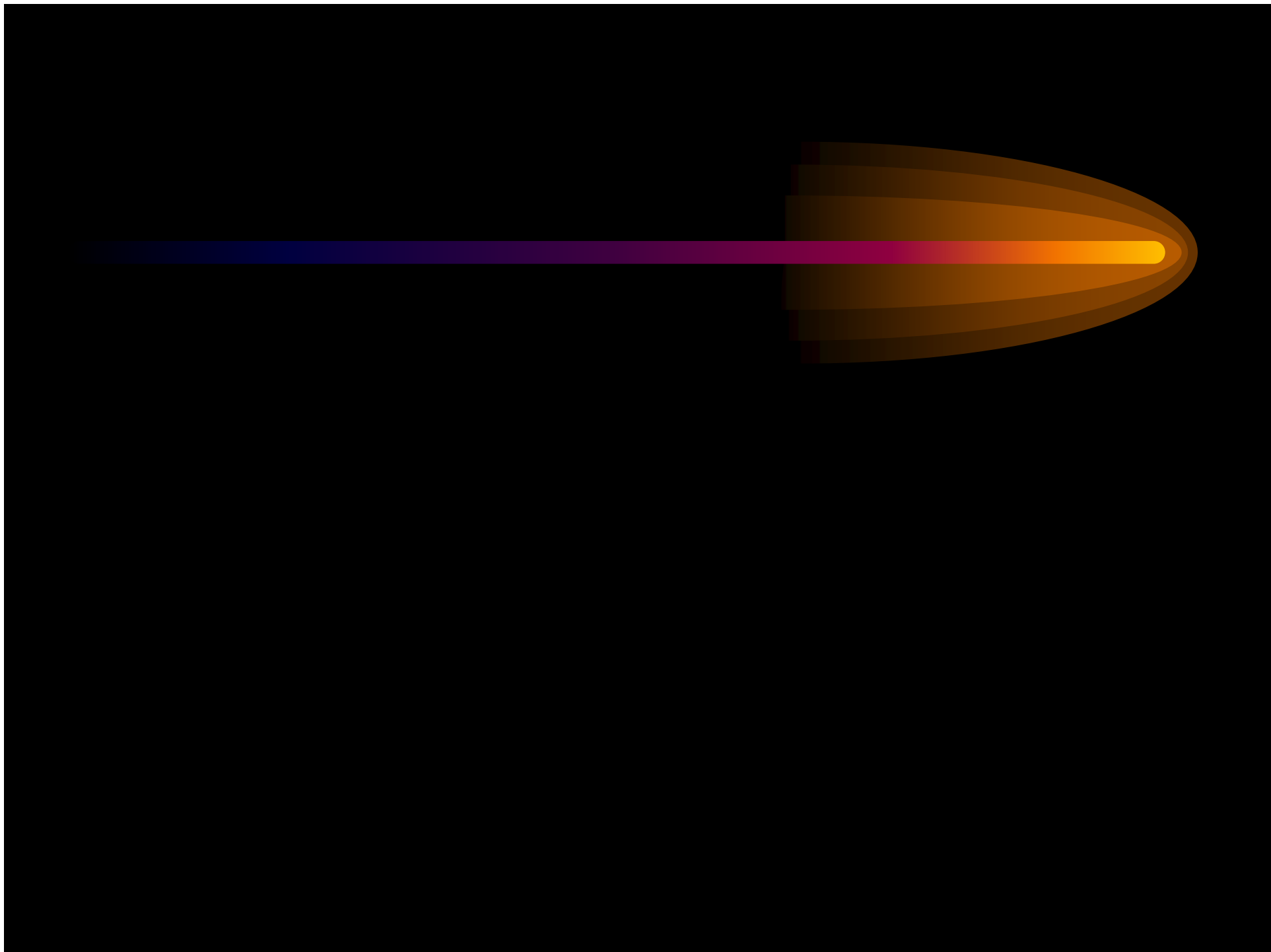
*Sistema
generado*

A

B

C

D



Generador de Programas

Generador de Sistemas

Vectores Teóricos

Gramáticas

Gramática



*Representación lingüística de la
estructura de un lenguaje*

Enfoque Lingüístico y Lingüística Matemática
<http://www.fgalindosoria.com/lingüisticamatematica/index.htm>

Enfoque Lingüístico

**una forma de ver la realidad,
en la cual se considera que,**

*cualquier cosa se puede ver como una oración de
algún lenguaje.*

Mediante el enfoque lingüístico se pueden representar como oraciones de algún lenguaje los textos, sonidos, las señales de voz, las imágenes, las reglas de un sistema experto, las trayectorias de un planeta, el movimiento de la mano, los sentimientos, la trayectoria que sigue una pieza de ajedrez al moverse, un carácter chino, una huella digital, la señal de un electrocardiograma, etc.

Con lo que se amplía el concepto de lenguaje que normalmente se restringía a los lenguajes naturales (como el Español, Inglés, Chino o Árabe) y artificiales (como Fortran, Pascal o C), para incluir cualquier cosa.

Por lo que, aunque en algunos casos las técnicas que manejamos las aplicamos solo para algún tipo de oración específico, como lenguaje natural o imágenes, se pueden aplicar a otros tipos de oraciones.

Programación Dirigida por Lenguaje

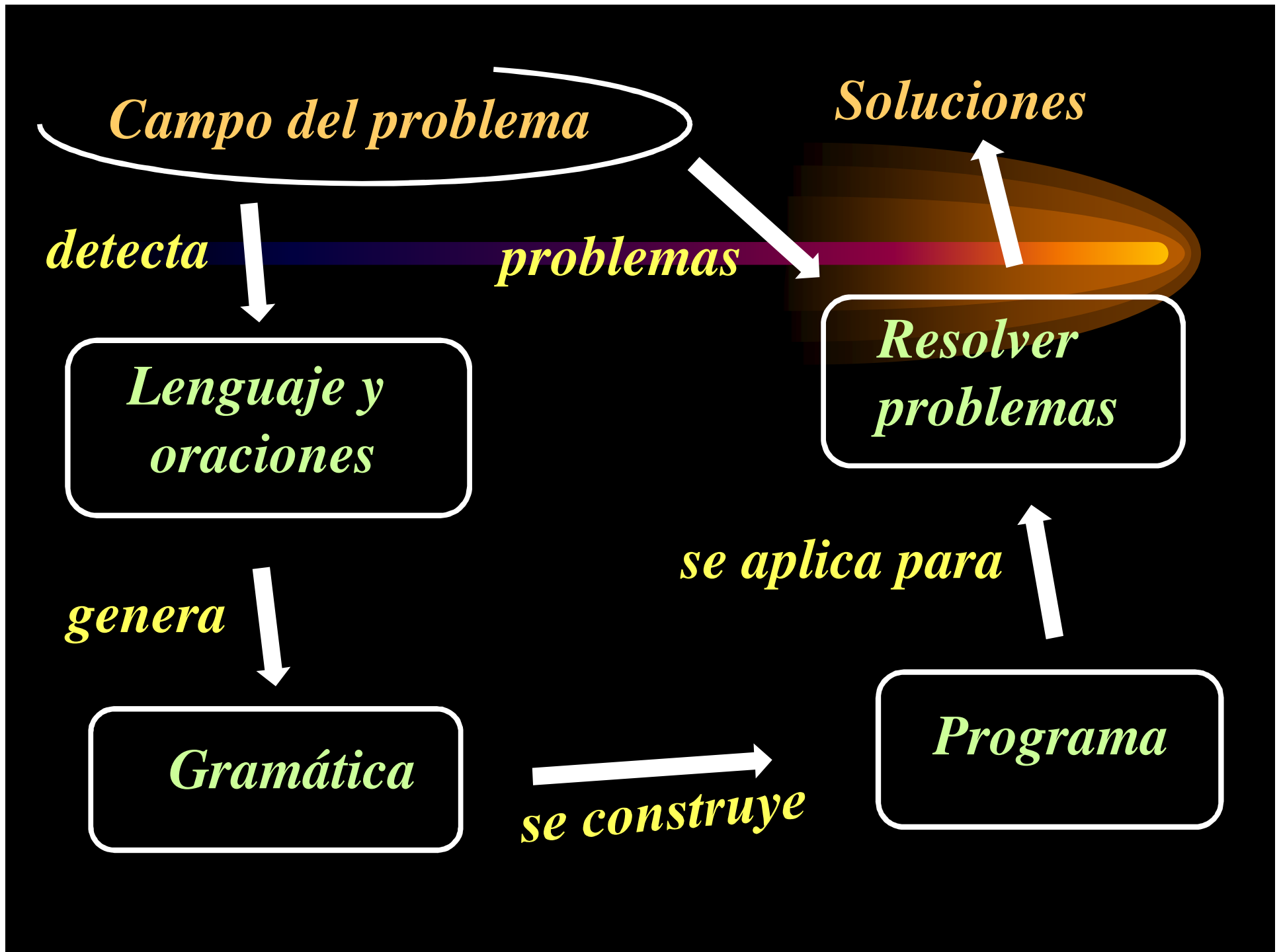
1975 - 1985

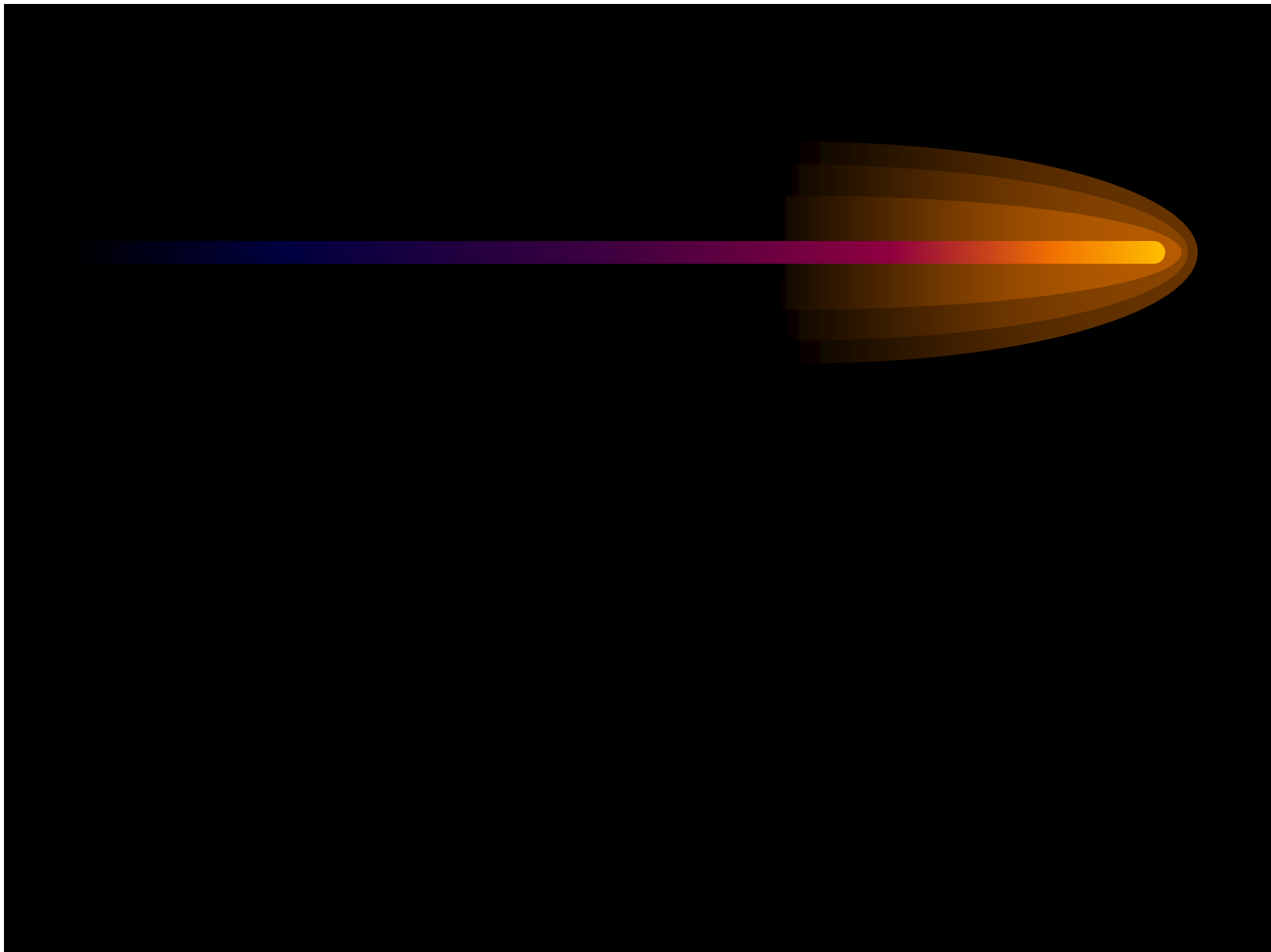
Los problemas informáticos en general son susceptibles de un tratamiento lingüístico, por lo que:

dado un problema se busca el lenguaje que lo representa

de ahí se encuentra su gramática

y de ahí el sistema de información.





Las aplicaciones de este tipo de herramientas son múltiples, por ejemplo:

Generadores de Programas y Sistemas

Validadores de datos

Generadores de Productos

Migradores Automáticos

Documentadores Automáticos

Etc., etc.

Desde 1975 hemos desarrollado una gran cantidad de sistemas, basados en la representación lingüística y mediante gramáticas de diferentes problemas

Donde precisamente se establecen los principios del Enfoque Lingüístico, la Programación Dirigida por Lenguaje y su relación con las gramáticas

Sistema validador de datos

Guillermina Pérez Salas

1975



Por ejemplo, en 1975 Guillermina Pérez Salas desarrollo un Sistema validador de datos, para lo cual represento los registros y todos sus campos como oraciones de un lenguaje, donde represento las características de los campo.

Validar los campos de un archivo o tabla puede ser desde algo simple hasta algo muy complicado, por ejemplo, verificar si los datos de un campo son números enteros es relativamente fácil,

pero en el otro extremo se tiene problemas donde los datos de un campo pueden ser, combinaciones de letras y números en ciertas posiciones específicas, o pueden depender de los valores que se toman en otros campos, o pueden ser resultados de alguna operación matemática entre varios campos, por ejemplo podría ser que los valores que toma el campo X depende de que el producto del campo M y el campo N es mayor o menor de un número dado, o de que el campo Y es F (femenino) o M (masculino), y así como esos existen muchísimos otros problemas,

precisamente es aquí donde:

se ven los campos de la tabla o archivo como oraciones de un lenguaje,

se construye el programa asociado al lenguaje y

se usa para validar los campos del archivo.

*Campos de la
tabla o archivo*

*Datos de los
Campos validados*

detecta

problemas

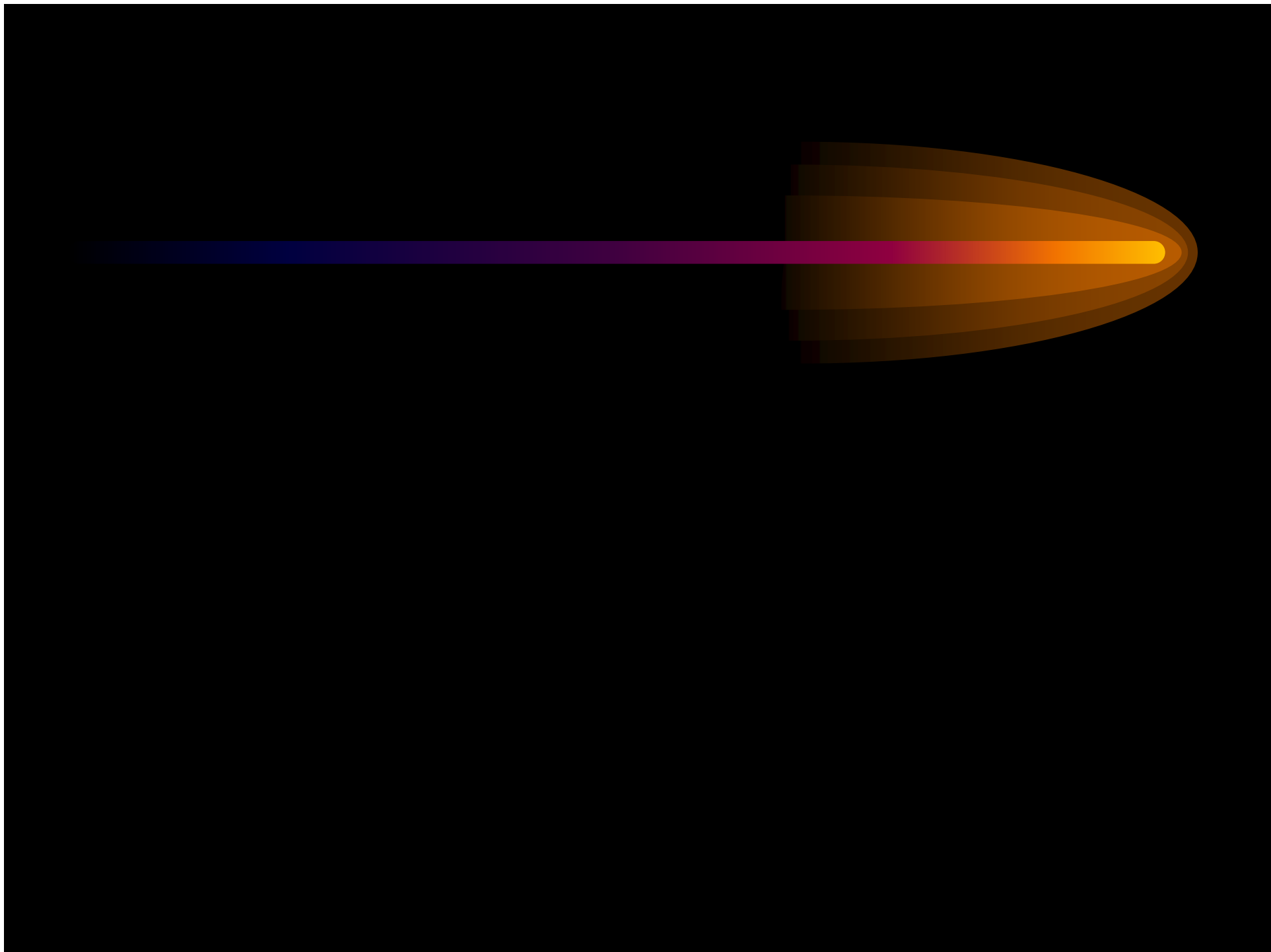
*Oraciones
del lenguaje*

*Validar
los campos
del archivo*

se construye


*Programa
Asociado
al lenguaje*

se aplica para





**Desarrollo de productos para
aplicaciones específicas**



El Generadores de Sistemas es una plataforma o sea una herramienta con la cual se pueden desarrollar otras herramienta, productos y sistemas

Ya que se tiene un Generadores de Sistemas es relativamente fácil construir productos,

ya que lo que cambia de un producto a otro son: la tabla de parámetros, el vector teórico y los esqueletos (aun mas para muchos productos los esqueletos y sus parámetros son prácticamente los mismos) pero el generador de sistemas ya esta hecho y es el mismo.

O sea que *el Generador de Sistemas se construye una sola vez y con el se pueden construir una gran cantidad de productos específicos.*

Cuando se tiene un Generador de sistemas no tiene sentido estar programando la misma aplicación una y otra vez.

El generador de sistemas funciona como una plataforma de desarrollo, a la que se le tiene que proporcionar la tabla de parámetros, los esqueletos y el vector teórico del sistema específico que se quiere convertir en un producto.

Con esta información el Generador de Sistemas genera el producto o sea una herramienta que solo se tiene que llamar e instalar en el entorno específico del usuario para que funcione,

o sea que en lugar de desarrollar una y otra y otra vez la misma aplicación para diferentes usuarios, simplemente se instala el producto en el entorno de cada usuario.

Con la

Plataforma

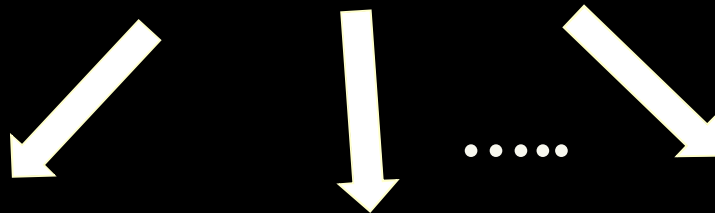
se pueden desarrollar múltiples productos, herramienta, sistemas y aplicaciones.



Con el

Producto **Producto** **Producto** **Producto**

se pueden desarrollar múltiples sistemas o aplicaciones



Sistema o
Aplicación **Sistema o**
Aplicación **Sistema o**
Aplicación

Cuatro Niveles de Empresas

1.- Servicios, sistemas, aplicaciones, consultorias,...

Inversión 1 -> Ganancia 1 a 10

2.- Productos

Inversión 1 -> Ganancia 10 a 1000

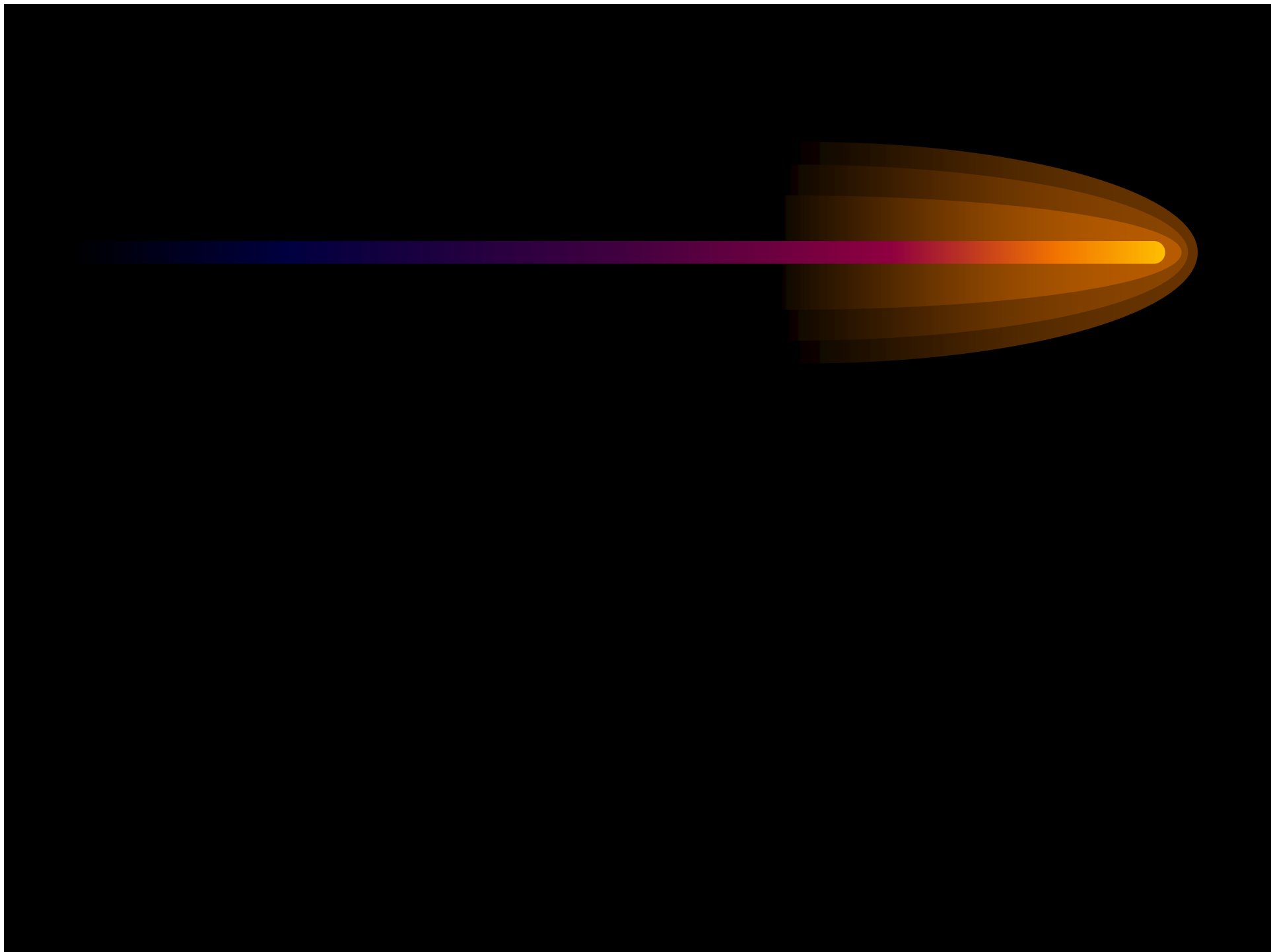
La idea es llegar a millones de copias

3.- Plataformas

Inversión 1 -> Ganancia 100 a 10,000

4.- Base

Inversión 1 -> Ganancia 1000 a ...





**Sistema Evolutivo Generador de
Nominas Federales**

Juan Martín González Vázquez 1983

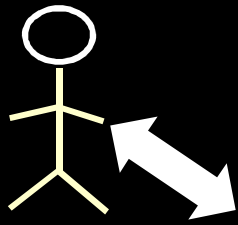
En 1983 Juan Martín González Vázquez desarrolló un sistema evolutivo generador de nominas federales, para lo cual analizó múltiples nominas federales buscando cuales eran los procesos comunes en este tipo de problema y encontró alrededor de 50 procesos que al combinarlos generaban cualquier tipo de nomina, a partir de esto construyo el generador de sistemas y los esqueletos para cada proceso, junto con su tabla de variables y el vector teórico de una nomina federal,

Y dado que lo que buscaba era desarrollar un producto que se pudiera instalar en múltiples organizaciones, y cada organización cambia la forma como se almacenan los datos (base de datos), integró un mecanismo que para cada nueva aplicación, relacionaba la información de la base de datos de la organización con el generador de sistemas,

para lo cual relacionaba la tabla de variables con el diccionario de datos de la base de datos de la organización.

Sistema Evolutivo Generador de Nominas Federales

Juan Martín González Vázquez 1983



**Recaba información
del sistema a desarrollar**

**Procesos
(Esqueletos)
de una
nomina federal**

**Estructura
(vectores teóricos.)
de una
nomina federal**

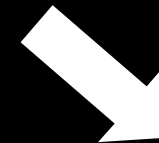
**Diccionario de Datos
de la Base de Datos
de la Organización**

Generador de Sistemas

**integra información del
Diccionario de Datos
con los parámetros**

**Datos del sistema
Parámetros**

**Nomina
Federal**



Generalizando esta idea se tiene un producto donde solo cambiando la Base de Datos y en particular el diccionario de Datos, que es donde esta los atributos de los datos del sistema, como posición, tamaño, tipo, etc., dentro de la Base de Datos, el sistema genera la nueva aplicación sin necesidad de programar.

Esta idea se aplica a muchos sistemas, por lo que, dado que se tiene el sistema evolutivo generador de sistemas, se pueden desarrollar multiples productos, por lo que en lugar de hacer una y otra vez el mismo sistema, se hace el sistema evolutivo generador de sistemas.

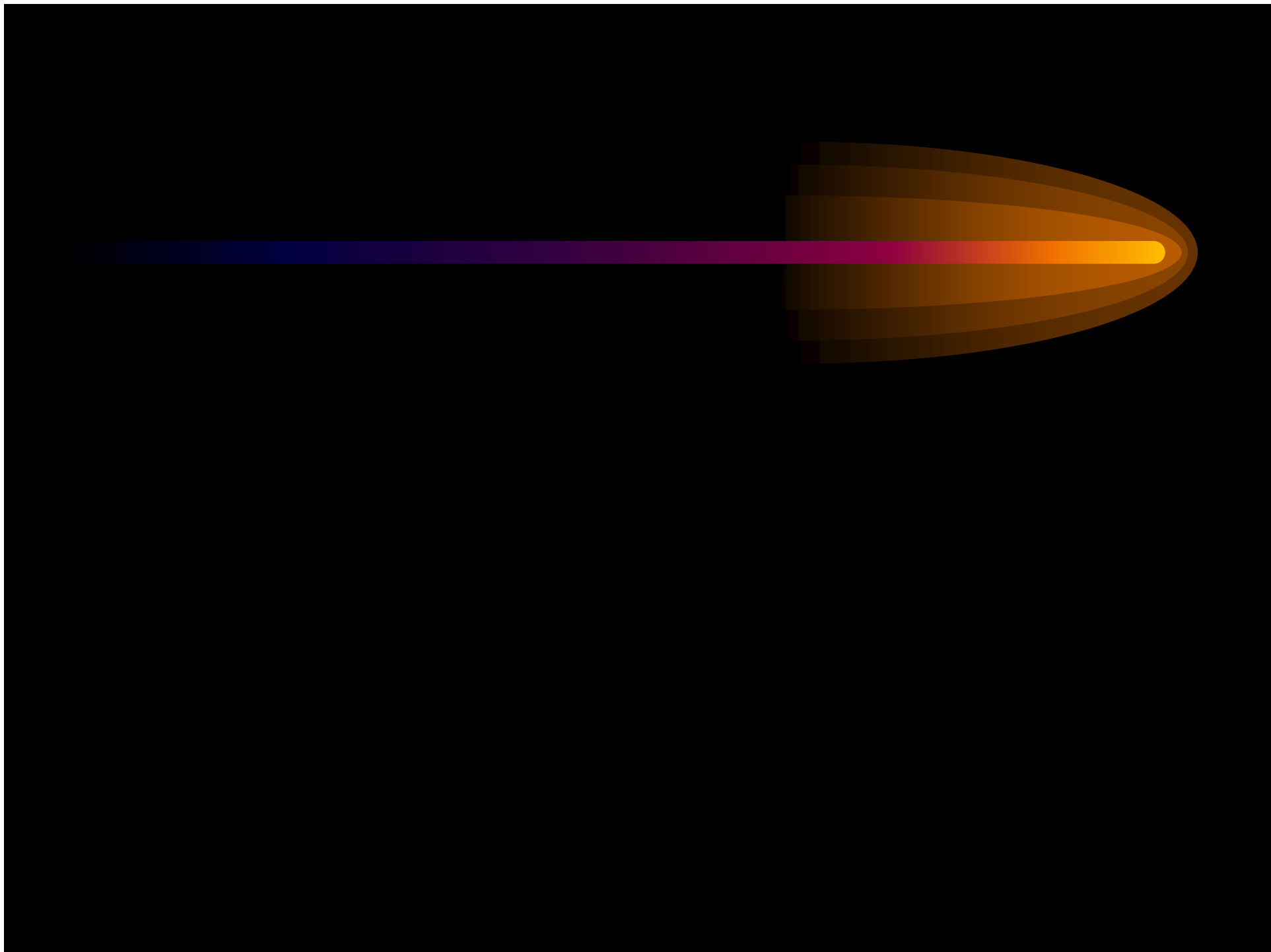
Y en lugar de enseñar a hacer una y otra vez el mismo sistema es necesario que aprendan a hacer y desarrollar plataformas y productos.

Actualmente se tiene una gran oportunidad, con el surgimiento de la Informática Ubicua y multiplataforma surge la gran oportunidad de desarrollar estos productos con un mercado de millones de copias.

Son herramientas con las cuales se pueden desarrollar productos ahora.

La proliferación de plataformas hace que se cree obsolescencia muy rápidamente y requiere herramientas que permitan migrar fácilmente de una plataforma a otra.

Por ejemplo con los generadores de sistemas cambio el esqueleto y tengo la aplicación para la nueva plataforma.





Migradores Automáticos

Christian Zempoaltecatl Ibarra

Principios 80's

Antes del uso masivo de las PC's existían muchos equipos que tenían arquitecturas diferentes, y entre 2 equipos era común que sus sistemas operativos fueran diferentes,

la forma de almacenar los datos era diferentes, por ejemplo además del código ASCII existía el EBCDIC (Extended Binary Coded Decimal Interchange Code), y muchos otros codigos, existían códigos que manejaban 9 o 10 bits en lugar de 8 o como el Radix50 que requería alrededor de 5.64 bits, el tamaño de la palabra no era de 32 o 64 bits como en la PC, había computadoras con palabras de 8, 9, 12 y hasta 72 bits, y existían otras muchas diferencias entre los diferentes equipos de computo,

o sea que no existía prácticamente compatibilidad entre los diferentes equipos.

El problema surgía cuando se adquiría un nuevo equipo de computo y se quería migrar todo lo desarrollado de una plataforma a otra.

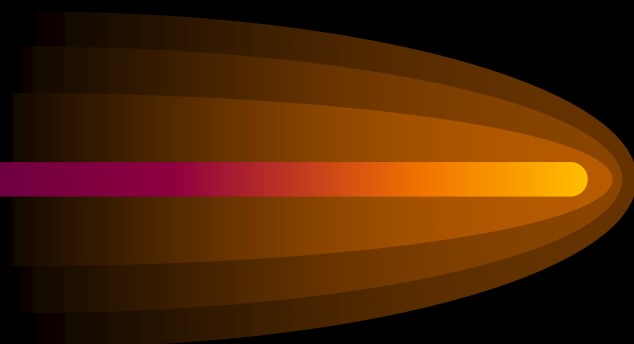
A principio de los 80's Christian Zempoaltecatl Ibarra, que precisamente tenía el problema de migrar de una plataforma a otra, porque su dependencia adquirió un nuevo equipo, desarrollo una herramienta para migrar lo desarrollado de una plataforma a otra.

Este migrador automático tomaba como datos de entrada el código escrito para la plataforma original y generaba el código para la nueva plataforma, y migraba rápidamente miles de líneas de código prácticamente sin errores.

**código escrito para la
plataforma original**

**migrador
automático**

**código para la nueva
plataforma**



Migradores, Convertidores de archivos, Recodificadores, Simuladores, Recuperadores de información, etc., Automáticos

Con la explosión de nuevos dispositivos como teléfonos inteligentes, tables, y dispositivos integrados a las cosas (carros, televisiones, refrigeradores, etc,) y en general de la informática ubicua y con la presencia cada vez mayor de archivos, programas y aplicaciones que ya no son compatibles con las tecnologías que están surgiendo, es cada vez mas necesario desarrollar herramientas que permitan migrar entre múltiples plataformas y sistemas operativos, es ahí donde surge cada vez mas la necesidad de contar con productos que permitan migrar en forma automática.

Cada vez existe mas información que se esta quedando obsoleta y que requiere migrar de plataformas antiguas a las nuevas arquitecturas,
por ejemplo archivos escritos en algún procesador de textos que ya no es comercial o código escrito para algún lenguaje o versión de lenguaje que ya quedo obsoleto

Prácticamente no existen en el mundo herramientas que permitan recuperar toda la información que se esta perdiendo y que esta impactando fuertemente en las organizaciones,
lo cual crea un nicho de mercado impresionante para nuevos productos que en general son fáciles de desarrollar,

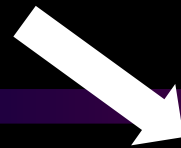
Por ejemplo, en varios de los caso de los archivos de texto es fácil construir los migradores, ya que lo único que se requiere es encontrar el formato de cada carácter (o sea la combinación de números que representan a cada carácter) en el archivo original y su equivalente en el nuevo formato y construir una tabla de conversión de datos entre los dos formatos y un programa que para cada carácter en el formato original lo reescriba en el nuevo formato

Entonces, si en el formato original la letra a toma el valor 30 y en el nuevo formato toma el valor 48, el migrador cada vez que encuentra en el archivo original el numero 30 lo reescribe en el nuevo archivo como 48

Tabla de conversión de datos

Carácter	Formato original	Nuevo formato
a	30	48
b	31	49
c	32	50
d	33	51
..

Archivo en el formato original



Migrador

Mientras existan caracteres en el Archivo

Lee carácter (número en el Formato original)

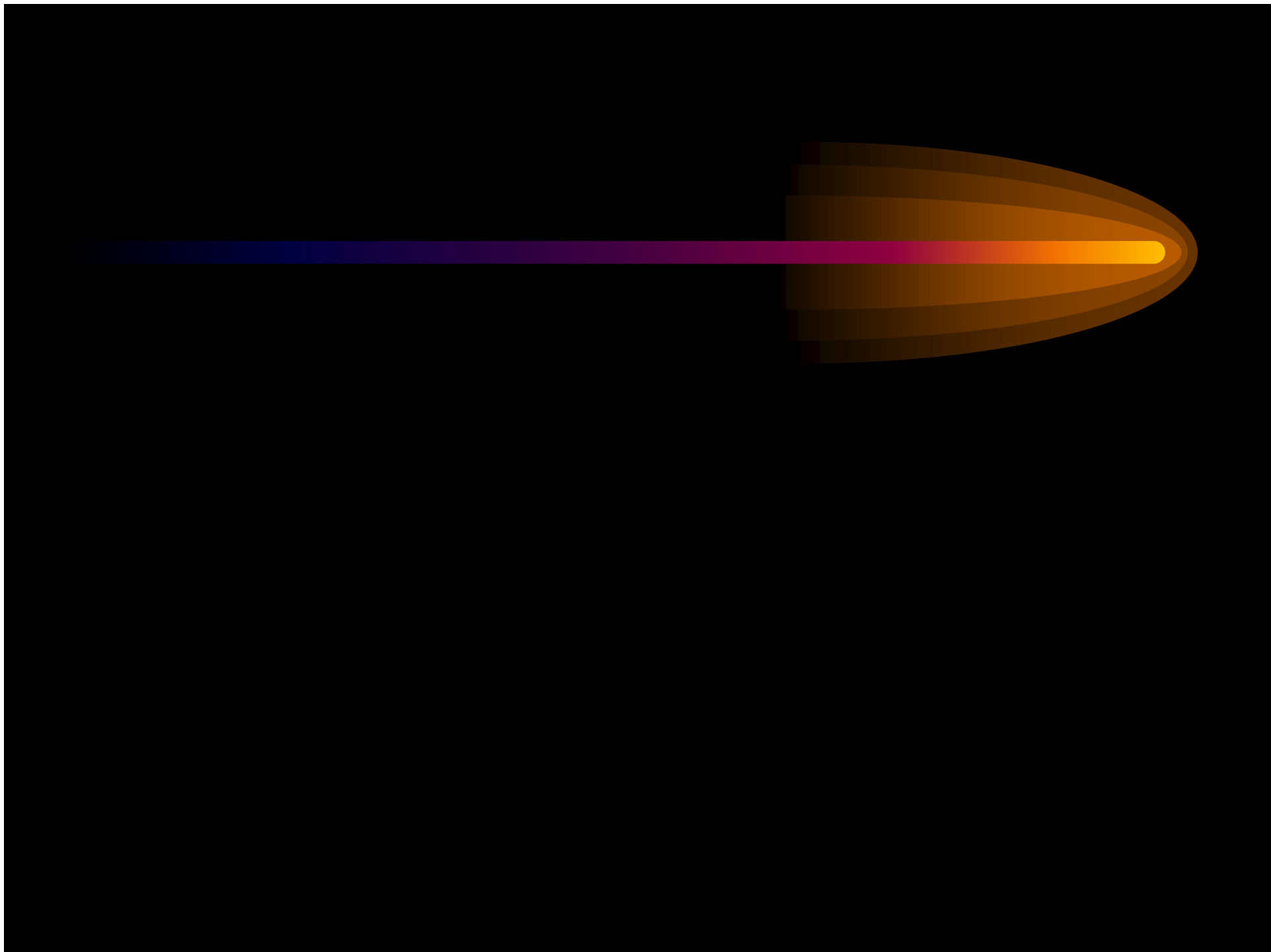
Busca el carácter en la columna de Formato original

Toma el número de la columna del Nuevo formato

Escribe el nuevo numero en el Archivo de nuevo formato



Archivo en el nuevo formato



Documentadores Automáticos

Desarrollados por

Christian Zempoaltecatl Ibarra

Y por

Vicente López Trueba, Julio Cesar León Carrillo, Oscar Olvera Posadas, Carlos Ortega Hurtado

Básicamente un documentador automático es un programa al que se le proporciona la información de un sistema, o de la base de datos o el código de algún programa o sistema y genera la documentación del sistema.

A system for automatic Cobol program documentation

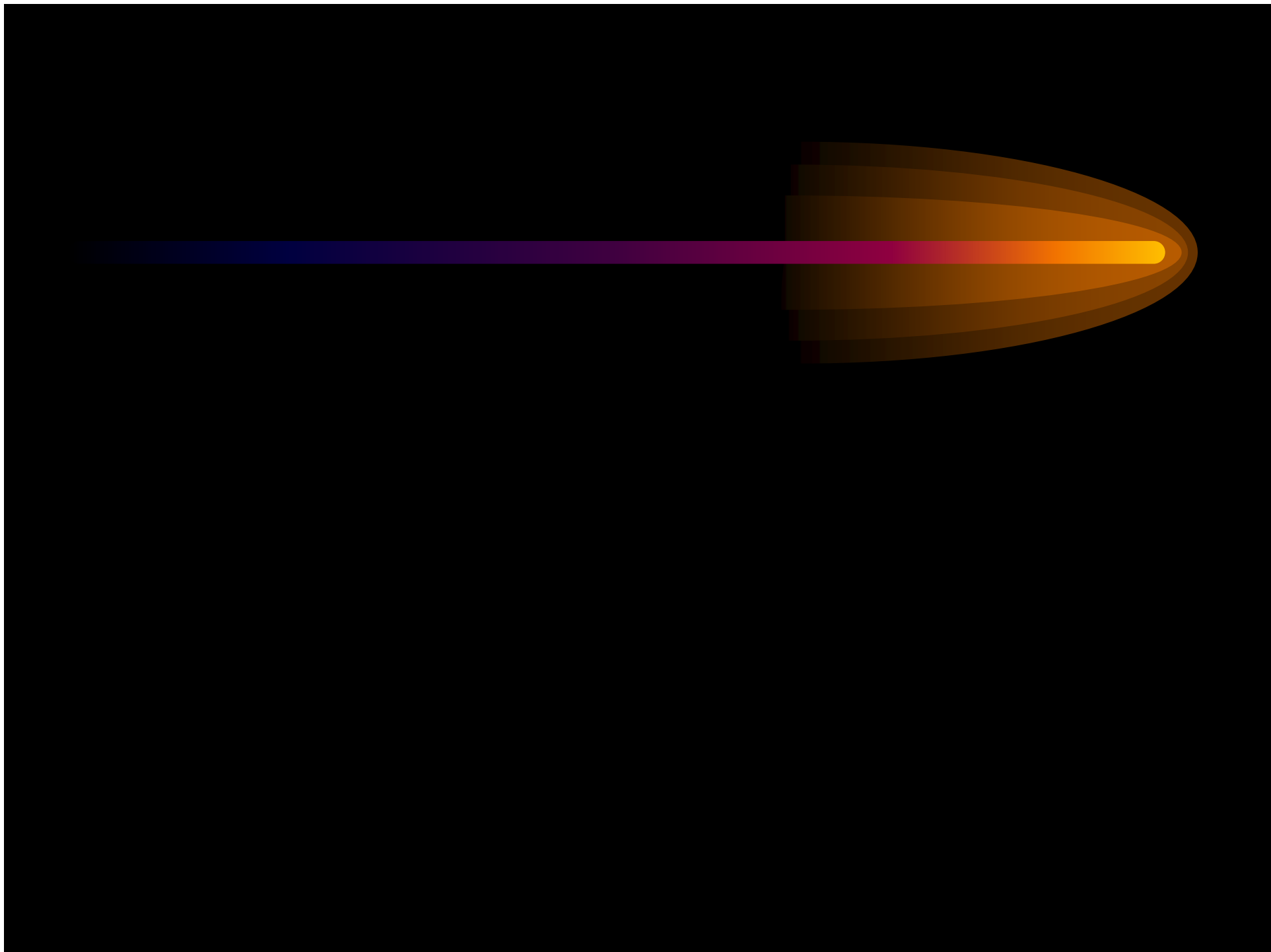
Vicente López Trueba, Julio Cesar León Carrillo, Oscar Olvera Posadas, Carlos Ortega Hurtado

Proceedings of the 3rd annual international conference on Systems documentation

ACM Special Interest Group for Design of Communication

Pages: 36 - 43, Mexico City, México, 1984

<http://dl.acm.org/citation.cfm?id=800551>



Generador de Programas

Generador de Sistemas

Vectores Teóricos

Gramáticas

enfoque lingüístico

Programación Dirigida por Lenguaje

Independencia Relativa

Sistemas Evolutivos

<http://www.fgalindosoria.com/eac/evolucion/>

http://www.fgalindosoria.com/eac/evolucion/evolucion_sev/evolucion_sev.pdf

http://www.fgalindosoria.com/eac/evolucion/evolucion_sev/evolucion_sev_a.pdf

Vicente López Trueba

Christian Zempoaltecatl Ibarra



Ricardo García

Juan Martín González Vázquez



INDEPENDENCIA RELATIVA

En general se considera que un Sistema de Información o programa de cómputo tiene la siguiente arquitectura



Arquitectura de los sistemas de información en los 50's

Para finales de los 70's se manejaba un modelo generalizado en el cual se considera que cualquier sistema de información tiene la siguiente arquitectura



Arquitectura de los sistemas de información en los 70s
Vicente López Trueba

los Generadores de Sistemas, desarrollados con las técnicas mostradas anteriormente, manejan separados explícitamente los datos, procesos y estructura del sistema

Esqueletos



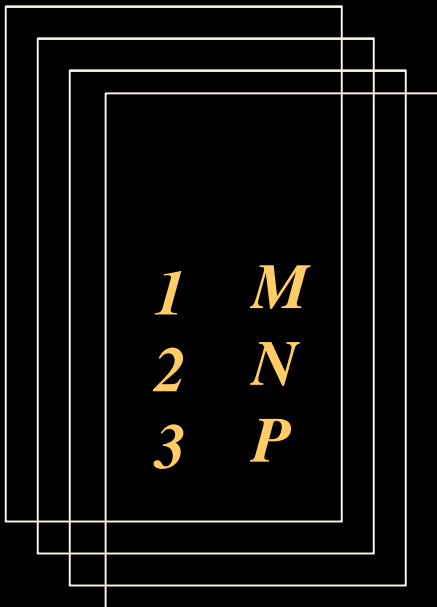
Vector Teórico

Orden en que se ejecutan los procesos



Generador de Sistemas

Parámetros



Generador de Programas

Sistema generado

A
B
C
D

Un sistema tiene datos, procesos y estructura

Vicente López Trueba Finales 70's

Datos ↔ *Parámetros*

Procesos ↔ *Esqueletos*

Estructura ↔ *Vectores Teóricos*
Gramáticas

Desde los años 70's en los Generadores de Sistemas se manejan separados explícitamente los datos, procesos y estructura del sistema

Esto permite que sea relativamente fácil modificar los sistemas,

por ejemplo si se quiere general el sistema en un lenguaje diferente al original, lo único que se requiere es cambiar el esqueleto y reescribirlo en el nuevo lenguaje,

si cambia la posición o características de los dato, solo se requiere modificar la tabla de parámetros o diccionario de datos

si lo que se requiere es modificar la estructura o el orden en que se ejecutan los procesos, solo se modifica el vector teórico o la gramática

una característica fundamental que se debe buscar cuando se desarrolla un sistema de información es la de que exista una

Independencia Relativa

donde los datos, procesos y estructura del sistema queden separados y únicamente existe la relación mínima necesaria entre las tres componentes.

El desconocimiento o el hecho de no tomar en cuenta está cuando se desarrollan sistemas, es una de las causas principales por las que los sistemas se vuelven altamente estáticos y difíciles de mantener

en los sistemas y programas tradicionales las tres componentes se encuentran revueltas, por lo que, un cambio "pequeño" en los datos o procesos o en el orden de atacar un problema ocasiona que prácticamente se tenga que volver a programar todo

por otro lado si se desarrolla el sistema de tal manera que los datos queden en un lado, los procesos en otro y finalmente la estructura del sistema en otro, el proceso de actualización puede ser relativamente fácil

casos particulares de este enfoque es el del desarrollo de Bases de Datos donde las características de los datos quedan en el diccionario de datos y el de los Compiladores donde la gramática o estructura del sistema se almacena en una tabla independiente.

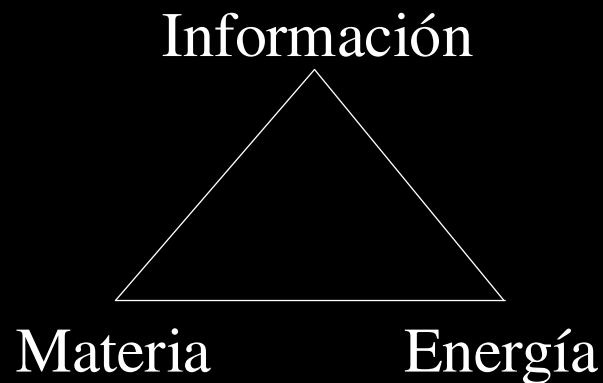
*Independencia relativa entre datos,
procesos y estructura*



Sistemas evolutivos

*El sistema se transforma permanentemente a partir
de los flujos de Materia, Energía e Información en
los que esta inmerso*

De los Generadores de Programas a los Sistemas Evolutivos



REDI

Fernando Galindo Soria

fgalindo@ipn.mx

www.fgalindosoria.com

Abril del 2012, Junio del 2014

MÉXICO

Ligas

Sistemas Evolutivo-Afectivo-Concientes SEAC

Página <http://www.fgalindosoria.com/eac/>

Evolución y Sistemas Evolutivos

Página <http://www.fgalindosoria.com/eac/evolucion/>

De los generadores de Programas a los Sistemas Evolutivos

ppt http://www.fgalindosoria.com/eac/evolucion/gp_a_se/generadores_programas_a_se.ppt

pdf http://www.fgalindosoria.com/eac/evolucion/gp_a_se/generadores_programas_a_se.pdf

Enfoque Lingüístico y Lingüística Matemática

Página <http://www.fgalindosoria.com/lingüisticamatematica/index.htm>

Enfoque Lingüístico

html http://www.fgalindosoria.com/linguisticamatematica/enfoque_linguistico/enfoque_linguistico.htm

pdf http://www.fgalindosoria.com/linguisticamatematica/enfoque_linguistico/enfoque_linguistico.pdf

Word http://www.fgalindosoria.com/linguisticamatematica/enfoque_linguistico/enfoque_linguistico.doc

Programación Dirigida por Sintaxis

html www.fgalindosoria.com/linguisticamatematica/programacion_dirigida_por_lenguaje/programaciondirigidaporsintaxis.htm

pdf www.fgalindosoria.com/linguisticamatematica/programacion_dirigida_por_lenguaje/programaciondirigidaporsintaxis.pdf

Word www.fgalindosoria.com/linguisticamatematica/programacion_dirigida_por_lenguaje/programaciondirigidaporsintaxis.doc