

Consciencia

1

EAC

Evolución

Afectividad

Seminario de Investigación Sistemas Evolutivos, Afectivos y Conscientes

www.fgalindosoria.com/seac/2017/

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

Fernando Galindo Soria

www.fgalindosoria.com

Paola Neri Ortiz

www.eventbrite.com/o/paola-neri-ortiz-12877179662

**Auditorio Sor Juana (edificio C)
Tecnológico de Estudios Superiores de Ecatepec (TESE)
Del Lunes 13 al Jueves 16 de Marzo de 2017
de 12:00 a 17:00 horas de la Ciudad de México**

Se muestra la investigación desarrollada en las áreas de
la Evolución, los Sistemas Evolutivos,
los Sistemas Afectivos
y los Sistemas Conscientes

Finalmente se muestra las bases de construcción de un
seac 0.0, ““sistema consciente””
que surge de la afectividad y siempre está evolucionando.

Un seac es un sistema (organismo, ser, bicho, ente, etc.)
evolutivo-afectivo-consciente

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

Evolution and Evolutionary Systems

Evolución y Sistemas Evolutivos

www.fgalindosoria.com/eac/evolucion/

Evolución y Sistemas Evolutivos

Los Sistemas Evolutivos, son sistemas capaces de transformarse permanentemente a partir de los flujos de materia, energía e información en los que están inmersos

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

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

Afectividad y Sistemas Afectivos

www.fgalindosoria.com/eac/afectividad/

Sistemas Afectivos

Los Sistemas Afectivos (Saf), son sistemas que detectan y muestran sentimientos, incluyendo sistemas que interactúan con su entorno, con las personas y entre ellos mismos.

www.fgalindosoria.com/eac/afectividad/saf/sistemas_afectivos.pdf

www.fgalindosoria.com/eac/afectividad/saf/sistemas_afectivos_a.pdf

Consciencia y Sistemas conscientes

www.fgalindosoria.com/eac/consciencia/

Sistemas Conscientes

Se presenta una propuesta de construcción de “sistemas conscientes”, o sea ““sistemas conscientes de los demás””; ““sistemas conscientes de su entorno””, y ““sistemas conscientes de si mismos””. Y se muestra las bases de construcción de un SEAC 0.0, ““sistema consciente”” que surge de la afectividad y siempre está evolucionando.

www.fgalindosoria.com/eac/consciencia/scon/sistemas_conscientes.pdf

www.fgalindosoria.com/eac/consciencia/scon/sistemas_conscientes_a.pdf

Informático

Y

Lingüista

Matemático

www.fgalindosoria.com/informatica/

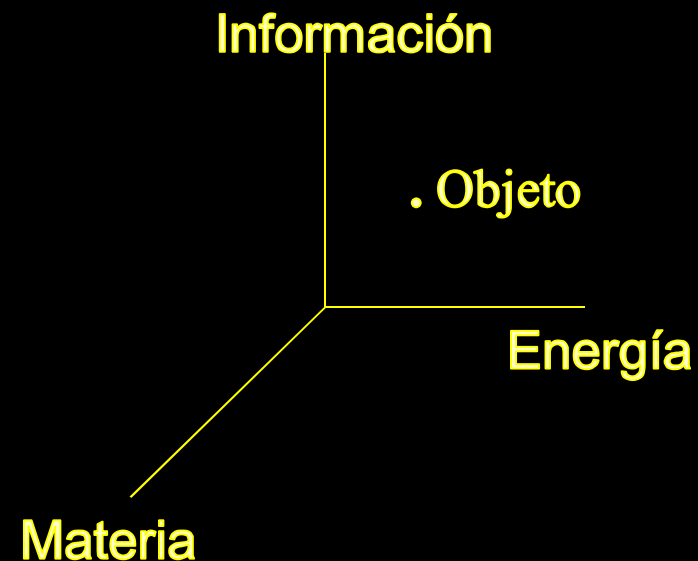
www.fgalindosoria.com/linguisticamatematica/

Enfoque Informático

*La Informática es la ciencia y tecnología de la Información,
Estudia la información, sus fundamentos, sus propiedades y sus
aplicaciones, en los seres vivos, sociedades, computadoras,
organizaciones, universos y en general donde quiera que se presente.*

Información

*in, que significa en, dentro de, y formare que significa dar forma.
significa “forma interna”, “lo que da la forma interna”*



Enfoque Lingüístico

*Cualquier cosa se puede ver
como oración de algún
lenguaje*

www.fgalindosoria.com/linguisticamatematica/

*Hace mucho tiempo, empezó un viaje
mágico y misterioso que todavía sigue y
sigue y no se le ve fin*

Sobre el Concepto de Inteligencia Artificial
Inteligencia Artificial y Vida Artificial
<http://www.fgalindosoria.com/areas/ai/>

Santiago Ramón y Cajal Spanish pathologist, histologist and Nobel laureate. the father of modern neuroscience.

Jacques Lacan, est un psychiatre et psychanalyste français.

Jean Piaget, Swiss, psychologist and philosopher, epistemological studies with children, the great pioneer of the constructivist theory of knowing

Lev Semiónovich Vygotsky psicólogo ruso, psicología del desarrollo, fundador de la psicología histórico-cultural, precursor de la neuropsicología

Alexander Luria neuropsicólogo y médico ruso. uno de los fundadores de la neurociencia cognitiva

Pierre Teilhard de Chardin, French philosopher and Jesuit priest ,Paleontology, Philosophy, Cosmology, Evolutionary theory

Kurt Gödel austrohúngaro lógico, matemático y filósofo

Karl Ludwig von Bertalanffy *Austrian-born biologist, founders of general systems theory*

Warren McCulloch *American neurophysiologist*

Walter Harry Pitts, *logician worked cognitive psychology*

Norbert Wiener *Columbia Missouri, matemático estadounidense*

Arturo Rosenblueth Stearns, *Mexican physician and physiologist*

Oskar Ryszard Lange, *economista, diplomático y político polaco. En 1957 Vicepresidente de Polonia. Cibernética social*

Stafford Beer *Teórico británico, académico, consultor, Cibernética social, primer proyecto de Gobierno Informático*

Pierre de Latil *journaliste scientifique français,*

Andréi Nikoláyevich Kolmogórov, *matemático ruso, probabilidad, topología. fundador de la teoría de la complejidad algorítmica*

William Ross Ashby *médico y neurólogo inglés, Homeostat*

W. Grey Walter *neurophysiologist and robotician. Tortoises*

Claude Shannon *ingeniero electrónico y matemático
estadounidense*

John von Neumann *Hungarian-American mathematician*

Stephen Cole Kleene, *American mathematician.*

Herbert Simon *American political scientist, economist, sociologist,
and psychologist,*

Harold V. McIntosh *Físico, fundamental en el desarrollo de la
Informática*

Pablo Rudomín Zevnovaty *biólogo, fisiólogo, neurocientífico y
académico mexicano.*

Noam Chomsky *lingüista, filósofo y activista estadounidense*

José Negrete-Martínez. México, D.F. Cirujano, Biofísica y Física

Jacob David Bekenstein (Ciudad de México) físico, agujeros negros, entropía y su relación con teoría de la información

Adolfo Guzmán Arenas Ixtaltepec, Oaxaca, ingeniero politécnico mexicano, especialista en cómputo institucional y aplicaciones de alta tecnología.

Humberto Maturana biólogo y epistemólogo chileno. autopoiesis,

Benoît B. Mandelbrot, French American mathematician. Born in Poland, mathematical physics and quantitative finance, as the father of fractal geometry.

Marvin Lee Minsky PhD in mathematics from Princeton

Seymour Papert South Africa mathematician, computer scientist, and educator.

Vicente Lopez Trueba Mexicano, Fundamentos de la Informática

Algunos de los Precusores, Creadores y Fundadores de la Informática

<http://www.fgalindosoria.com/informaticos/pf/>

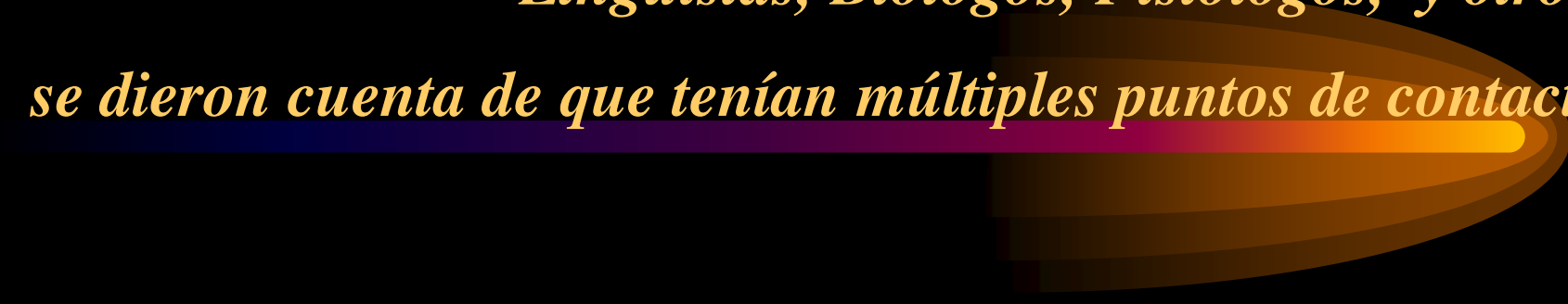
*Psicólogos, Neurofisiólogos, Filósofos, Físicos,
Matemáticos, Lingüistas, Biólogos, Fisiólogos, y otros,
se dieron cuenta de que tenían múltiples puntos de
contacto*

W.Ross Ashby,
Warren
McCulloch, Grey
Walter, and
Norbert Wiener at
a meeting in Paris.

From Pierre de Latil
La Pensée
Artificielle
(Gallimard 1953)



*Psicólogos, Neurofisiólogos, Filósofos, Físicos, Matemáticos,
Lingüistas, Biólogos, Fisiólogos, y otros,
se dieron cuenta de que tenían múltiples puntos de contacto*



*Psicólogos, Neurofisiólogos, Filósofos, Físicos, Matemáticos,
Lingüistas, Biólogos, Fisiólogos, y otros,
se dieron cuenta de que tenían múltiples puntos de contacto
contribuyendo y pescando en una sopa enorme de
conocimiento que se esta transformando a una velocidad
impresionante y en esa sopa se esta pescando y tratando de
encontrar un poco de orden.*

*Psicólogos, Neurofisiólogos, Filósofos, Físicos, Matemáticos,
Lingüistas, Biólogos, Fisiólogos, y otros,
se dieron cuenta de que tenían múltiples puntos de contacto
contribuyendo y pescando en una sopa enorme de
conocimiento que se está transformando a una velocidad
impresionante y en esa sopa se está pescando y tratando de
encontrar un poco de orden.*

*Surgiendo nuevas áreas en tiempo real como la Cibernética,
la Computación, la Inteligencia Artificial, la Informática, la
Robótica, entre otras,*

*Psicólogos, Neurofisiólogos, Filósofos, Físicos, Matemáticos,
Lingüistas, Biólogos, Fisiólogos, y otros,
se dieron cuenta de que tenían múltiples puntos de contacto
contribuyendo y pescando en una sopa enorme de
conocimiento que se esta transformando a una velocidad
impresionante y en esa sopa se esta pescando y tratando de
encontrar un poco de orden.*

*Surgiendo nuevas áreas en tiempo real como la Cibernética,
la Computación, la Inteligencia Artificial, la Informática, la
Robótica, entre otras,
que en la practica usan las mismas herramienta y en muchas
ocasiones entrelazan sus áreas de investigación
dándose el fenómeno de que para mucha gente son lo mismo*

Cibernética, Computación, Informática, Inteligencia Artificial, Robótica, ...



Desde antes de su creación, su creación y actualmente confluyen investigadores de múltiples áreas que se conjugan con los investigadores que han surgido ya dentro del área y que lo mismo tratan de desarrollar los campos de los que surgieron como aquellos fenómenos nuevos o en la frontera

Vida artificial

Neurolingüística

Ubicuidad

<http://www.fgalindosoria.com/icu/>

*Integración de IA, interactividad, robótica,
ubicuidad*

5.- Generación (Tejidos de sistema)

Lingüística Matemática

<http://www.fgalindosoria.com/linguisticamatematica/>

Es un espacio multidimensional en el cual se tratan de entender, desarrollar y aplicar conceptos como conocimiento, pensamiento, inteligencia, evolución, complejidad, afectividad, aprendizaje, conciencia, vida, y muchos otros.

Y se extiende a cualquier área donde se pueda presentar algo relacionado con esos conceptos,.

Y entre otras cosas se busca la construcción y aplicación de modelos del pensamiento, afectividad, aprendizaje, inteligencia, conciencia,...

*La Informatica es una disciplina científico –
tecnológica, por lo que la idea es desarrollar
toda el área contemplando sus diferentes
aspectos*

En el proceso de concretización de una disciplina o área del conocimiento se puede observar que en general esta pasa por varios niveles:

- 1) Temas aislados, se tiene una serie de temas reunidos en forma circunstancial (diccionario).*
- 2) Temas agrupados, los temas se agrupan en temas generales según un criterio de agrupamiento (enciclopedia).*
- 3) Temas jerarquizados o disciplina taxonómica, los temas se integran según un orden y se tiene una primera relación entre ellos (biblioteca especializada).*
- 4) Sistema de conocimiento o disciplina empírica, en este nivel a partir de los temas se abstraen reglas, técnicas, métodos y principios no aceptados en general y la disciplina se describe en término de estas abstracciones y no de los temas específicos.*
- 5) Sistema de conocimiento o disciplina*

REPRESENTACIÓN DEL CONOCIMIENTO

http://www.fgalindosoria.com/eac/evolucion/libro_sistemas_evolutivos/VII1-RCO.pdf

Se busca fundamentar

**desde el nivel funcional
(herramientas, técnicas, métodos, etc.)**

**hasta el nivel conceptual o fundamental
(principios, conceptos, paradigmas, leyes,
teorías, propiedades, teoremas, ecuaciones,...,
etc.).**

*Es importante buscar los conceptos
generales, no solo los casos particulares o
aplicaciones*

*Necesitamos encontrar reglas generales y
aplicarlas a los diferentes casos particulares*

*Por ejemplo el concepto de aprendizaje
Como aprenden las personas, maquinas,
organizaciones, delfines, ...*

Algunos avances dentro de la Fundamentación de las áreas

Teorema de Gödel

Algoritmo, algoritmo recursivo, maquina de Turing,

Shannon

Tesis de Maestría

equivalencia entre álgebra booleana y circuitos

SYMMETRY / SIMETRÍA

<http://www.fgalindosoria.com/informatica/properties/symmetry/>

Symmetry, invariance, theory of groups

Simetría, Invariancia, Teoría de Grupos

www.fgalindosoria.com/informatica/properties/symmetry/symmetry_invariance_groups.htm

Fractal symmetry / Simetria Fractal

[invariancia fractal, autosimilaridad, invarianza bajo cambios de escala,](http://www.fgalindosoria.com/informatica/properties/symmetry/fractal_symmetry.htm)

www.fgalindosoria.com/informatica/properties/symmetry/fractal_symmetry.htm

[Similarity, Power-Law Distributions, Exponential Decay, \$S \rightarrow e^* S^*\$ Una Ecuación de la Naturaleza,](#)

[Zipf's Law / Ley de Zipf,](#)

[Benford's Law / Ley de Benford,](#)

[Sucesión de Farey](#)

[Long Tail / Larga cola,](#)

[Ondas de Elliott,](#)

[Invariancia de Escala y Redes Complejas](#)

Fractal Space-Time / Espacio Tiempo Fractal

www.fgalindosoria.com/informatica/properties/symmetry/fractal_space_time.htm

Gauge Invariance / Invariancia Gauge

www.fgalindosoria.com/informatica/properties/symmetry/gauge_invariance.htm

Symmetric music / Simetría y Música

www.fgalindosoria.com/informatica/properties/symmetry/symmetric_music.htm

**Symmetry, invariance, Conservation Law /
simetría, invariancia, leyes de conservación**

Teorema de Noether

“uno de los teoremas más bellos y profundos que ha dado la ciencia”

<http://bloxito.blogalia.com/historias/35856>

**Campo frontera que se da en la intercepción entre dimensión,
simetría, invariancia, leyes de conservación, teoría de Grupos, etc.**

**el Teorema de Noether se puede establecer como:
A cada simetría (continua), le corresponde una ley de conservación y
viceversa**

[http://es.wikipedia.org/wiki/Teorema de Noether](http://es.wikipedia.org/wiki/Teorema_de_Noether)

www.fgalindosoria.com/informatica/fundamentals/noether/



El viaje sigue y sigue y no se le ve fin....

Cognitive science

Figure illustrating the fields that contributed to

the birth of cognitive science, including

linguistics, education,
neuroscience, artificial
Intelligence, philosophy,
anthropology, and psychology.

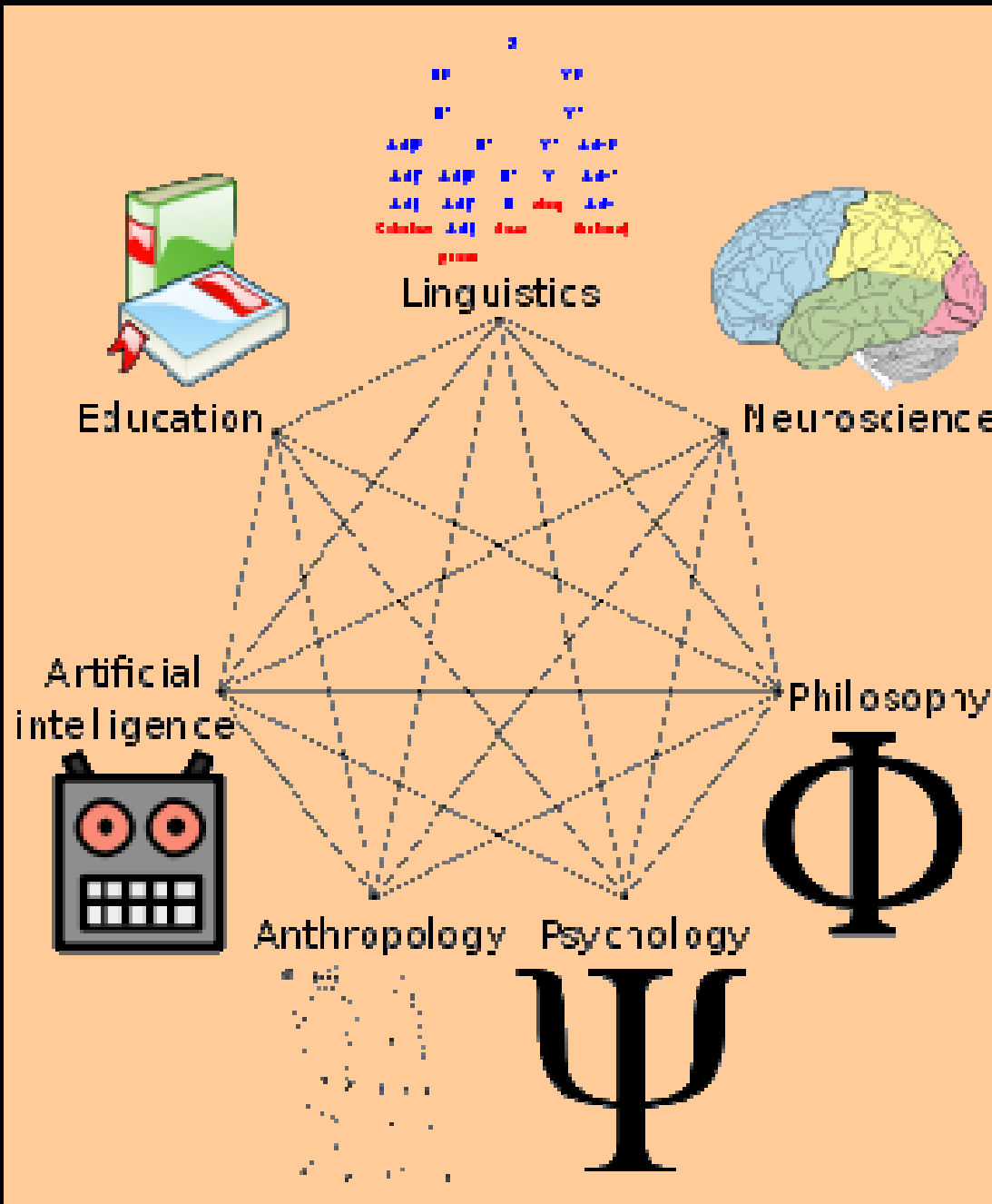
Adapted from Miller, George A (2003). "The cognitive revolution: a historical perspective". *TRENDS in Cognitive Sciences* 7.

http://en.wikipedia.org/wiki/Cognitive_scientist

Se denomina ciencia
cognitiva al estudio
interdisciplinario de cómo la
información es representada y
transformada en la mente /
cerebro. ...

surgiendo en un primer
momento a partir de
disciplinas autónomas como
la lingüística, la inteligencia
artificial, la neurociencia,...

http://es.wikipedia.org/wiki/Ciencia_cognitiva.



Complex Systems

Complejidad de Sistemas o Sistemas Complejos

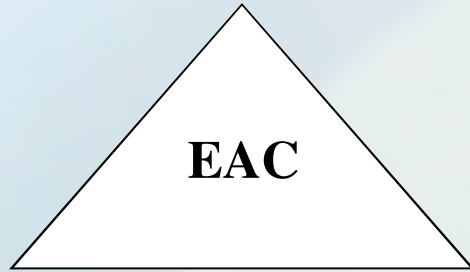


*CCSIPN is a Mexican institution researching on Complex Systems
Instituto Politécnico Nacional
México, D.F.*

<http://comunidad.escom.ipn.mx/sistemascomplejos/Welcome.html>

Es un espacio multidimensional
en el cual se tratan de **entender,**
desarrollar y aplicar conceptos como
conocimiento, pensamiento,
inteligencia, aprendizaje, vida, complejidad,
evolución,
afectividad,
conciencia,
y muchos otros.

Consciencia



Evolución

Afectividad

www.fgalindosoria.com/eac/

Seminario de Investigación Sistemas Evolutivos, Afectivos y Conscientes

www.fgalindosoria.com/seac/2017/

Evolución y Sistemas Evolutivos Sev

www.fgalindosoria.com/eac/evolucion/

La evolución, el crecimiento, el aprendizaje, el pensamiento, la vida, la transformación de nuestra imagen de la realidad, los procesos de descomposición, el desarrollo y transformación de las empresas, sociedades, organizaciones, países, galaxias, universos, etc., son manifestaciones de un mismo proceso general de transformación o cambio al que por facilidad llamamos evolución.

Uno de los problemas más graves de la Informática es su *poca capacidad para modelar en tiempo real los fenómenos que ocurren en la realidad*,

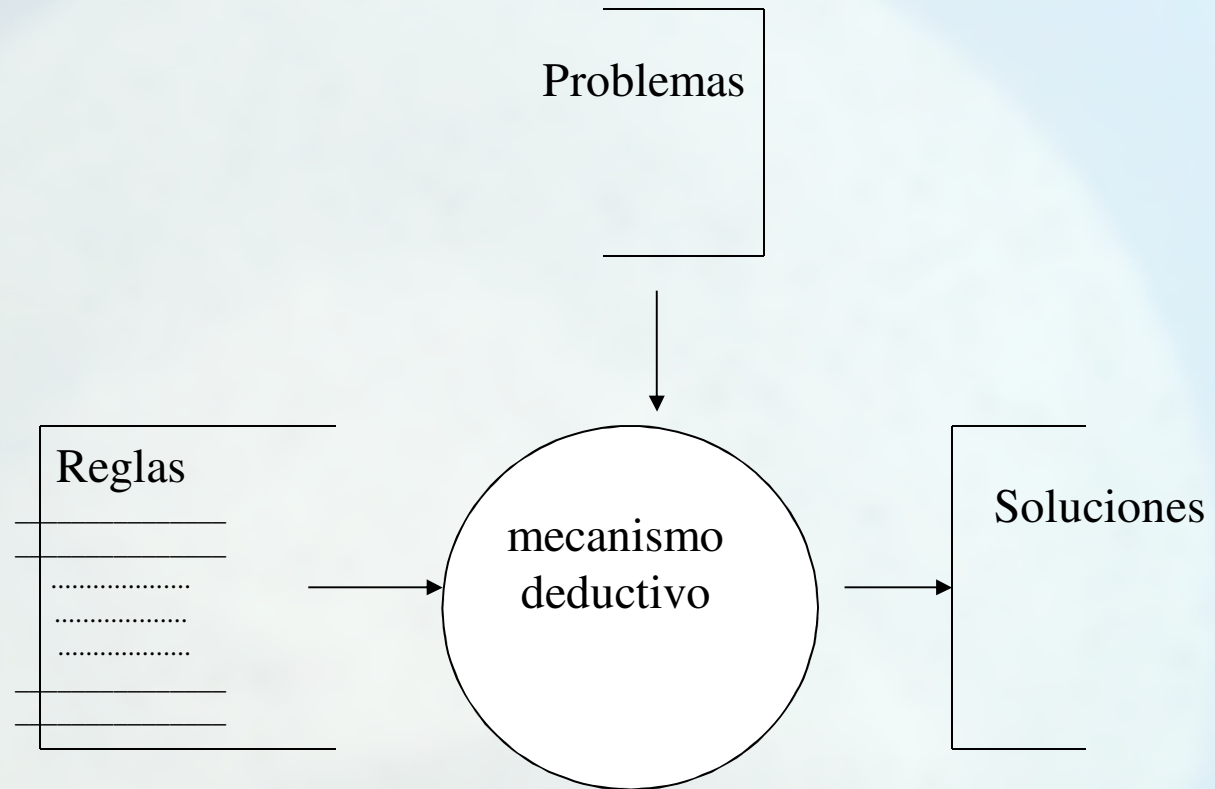
ya que es común que cuando un Sistema de Información: Nómina, Compilador, Reconocedor de Imágenes, Sistema de Inventarios, Sistema Experto de Diagnóstico Médico, etc. se libera, ya prácticamente es obsoleto, *porque:*

- 1) El problema modelado se modificó.*
- 2) Porque el modelo no cubrió los aspectos esenciales.*
- 3) O simplemente la información y el conocimiento que se tiene sobre el tema ha quedado rebasado por algún nuevo dato o hecho no conocido previamente.*

Sistemas Evolutivos

www.fgalindosoria.com/eac/evolucion/sistemas_evolutivos/sistemas_evolutivos.pdf

Modelos
de
desarrollo
de
sistemas
de los años
50 del siglo
XX
**sistemas
formales**



Sistemas Evolutivos

www.fgalindosoria.com/eac/evolucion/sistemas_evolutivos/sistemas_evolutivos.pdf

El problema de la programación deductiva o de sistemas formales es que, se orienta a desarrollar sistemas fijos difíciles de modificar en tiempo real (ya que las modificaciones involucran reprogramar el sistema)

Es por lo anterior que es necesario *replantear el enfoque utilizado para resolver problemas* en Informática (representado por áreas como el Desarrollo de Sistemas, la Ingeniería de Software y la Ingeniería de Conocimiento) en el cual la tendencia es a la construcción de sistemas estáticos e incapaces de automantenerse y buscar métodos y herramientas capaces de recrear en forma continua su imagen de la realidad o del problema a resolver.

Redes Neuronales

Una representación matricial

seminario de IA 1976
modelo matricial de red neuronal 1976

Matrices Evolutivas

http://www.fgalindosoria.com/eac/evolucion/mev/mev/matrices_evolutivas.pdf

En 1976 en el seminario de Inteligencia Artificial organizado en el Centro Nacional de Cálculo (CENAC) del IPN de la Cd. de México en colaboración con la Escuela Superior de Física y Matemáticas (ESFM) del mismo instituto, investigando junto con Gustavo Nuñez Esquer llegamos a una **representación matricial de red neuronal**



McCulloch y Pitts

Redes Neuronales de McCulloch y Pitts

McCullochsabía que cada una de las células nerviosas del cerebro disparan sólo después de alcanzar un umbral mínimo: basta que las células nerviosas vecinas envíen sus señales a través de las sinapsis para que la neurona dispare su propio potencial eléctrico. A McCulloch se le ocurrió que esta organización era binaria, la neurona disparaba o no. Él se dio cuenta de que, la señal de una neurona es una proposición y de que las neuronas funcionarían como puertas lógicas, con múltiples entradas y una sola salida. De modo que, variando el umbral de disparo de una neurona podrían llevarse a cabo las funciones “y”, “o” y “no”.

McCulloch y Pitts escribieron sus hallazgos en un artículo seminal:
“Un cálculo lógico de las ideas inmanentes en la actividad nerviosa”,

A Logical Calculus of the Ideas Immanent in Nervous Activity
Bulletin of Mathematical Biophysics, 1943

<https://pdfs.semanticscholar.org/5272/8a99829792c3272043842455f3a110e841b1.pdf>

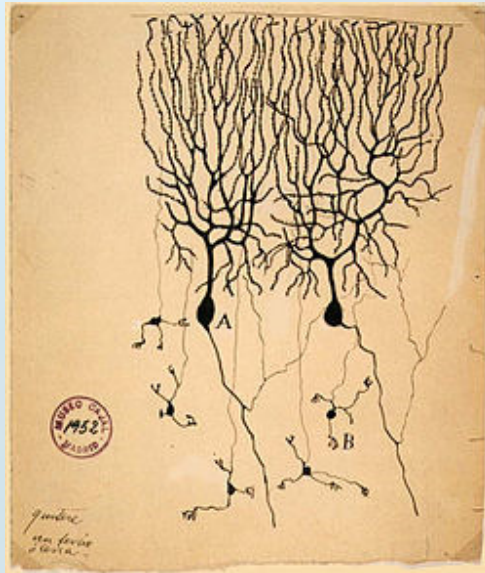
Tomado de un artículo precioso sobre el tema, Altamente recomendable
El hombre que intentó salvar al mundo con la lógica, Por Amanda Geffer

Traducido al español por Marta Lapid Volosin para Nautilus

Calíope: Ciencia Con Cerebro, 5 octubre, 2016 5 octubre, 2016

<http://cienciaconcerebro.com/2016/10/05/el-hombre-que-intento-salvar-al-mundo-con-la-logica/>

Neuronas



Drawing of neurons in the pigeon [cerebellum](#), by Spanish neuroscientist [Santiago Ramón y Cajal](#) in 1899. (A) denotes [Purkinje cells](#) and (B) denotes [granule cells](#), both of which are multipolar. <https://en.wikipedia.org/wiki/Neuron>

Redes Neuronales de McCulloch y Pitts

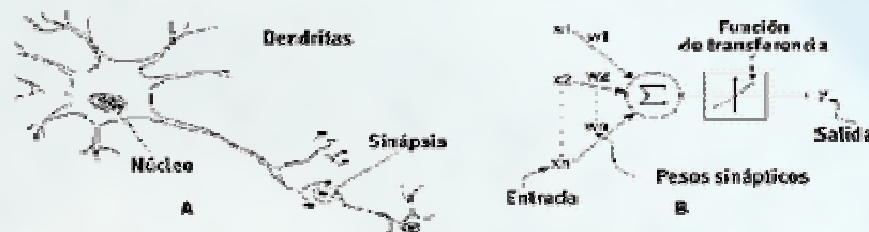
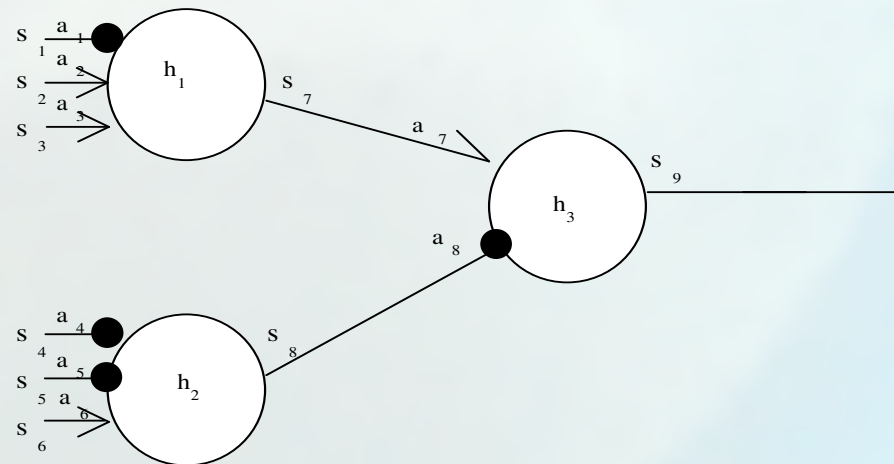


Fig. 3. A) *Neurona biológica*. B) *Neurona artificial*.

Sistemas bio-inspirados y aplicaciones energéticas
Julio R. Gómez Sarduy y Marcos A. de Armas Teyra
<http://www.cubasolar.cu/biblioteca/Energia/Energia42/HTML/Articulo07.htm>



Ejemplo de una red neuronal

a partir del modelo de neurona desarrollado por McCulloch y Pitts generamos nuestra propia propuesta, basada en una representación matricial de la red neuronal, en la cual cada una de las señales de entrada a las neuronas se representa como una columna de la matriz y cada una de las neuronas equivale a un renglón, de tal forma que, en la intersección de cada renglón y columna se almacena el valor que toma la dendrita en caso de que reciba una señal de entrada

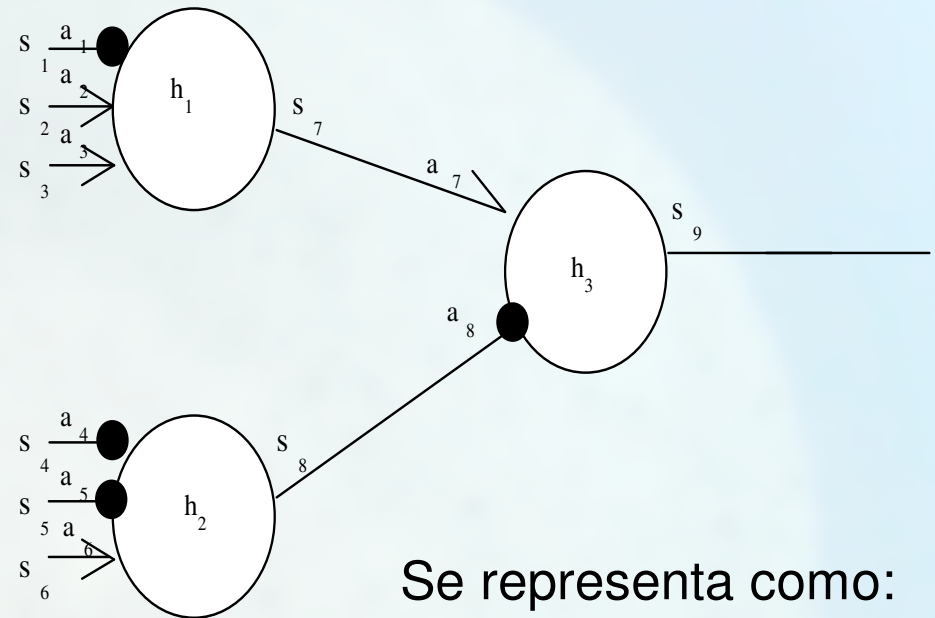
Representación matricial de la Red Neuronal 1976

Fernando Galindo Soria y Gustavo Núñez Esquer

La matriz representa a la red neuronal, el vector a a las señales de entrada / salida a la red (s_i), los renglones representa a las neuronas (n_j) y las columnas corresponden a las señales de entrada a la neurona (a_k).

La matriz se multiplica por el vector de señales de entrada al sistema y si el valor resultante es mayor o igual a cero entonces se considera que la neurona es excitada por lo que se genera un valor de salida de la neurona igual a uno.

La Red Neuronal:



Se representa como:

a1	a2	a3	a4	a5	a6	a7	a8	h			
-1	1	1	0	0	0	0	0	h_1	s_1	=	n_1
0	0	0	-1	-1	1	0	0	h_2	s_2		n_2
0	0	0	0	0	0	1	-1	h_3	s_3		n_3
									s_4		
									s_5		
									s_6		
									s_7		
									s_8		
									-1		

sí $n_1 \geq 0$ entonces	$s_7=1$
sí $n_2 \geq 0$ entonces	$s_8=1$
sí $n_3 \geq 0$ entonces	$s_9=1$

La red neuronal esta formada por un sistema de neuronas interconectadas, donde cada neurona tiene al menos entradas, salida y un valor de umbral y a su vez la red tiene múltiples interrelaciones, ahora bien cuando se construía la red era necesario dar todos estos valores al sistema durante la etapa de construcción (sistema formar) o lograr que la red los adquiriera

Al proceso que permiten que la red neuronal adquiriera los valores que necesita se le conoce como “aprendizaje”

Como logramos que la red neuronal aprenda?

Maquinas que aprenden

Machine learning

https://en.wikipedia.org/wiki/Machine_learning

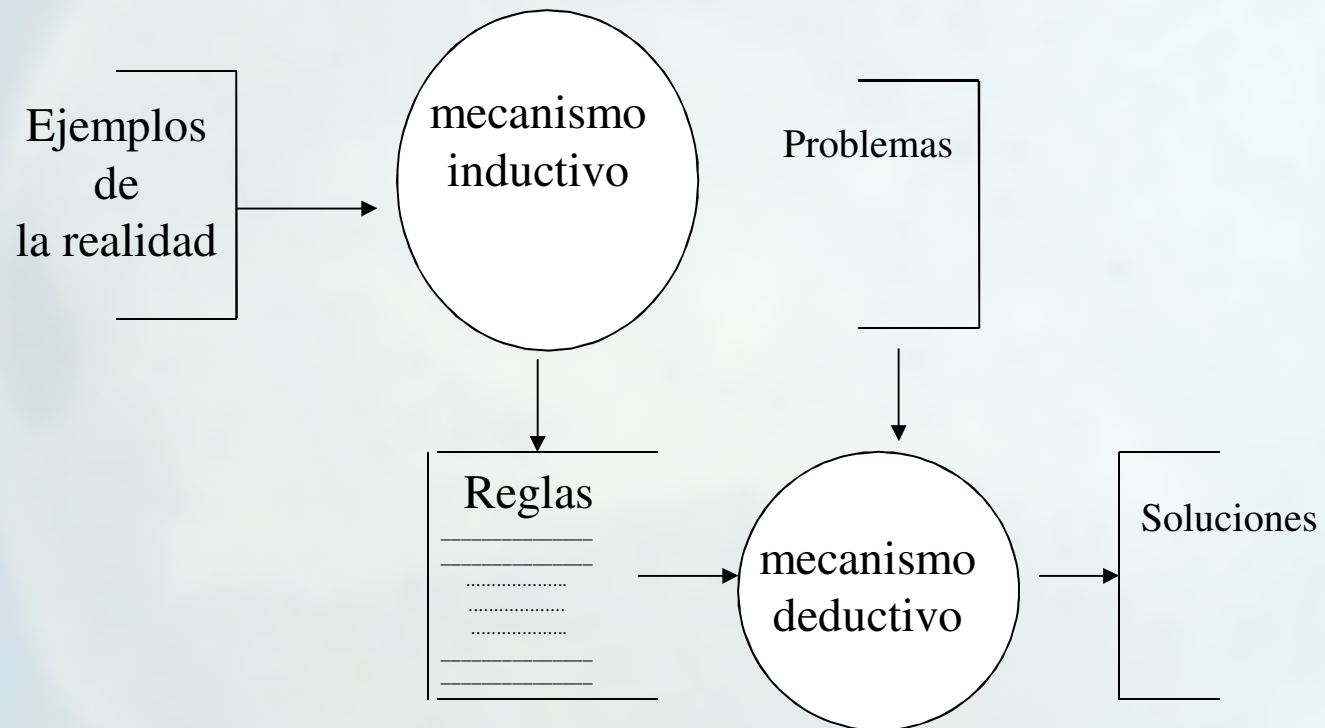
(1974)

desde principios de los 60, se ha buscado desarrollar herramientas capaces de obtener en forma automática el conjunto de reglas del sistema a partir de ejemplos y descripciones generales de un programa, en lo que se conoce como

Machine learning

o maquinas o sistemas que aprenden.

Machine learning (Sistemas que aprenden) años 60



Las Maquinas que Aprenden tiene dos grandes etapas: una “de aprendizaje” y otra Formal.

Por lo que solo resuelve problemas que previamente se le enseñó a resolver, pero cuando llegan problemas que previamente no aprendió no los reconoce,

si por ejemplo se le dan un montón de ejemplos de A's reconoce A's, pero si posteriormente se me ocurre colocar una A de lado y cuadrada, es posible que ya no la reconozca si previamente no se le enseñó a reconocer ese tipo de A's.

Por lo que si se quiere que la reconozcan es necesario enseñarles de nuevo

El problema tanto de los sistemas formales como de los sistemas que aprenden o machine learning es que tarde o temprano les llegara un problema que no pueden resolver.

Teorema de Gödel

En 1930 Kurt Gödel estableció un teorema que lleva su nombre, el teorema de Gödel puede considerarse como un teorema fundamental en informática.

El teorema dice en general que: un sistema Formal, es incompleto, o inconsistente,

Si se logra que un sistema formal sea completo será inconsistente y si es consistente entonces será incompleto.

Dado que no importa el numero de reglas que se den,

si el sistema formal interrelaciona permanentemente con un espacio abierto, tarde o temprano no podrá resolver los problemas

Porque no tiene capacidad de aprendizaje o esa capacidad la tiene independiente y desconectada del sistema que esta resolviendo los problemas

Desconectar el proceso de aprendizaje
de un sistema que aprende es como
desconectar el proceso de aprendizaje
de una persona

inteligencia

Para Jean Piaget, la inteligencia tiene dos atributos principales: la *organización* y la *adaptación*.

El primer atributo, la organización, se refiere a que la inteligencia está formada por estructuras o esquemas de conocimiento, cada una de las cuales conduce a conductas diferentes en situaciones específicas.

.....
La segunda característica de la inteligencia es la adaptación, que consta de dos procesos simultáneos: la asimilación y la acomodación

http://es.wikipedia.org/wiki/Teor%C3%ADa_Constructivista_del_Aprendizaje

asimilacion ingreso de nuevos **elementos** a los esquemas mentales preexistentes,.

acomodación modifica esquemas (**estructuras cognitivas**)

(2012)

Asimilación (psicología)

La **asimilación** (del Lat. *ad* = hacia + *similis* = semejante) es un concepto **psicológico** introducido por **Jean Piaget** para explicar el modo por el cual las personas ingresan nuevos **elementos** a sus esquemas mentales preexistentes, explicando el crecimiento o cambio cuantitativos de éste. en este caso no existe modificación en el esquema sino sólo la adición de nuevos elementos.

[http://es.wikipedia.org/wiki/Asimilaci%C3%B3n_\(psicolog%C3%ADa\)](http://es.wikipedia.org/wiki/Asimilaci%C3%B3n_(psicolog%C3%ADa))

Acomodación

La **acomodación** o **ajuste** es un concepto **psicológico** introducido por **Jean Piaget**. Por acomodación se entiende el proceso mediante el cual el sujeto modifica sus **esquemas (estructuras cognitivas)** para poder incorporar a esa estructura cognoscitiva nuevos objetos. Esto puede lograrse a partir de la creación de un nuevo esquema, o la modificación de un esquema ya existente para que el nuevo estímulo pueda ingresar en él. Por esta razón suele considerarse este mecanismo como un cambio cualitativo en el esquema. A partir de estos procedimientos, que Piaget denomina **funciones cognitivas** se establece el proceso de adaptación y equilibrio cognitivo entre esquema y medio del organismo.

<http://es.wikipedia.org/wiki/Acomodaci%C3%B3n>

la idea sería desarrollar programas que por ejemplo contuviera reglas para reconocer una A, y que cuando no reconociera un tipo especial de A, en forma natural se encargara de aprender sin necesidad de que fuera un proceso explicito e independiente.

de donde el programa no mandaría Fatal error o warnings, simple y sencillamente si el programa no sabe, aprende.

ese es el principio de lo que es un *sistema evolutivo*
o sea un programa que permanentemente esta aprendiendo y reestructurando sus reglas

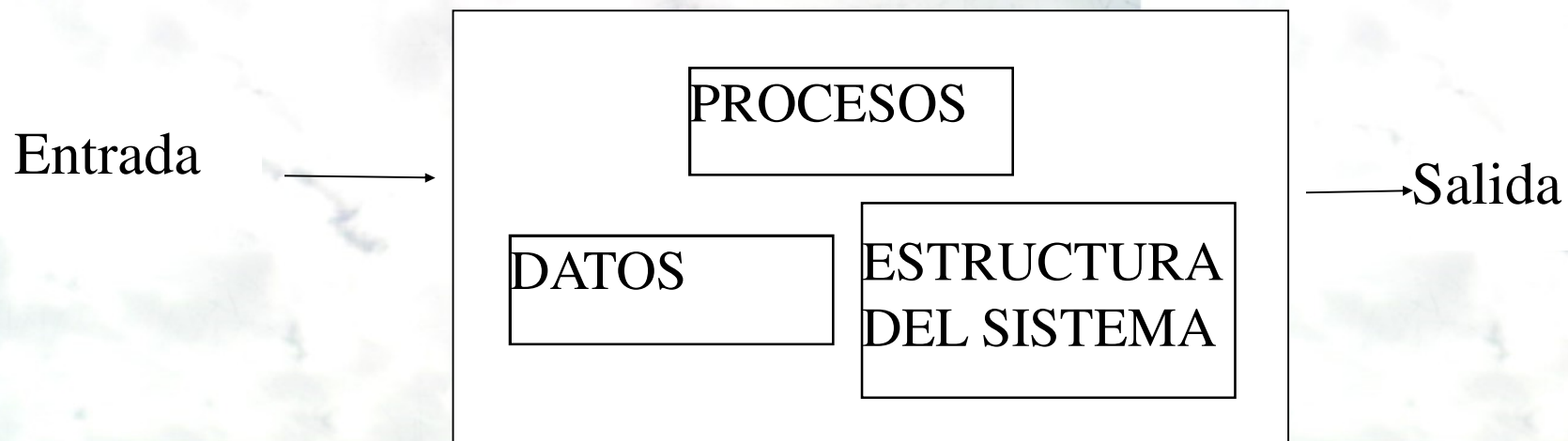
PERO

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



Arquitectura de los sistemas de información en los 50s

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



Arquitectura de los sistemas de información en los 70s

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** (un caso particular de este enfoque es el del desarrollo de Bases de Datos).*

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

Independencia Relativa

entre los datos, procesos y estructura del sistema buscando que queden separados lo mas posible y únicamente existe la relación mínima necesaria entre las tres componentes.

generadores de programas, 75 79

generadores de sistemas

independencia relativa

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

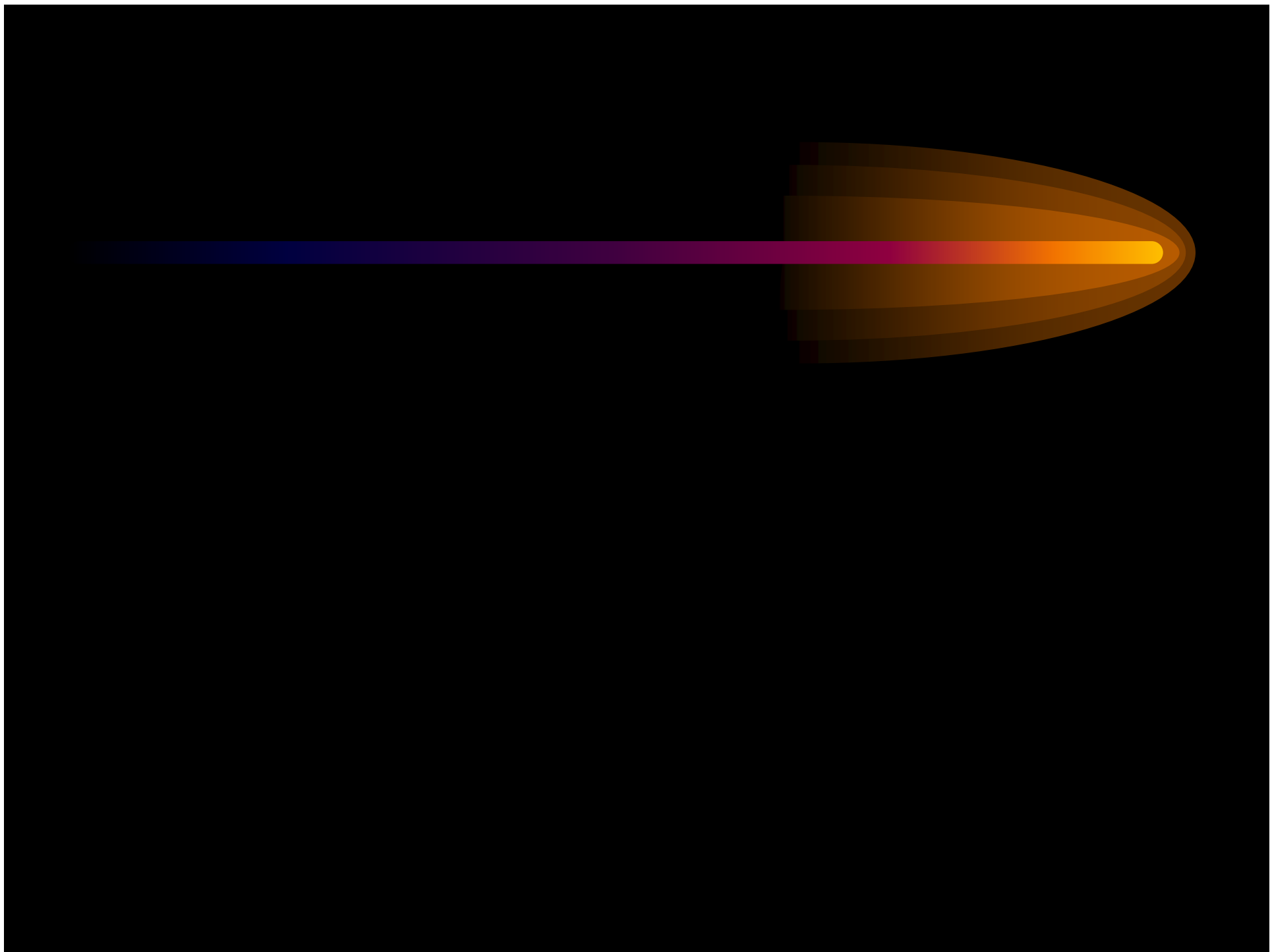
*programa generado
en el lenguaje del
paquete estadístico,*

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***

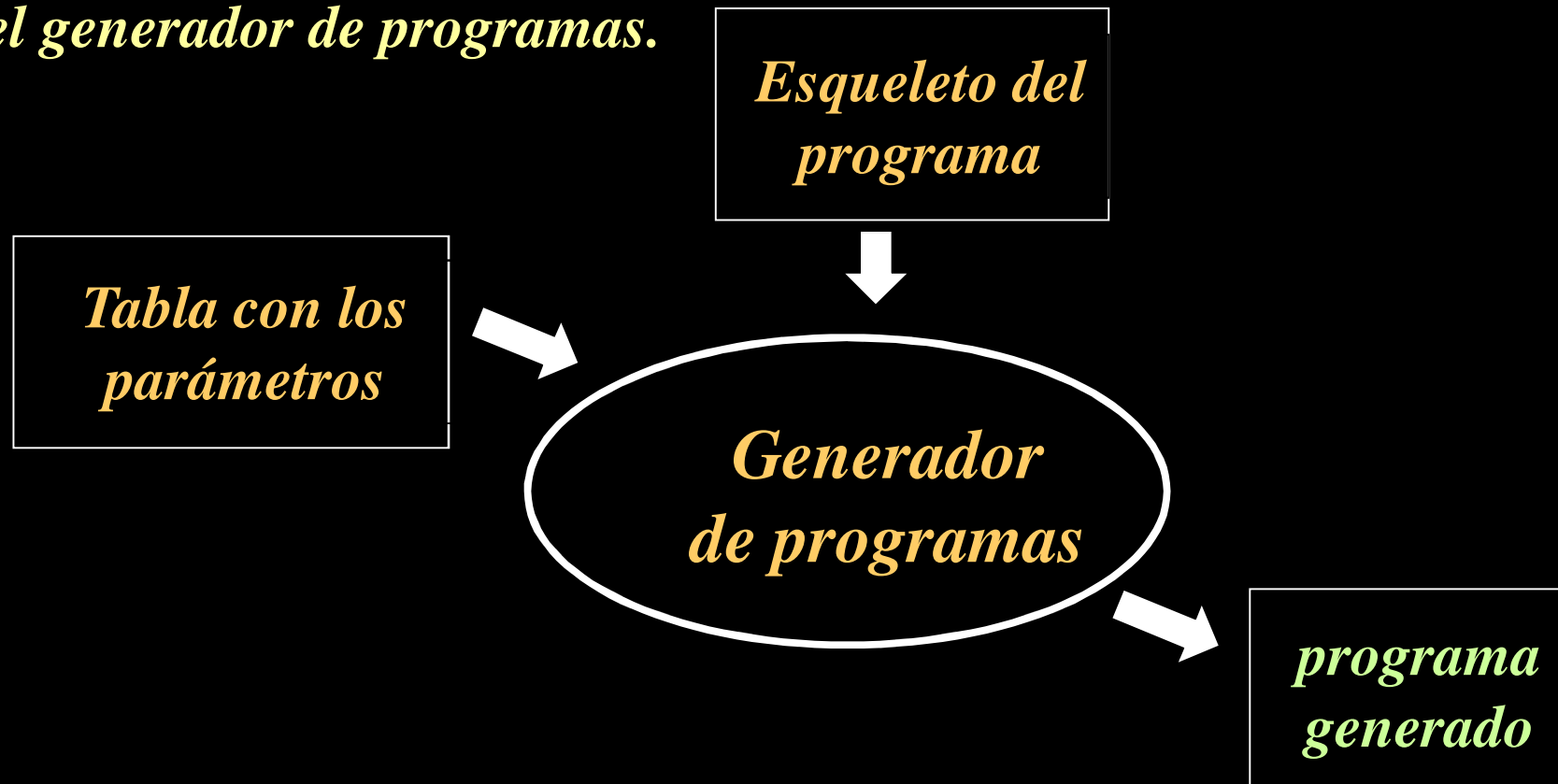
esqueleto

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.



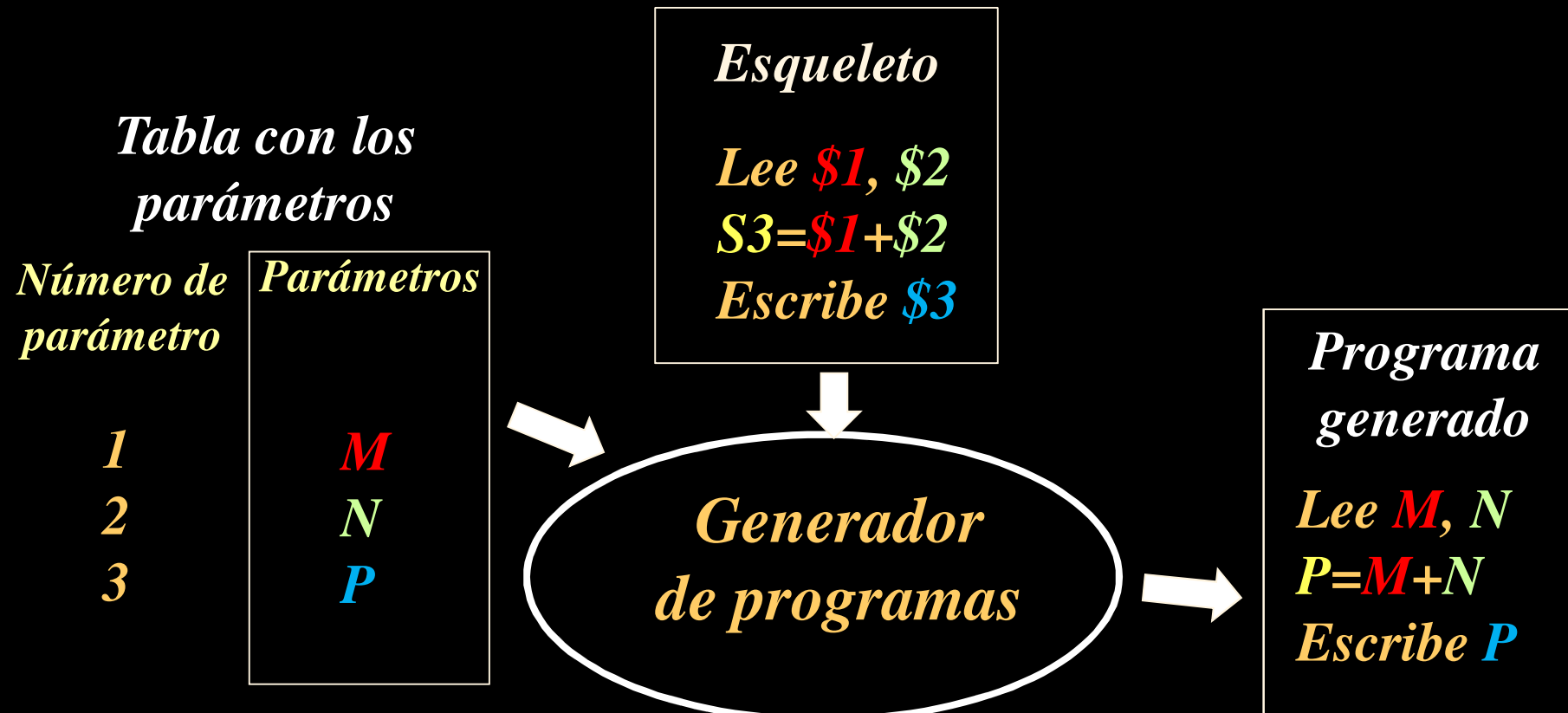
Generador de Programas

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;

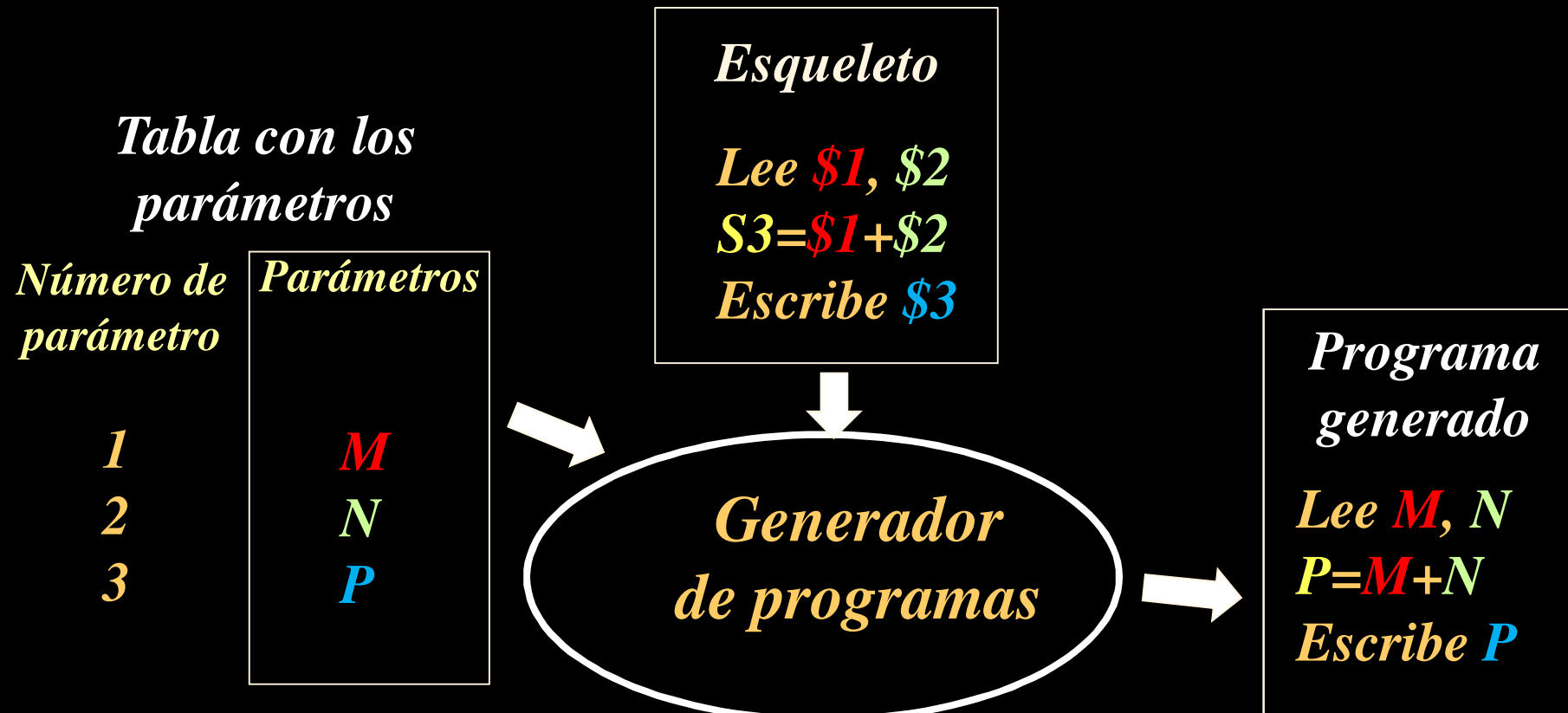
se colocan **M**, **N** y **P** en la tabla de parámetros;

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

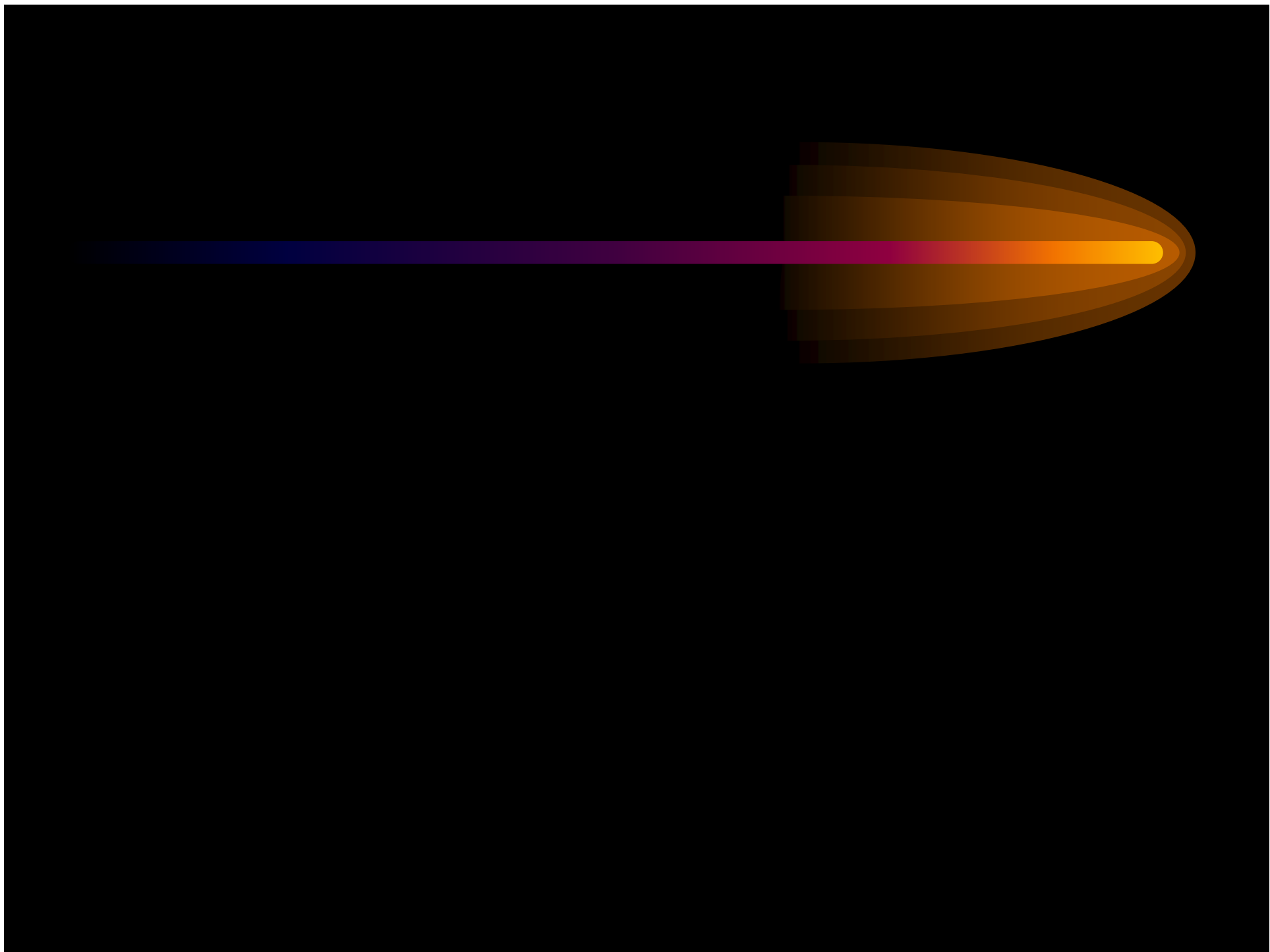
Parámetros

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

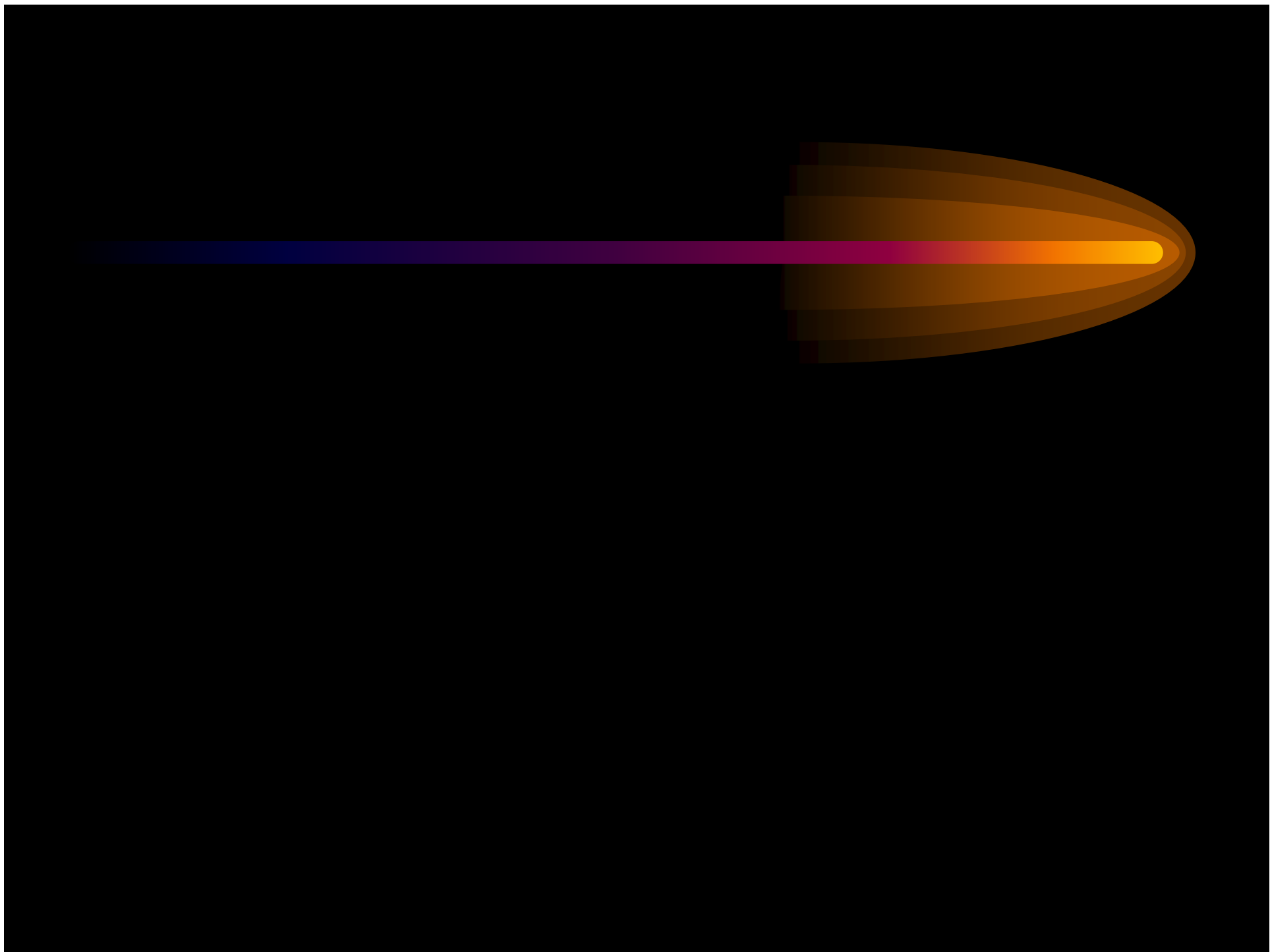
*Sistema
generado*

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

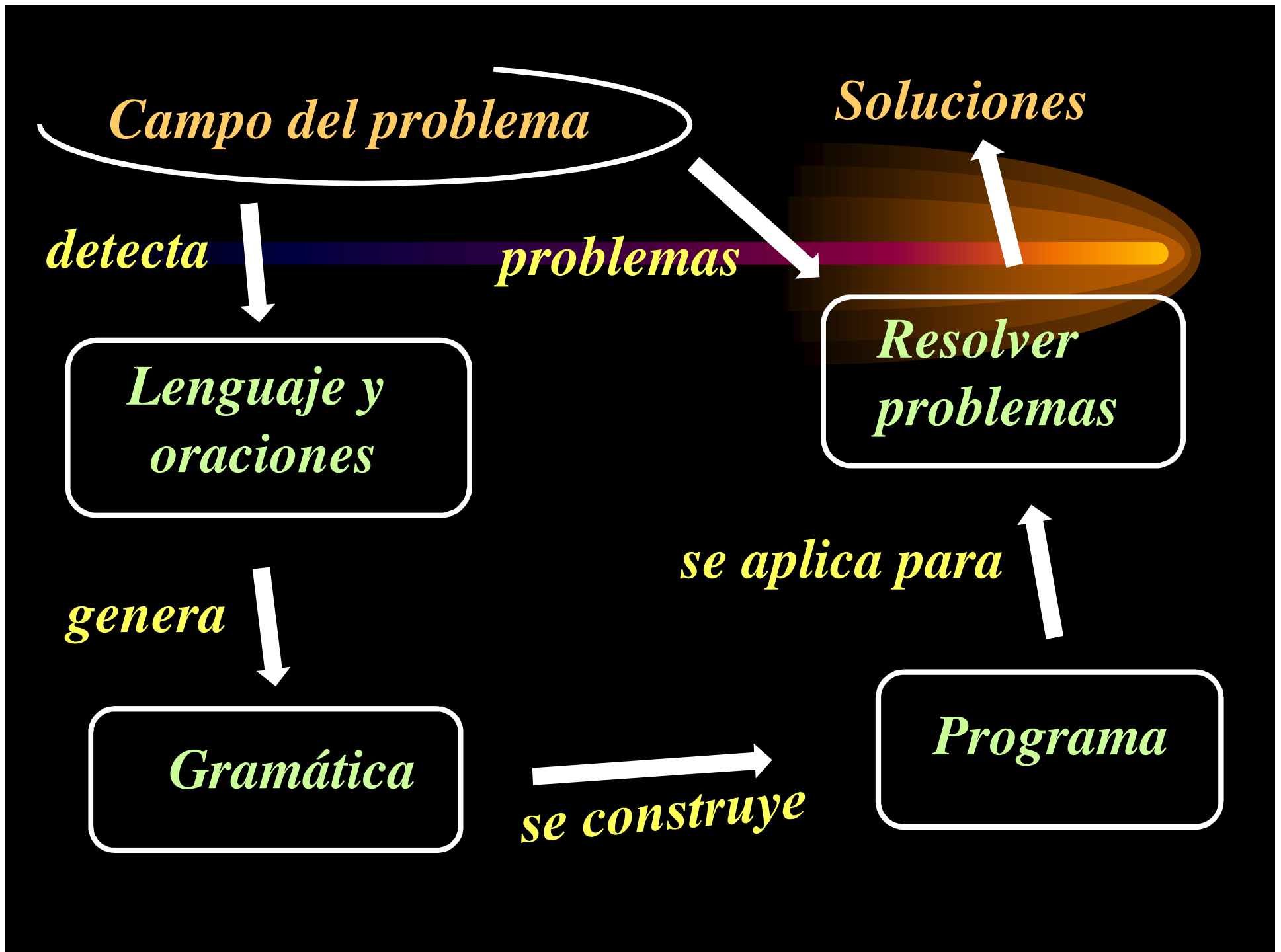
1975 - 1985

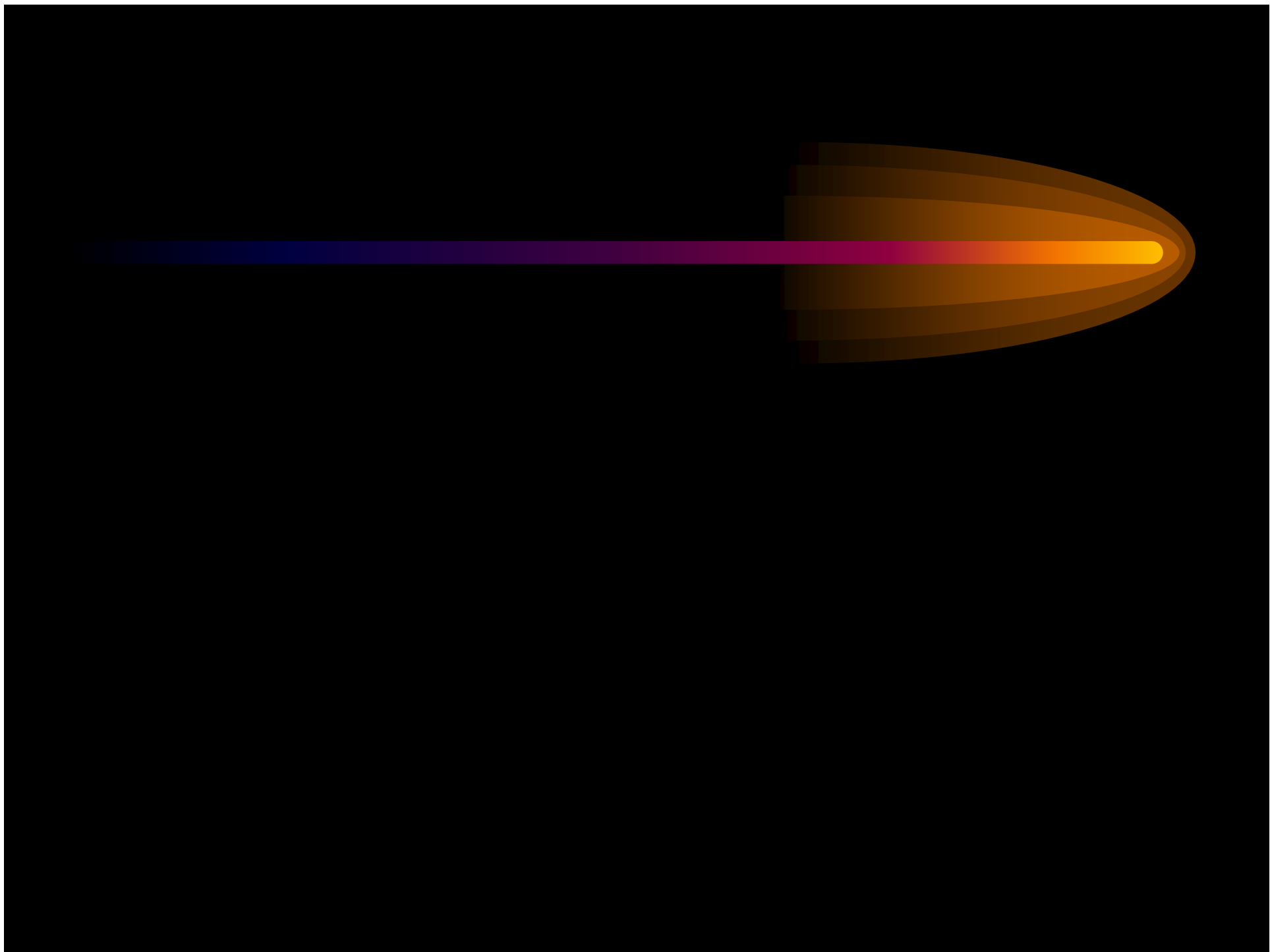
Los problemas informáticos en general son susceptibles de un tratamiento lingüístico, por lo que:

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.

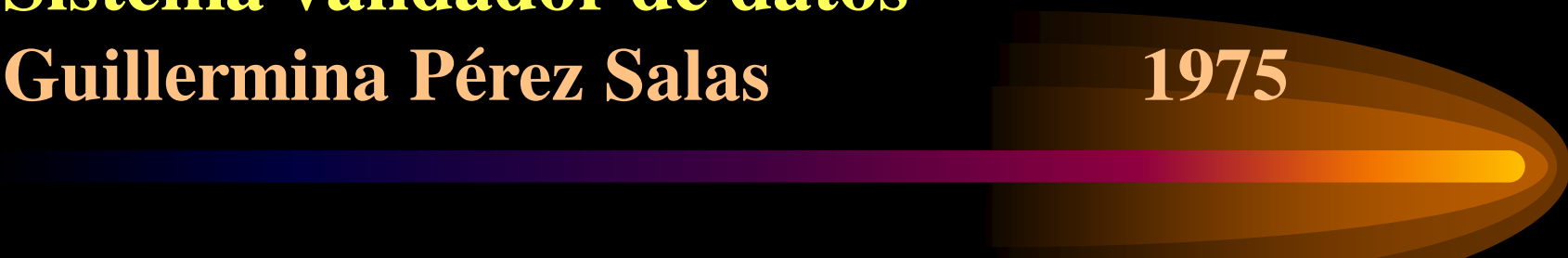
Desde 1975 hemos desarrollado una gran cantidad de sistemas, basados en la representación lingüística y mediante gramáticas de diferentes problemas

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

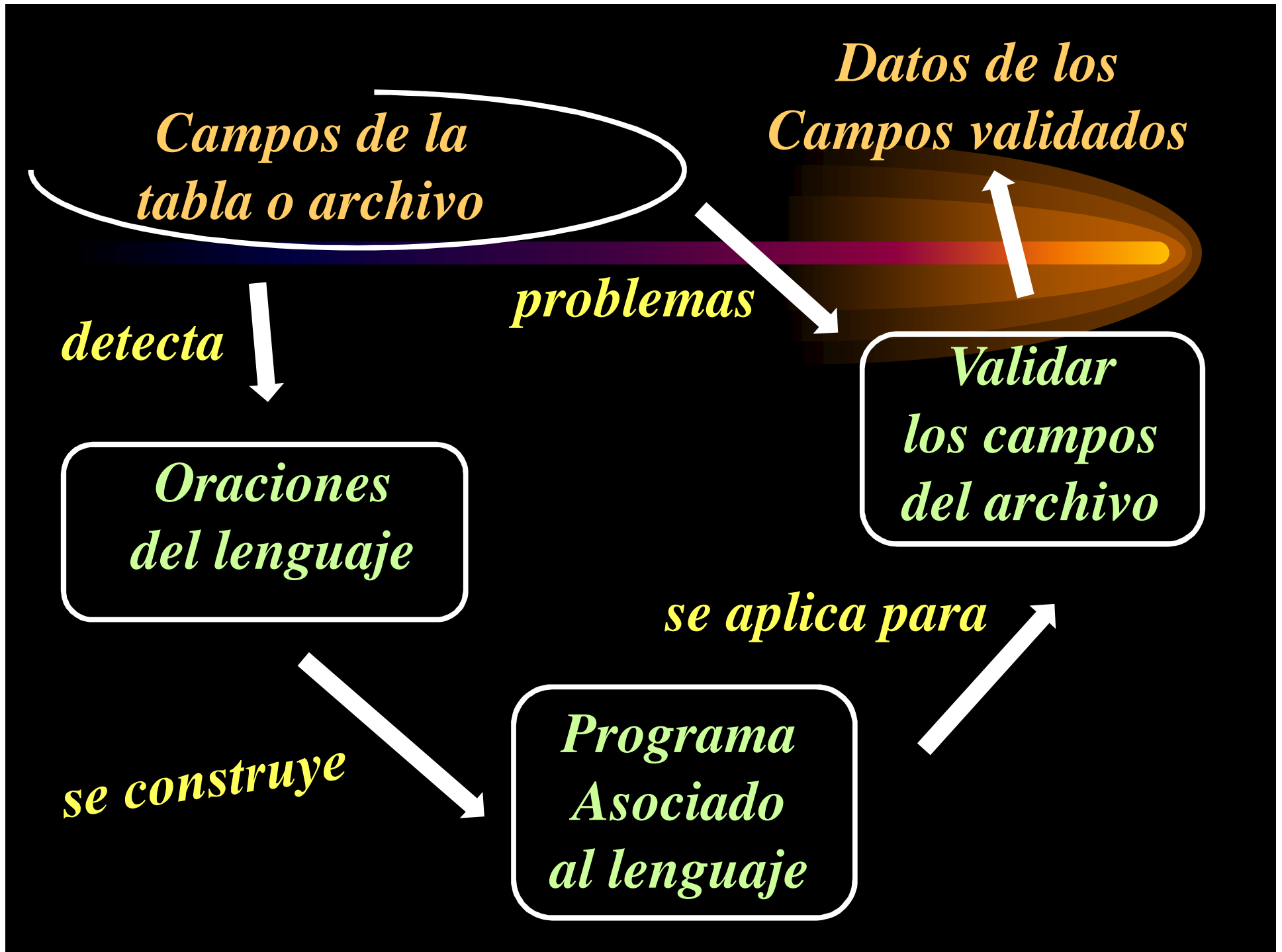
*Oraciones
del lenguaje*

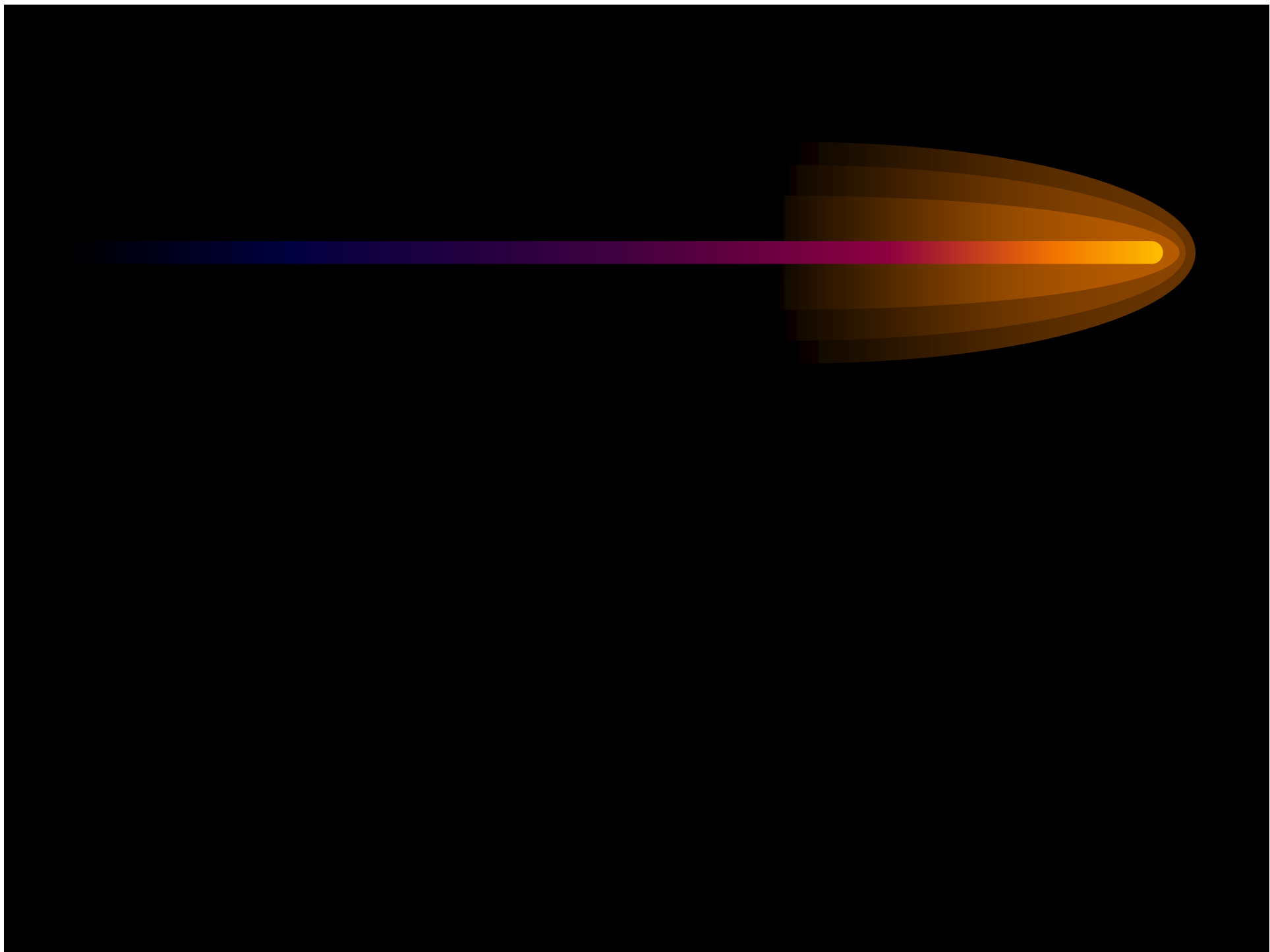
*Validar
los campos
del archivo*

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.

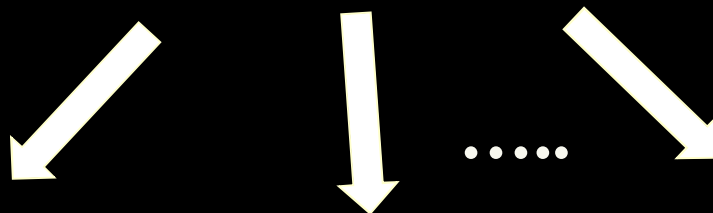
Con la
Plataforma

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

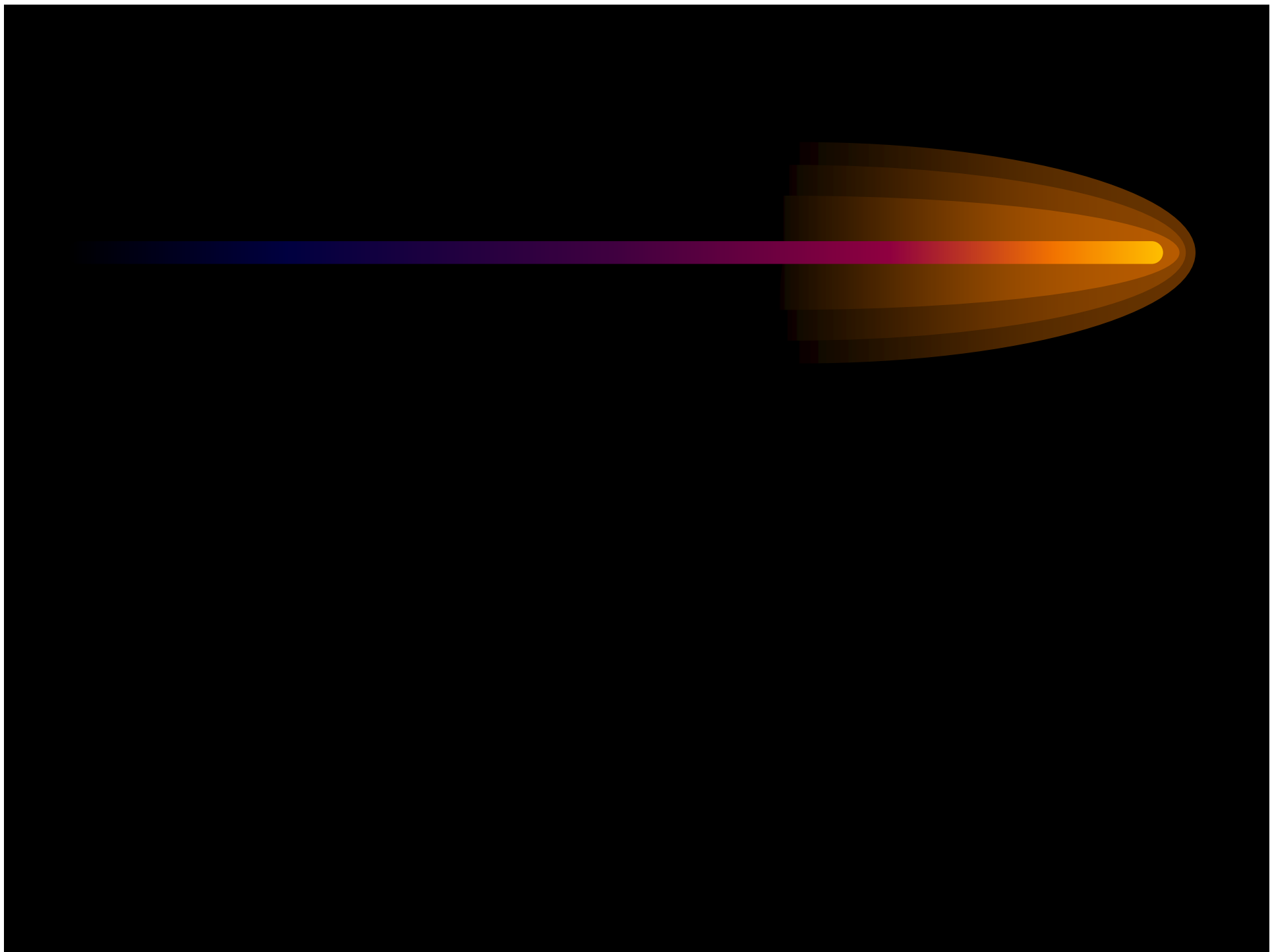
La idea es llegar a millones de copias

3.- Plataformas

Inversión 1 -> Ganancia 100 a 10,000

4.- Base

Inversión 1 -> Ganancia 1000 a





Sistema Evolutivo Generador de Nominas Federales

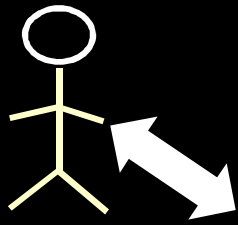
Juan Martín González Vázquez 1983

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**

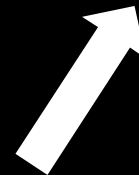
**Diccionario de Datos
de la Base de Datos
de la Organización**

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 sisyema, se hace el sitema evolutivo generador de sistemas.

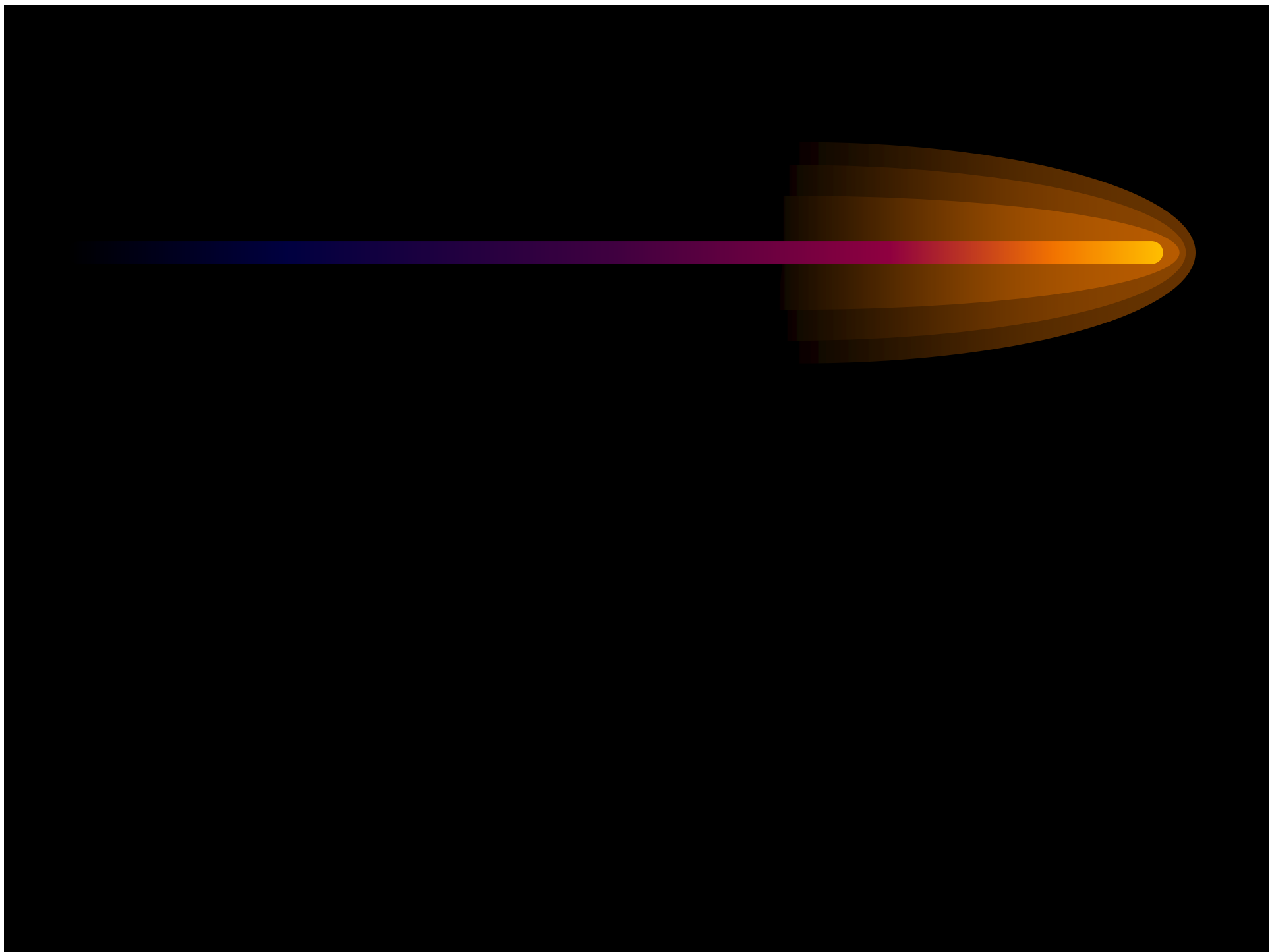
Y en lugar de enseñar a hacer una y otra vez el mismo sistema es necesario que aprendan a hacer y desarrollar plataformas y productos.

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.

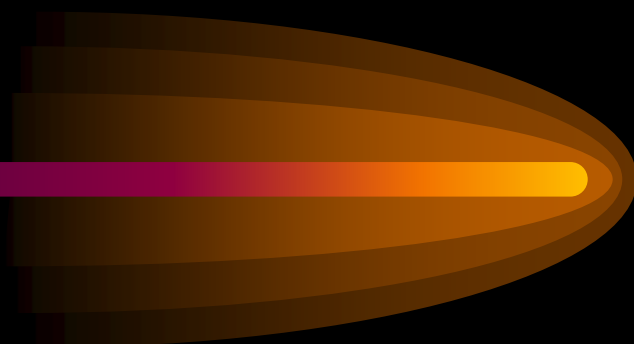
A principio de los 80's Christian Zempoaltecatl Ibarra, que precisamente tenía el problema de migrar de una plataforma a otra, porque su dependencia adquirió un nuevo equipo, desarrollo una herramienta para migrar lo desarrollado de una plataforma a otra.

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,

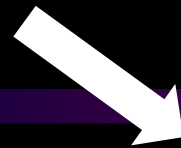
Por ejemplo, en varios de los caso de los archivos de texto es fácil construir los migradores, ya que lo único que se requiere es encontrar el formato de cada carácter (o sea la combinación de números que representan a cada carácter) en el archivo original y su equivalente en el nuevo formato y construir 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)

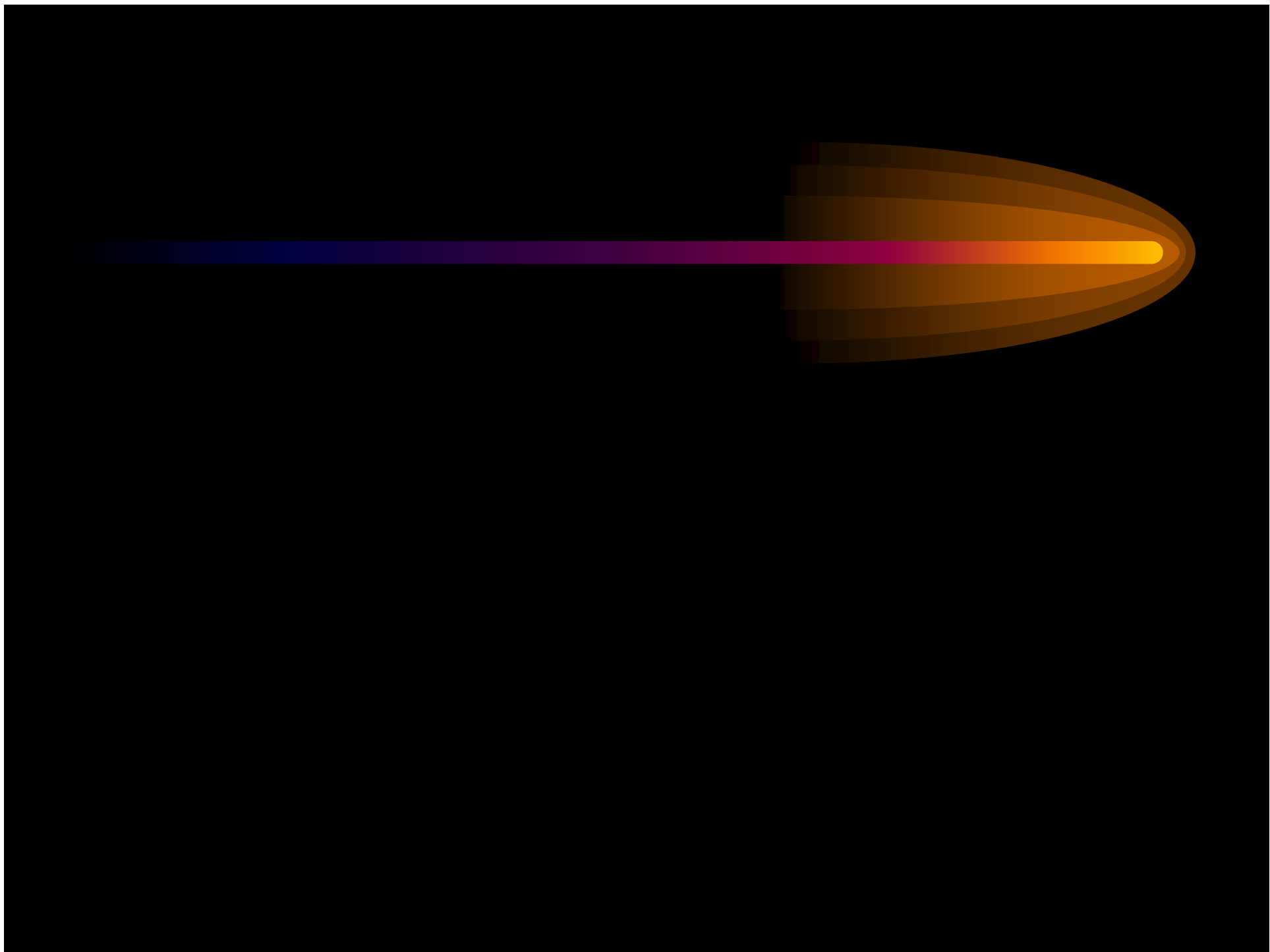
Busca el carácter en la columna de Formato original

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

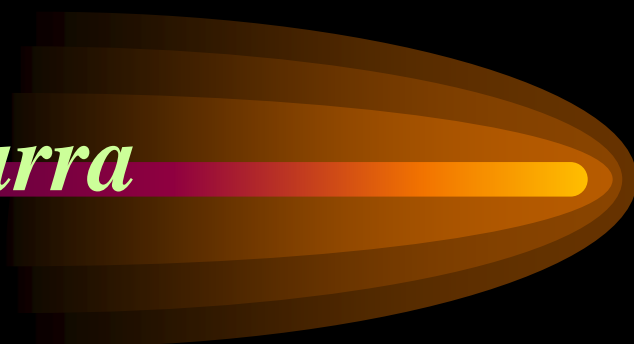
<http://www.fgalindosoria.com/eac/evolucion/>

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

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/linguisticamatematica/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

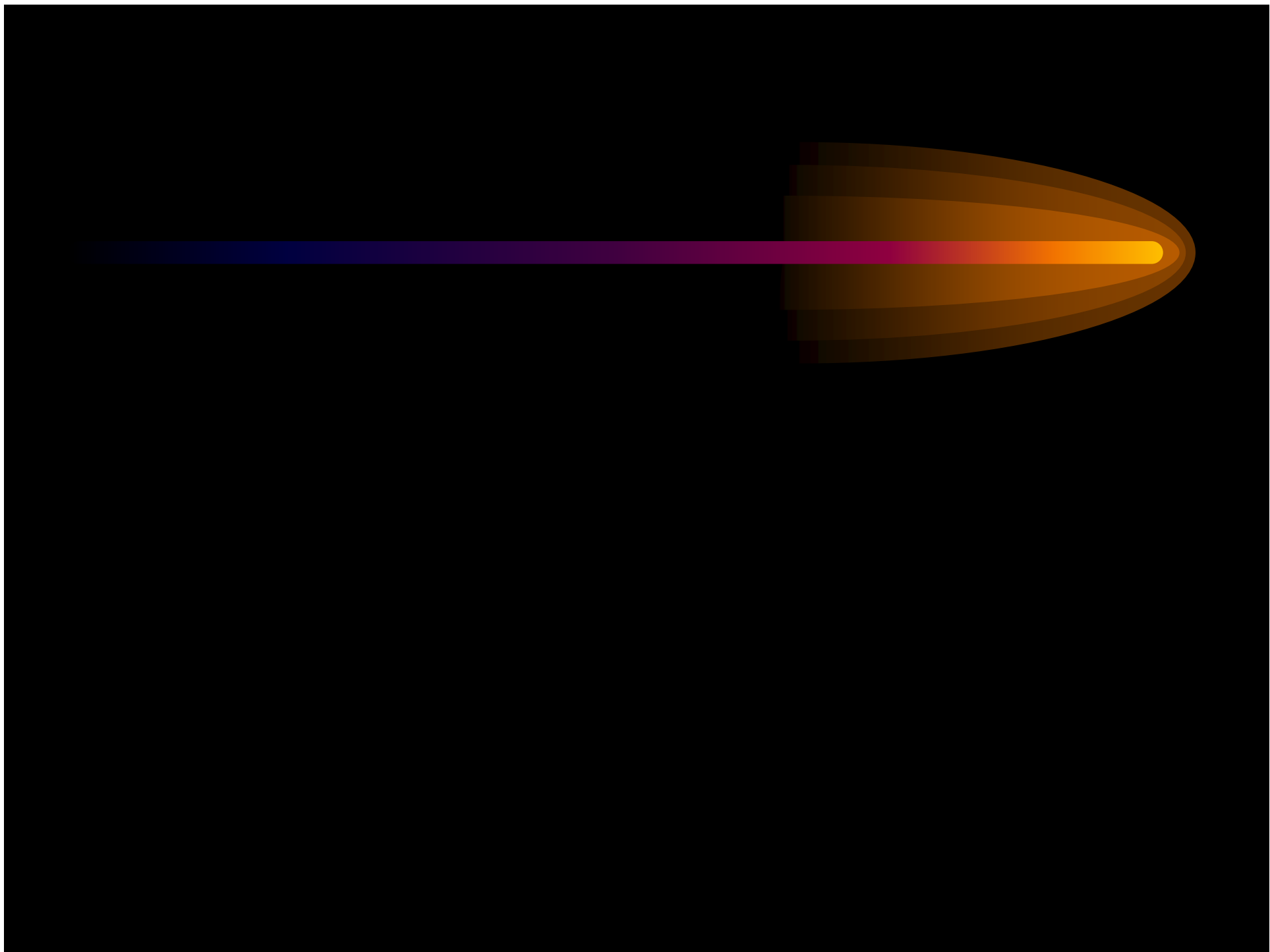
Word http://www.fgalindosoria.com/linguisticamatematica/enfoque_linguistico/enfoque_linguistico.doc

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





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

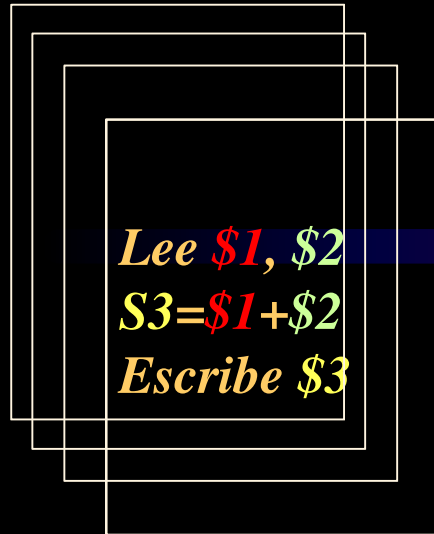


Arquitectura de los sistemas de información en los 70s
Vicente López Trueba

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

A decorative graphic element consisting of several overlapping, horizontal, teardrop-shaped ovals. The colors transition from dark purple on the left to bright orange and yellow on the right, creating a glowing effect. The shapes are layered, with some appearing more prominent than others.

Esqueletos Procesos

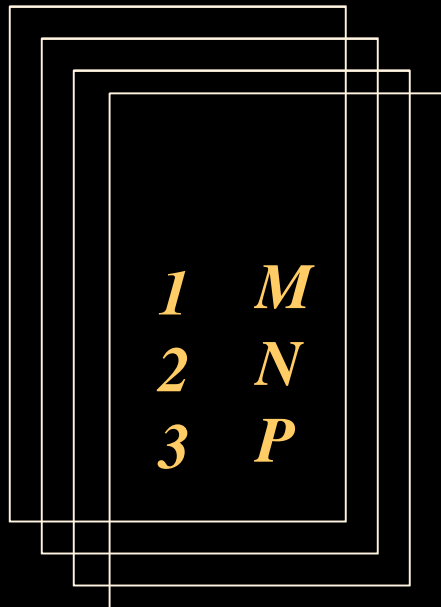


Vector Teórico Estructura
Orden en que se ejecutan los procesos



Generador de Sistemas

Parámetros Datos



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 es la de que exista una

Independencia Relativa

donde los elementos del sistema, como los datos, procesos y estructura queden separados y únicamente existe la relación mínima necesaria entre los 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

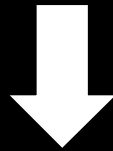
en los sistemas y programas tradicionales las componentes se encuentran revueltas y altamente interrelacionados, por lo que, un cambio "pequeño" en los datos o procesos o en el orden de atacar un problema ocasiona que prácticamente se tenga que modificar todo

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 las 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 los componentes del sistema

(como los datos, procesos y estructuras)



Va a facilitar que el sistema se transforme permanentemente a partir de los flujos de Materia, Energía e Información en los que esta inmerso

Sistemas evolutivos

Ahora el siguiente problema es buscar un mecanismo que nos permita separar los diferentes componentes del sistema a partir de la información que fluye o que describe al sistema

Sistemas Evolutivos

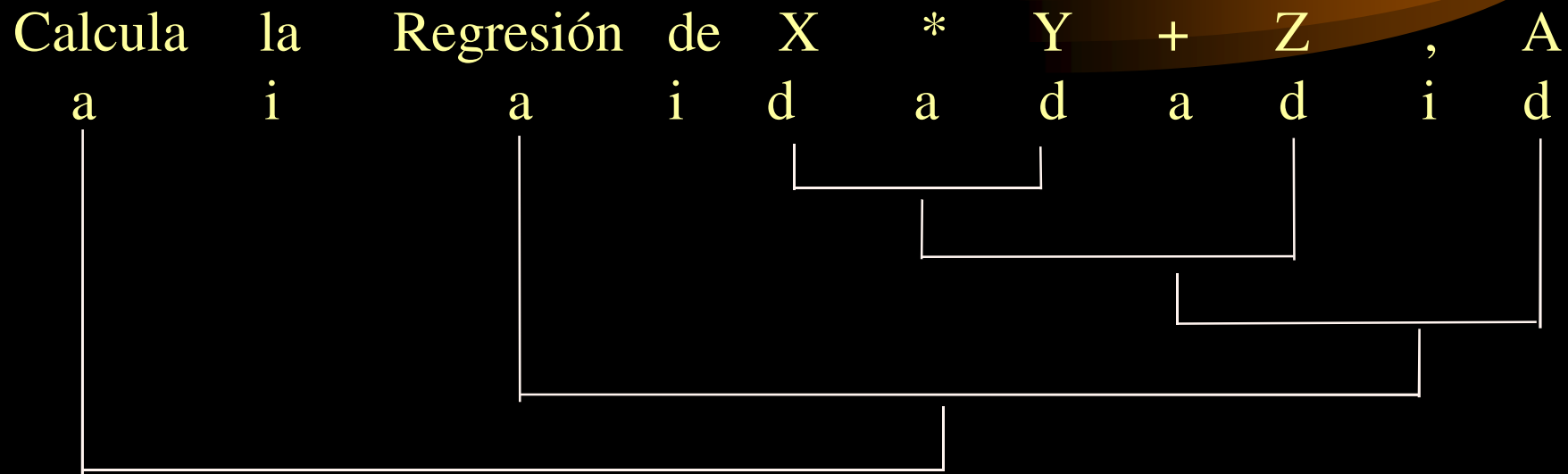
Enfoque lingüístico

Ver los problemas a resolver como oraciones de algún lenguaje y aplicar técnicas para encontrar y separar las componentes del sistema

por ejemplo en la oración:

Calcula la Regresión de $X * Y + Z$, A

Es relativamente fácil detectar los datos, acciones y estructura:



la estructura se indica con las líneas y representa el orden en que se ejecutan las acciones sobre los datos, los datos tienen el tipo **d**, las acciones el tipo **a** y las palabras no relevantes llevan **i** (ignora).

Sistemas evolutivos

Sistemas evolutivos

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

ENFOQUE EVOLUTIVO

un problema se puede resolver mediante los enfoque de los

*sistemas formales,
sistemas que aprenden
o sistemas evolutivos,*

Por ejemplo se puede hacer un sistema que reconozca la letra A usando el enfoque de los sistemas formales, donde se le da al sistema un conjunto de reglas que representan completamente a esa A específica y el sistema reconoce la letra A

también se puede usar el enfoque de los sistemas que aprenden y hacer un sistema que aprenda a reconocer letras (usando redes neuronales, algoritmos genéticos, etc.)

o usar el enfoque evolutivo.

Es un enfoque no una herramienta, por ejemplo existen redes neuronales usando el enfoque formal, el enfoque que aprende o el enfoque evolutivo.

Muchas redes neuronales y algoritmos genéticos son sistemas que aprenden y otros son sistemas evolutivos

El enfoque evolutivo se puede aplicar prácticamente con cualquier herramienta incluyendo redes neuronales, algoritmos genéticos y sistemas lingüísticos

mediante los sistemas evolutivos se busca que *sea el propio sistema* el que lleve a cabo acciones que le permitan construir su imagen de la realidad, mantenerla actualizada y usarla para interactuar con el medio.

cuando se desarrolla un Sistema Evolutivo

más que darle un conjunto de reglas prefijadas para resolver un problema, lo que se busca es darle la capacidad para que pueda construir y mantener su propia imagen de la realidad.

Cuando se esta modelando algo la estructura del modelo esta interrelacionada con la estructura del objeto o problema que se esta modelando.

Si la arquitectura del modelo es mas simple o no corresponde a la del problema la modelación se puede complicar.

si la arquitectura natural del problema es en paralelo y se utiliza para modelarlo un sistema secuencial, el método de solución se puede volver mas complejo y difícil de desarrollar,

lo mejor es tratar de que la arquitectura del sistema se acerque a la del problema o lo absorba.

si quiero representar un sistema real mediante reglas estáticas o sistemas formales, tarde o temprano la representación se queda desfasada de la realidad.

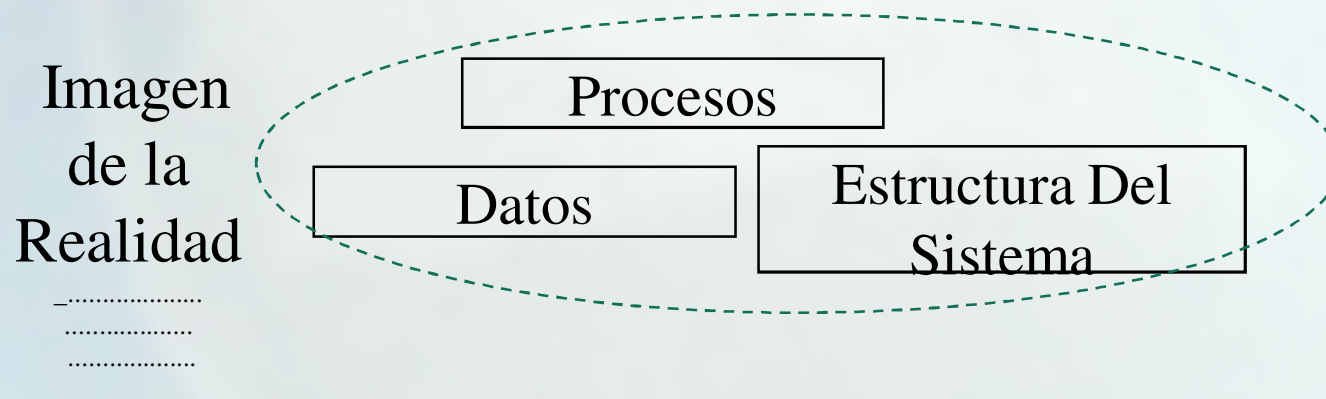
Los sistemas reales son sistemas evolutivos y la mejor forma de representarlos debe ser evolutiva.

Si se construye el sistema de información de tal forma que los *componentes sean independientes en forma relativa entre sí*

y se da una interrelación de tal forma que cualquier cambio en la descripción del sistema se refleje en tiempo real en el sistema de información,

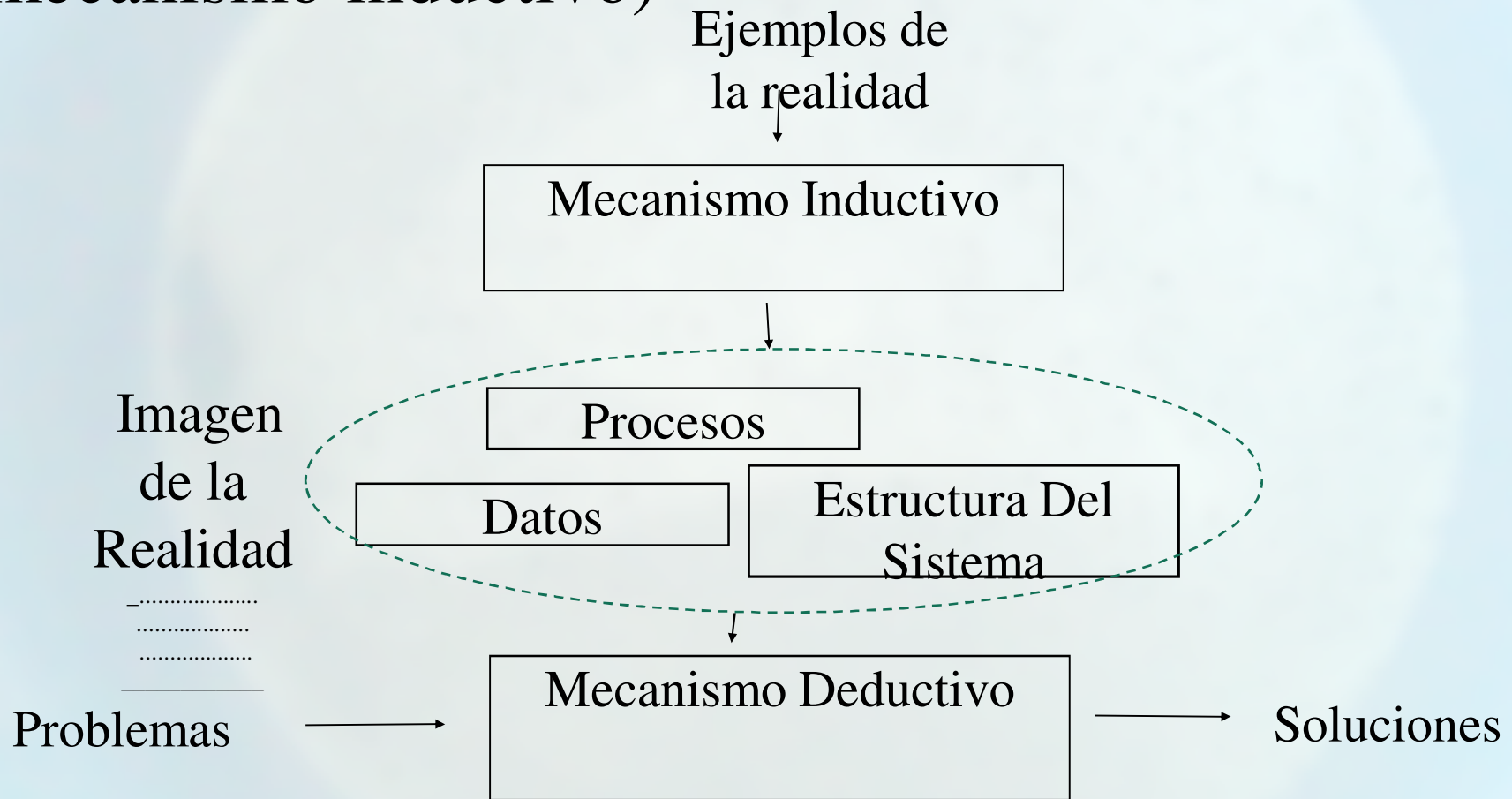
entonces se puede considerar que la *imagen de la realidad que tiene el sistema de información es cercana a la que se quiere reflejar.*

Datos, procesos y estructuras conforman la imagen de la realidad con independencia relativa



Retroalimentación para actualizar la imagen de la realidad

A partir de ejemplos de la realidad el sistema logra que evolucione la imagen de la realidad (mecanismo inductivo)

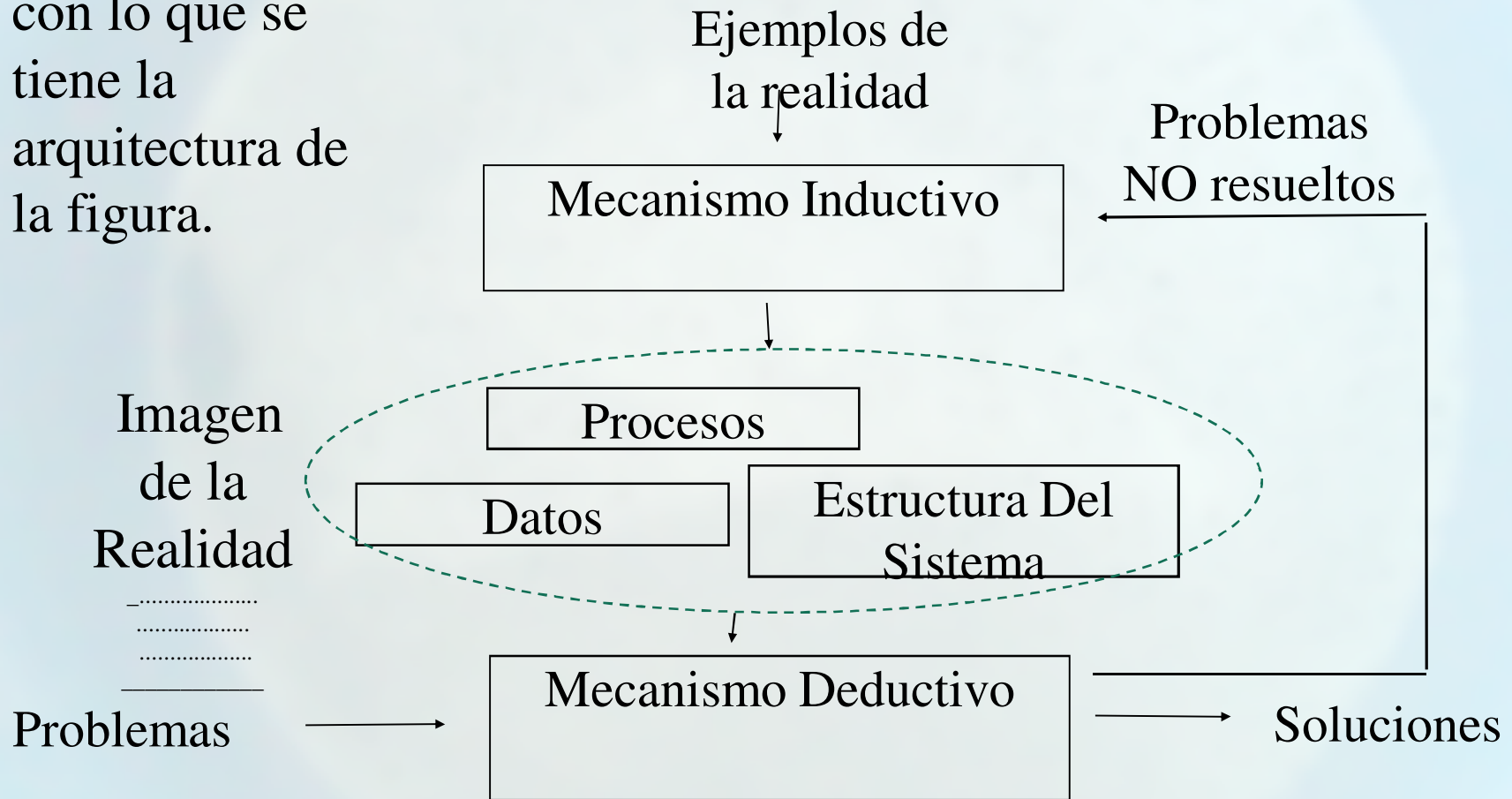


Retroalimentación para actualizar la imagen de la realidad

Entonces para contar con un **Sistema Evolutivo**

hace falta *dotar a esta herramienta con la capacidad de actualizar su imagen de la realidad en tiempo real*

con lo que se
tiene la
arquitectura de
la figura.



Retroalimentación para actualizar la imagen de la realidad

Existen muchos mecanismos de percepción y construcción de la imagen de la realidad,

como ejemplo mostraremos

las gramática evolutivas (Enfoque Lingüístico)

y las

Redes Neuronales y matrices evolutivas

se partirá de que

*el lenguaje de comunicación con el exterior
(o sea el lenguaje con el que se plantean los
problemas al Sistema Evolutivo y con el cual
éste responde a los requerimientos)*

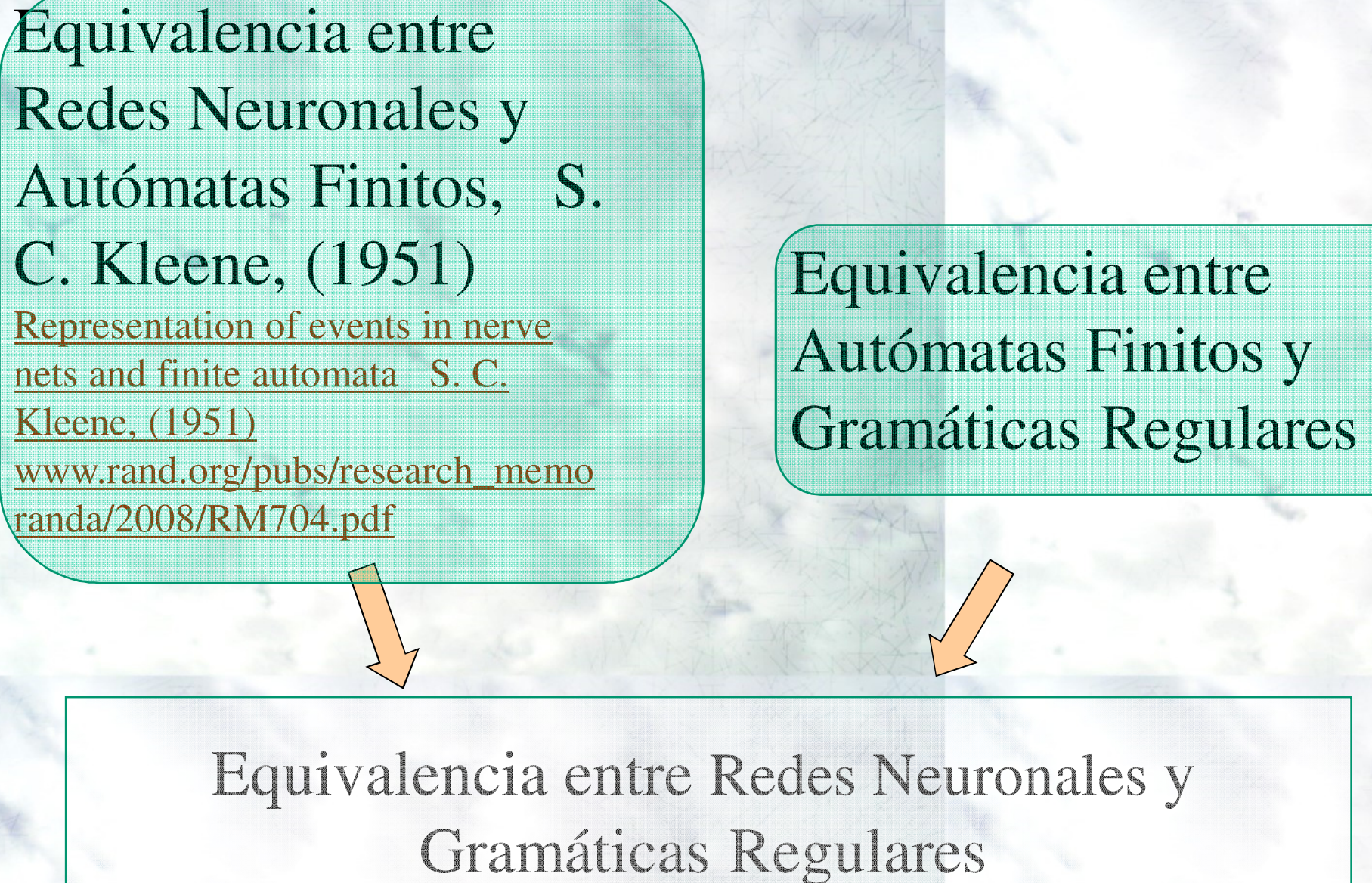
*y el lenguaje con el que el Sistema Evolutivo
construye su imagen de la realidad
son el mismo.*

Equivalencia entre
Redes Neuronales y
Autómatas Finitos, S.
C. Kleene, (1951)

Representation of events in nerve
nets and finite automata S. C.
Kleene, (1951)

www.rand.org/pubs/research_memo_randa/2008/RM704.pdf

Equivalencia entre
Autómatas Finitos y
Gramáticas Regulares



Equivalencia entre Redes Neuronales y
Gramáticas Regulares

Relaciona la Neurología con la Lingüística

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 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, un texto, un sonido, las señales de voz, las imágenes, una huella digital, la señal de un electrocardiograma, etc.

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.

Una gran cantidad de problemas aparentemente diferentes se pueden reducir en primera instancia al mismo sistema, ya que prácticamente se puede atacar cualquier problema que se pueda representar como una cadena de bits o caracteres, como por ejemplo *el reconocimiento de imágenes, texto, traductores, sistemas expertos, señales biofísica, voz, etc.*

Lingüística Matemática

Chomsky

Gramáticas Generativas

Reescritura

Reglas de Reescritura

El operador de Reescritura, un Operador fundamental $\rightarrow$

$\alpha \rightarrow \beta$

Significa que α se puede sustituir o rescribir por β

EJEMPLO

SISTEMAS EVOLUTIVOS DE REESCRITURA

Regla de Reescritura

$A \rightarrow B$

Sistemas Evolutivos de Reescritura

Desarrollados junto con Itztli

(Horacio Alberto García Salas) en 1995

FGS

Es relativamente fácil construir un sistema de información que representa a los sistemas de reescritura, por ejemplo el siguiente sistema formado por un programa y un archivo con dos columnas :

A_1	B_1
A_2	B_2
""	""
A_n	B_n

```
Programa()  
{i=1  
  lee Ax  
  mientras ((Ax != Ai) y (no fin de archivo))i++  
  si (no fin de archivo) escribe Bi  
  sino escribe "no reconozco Ax"  
}
```

Sistemas Evolutivo de Reescritura

A_1	B_1
A_2	B_2
""	""
A_n	B_n

```
Programa()  
{  
  i=1  
  lee Textox  
  mientras ((Textox != Textoi) y (no fin de archivo))i++  
  si (no fin de archivo) escribe Traduccióni  
  sino //entra al dialogo  
    escribe "desconozco Textox dame su traducción"  
    lee Traducciónx  
    almacena en el archivo Textox , Traducciónx  
}
```

sistema elemental traductor de Idiomas

le damos la entrada "Hola Perro",

si la computadora no encuentra en su lista esta frase,
entonces pregunta, ¿ que es "Hola Perro "?,

le decimos "Hello Dog",

entonces la computadora almacena en su primera columna
"Hola perro" y en la otra "Hello Dog",

la siguiente vez que introduzcamos "Hola Perro" la
computadora va a decir "Hello dog".

Traductor automático

tabla que contiene en la primera columna el texto en un idioma y en la segunda su traducción a otro idioma.

texto	traducción
este es un perro	this is a dog
el perro blanco	the white dog
""	""
el perro negro	the black dog

Programa()

```
{  
  i=1  
  lee Textox  
  mientras ((Textox != Textoi) y (no fin de archivo))i++  
  si (no fin de archivo) escribe Traduccióni  
  sino //entra al dialogo  
  escribe "desconozco Textox dame su traducción"  
  lee Traducciónx  
  almacena en el archivo Textox , Traducciónx  
}
```

Este programa es muy simple de hacer y tiene la ventaja de que no requiere tener almacenado el conocimiento previamente, si el archivo estuviera vacío y se preguntara por Texto_x el programa pediría Traducción_x y ya tendría la regla $\text{Texto}_x \rightarrow \text{Traducción}_x$

el sistema evolutivo puede empezar a construir la base de conocimiento y 'aprender' desde cero, en tiempo real y fácilmente.

Una opción es hacer que el sistema evolucione a partir de su interacción con el entorno, por ejemplo navegando en la red y encontrando equivalencias entre 2 textos con el mismo significado en diferentes idiomas

Encontrar los datos de un sistema es relativamente simple, y con algunas de las técnicas mencionadas al comentar la independencia relativa, se pueden encontrar los procesos (semántica) o esqueletos del sistema,

sin embargo encontrar la estructura del sistema en forma automática, aunque es relativamente simple y existen métodos y técnicas desde los años 60 del siglo XX,

como los desarrollados para el reconocimiento sintáctico de patrones (Syntactic Pattern Recognition), la inferencia gramatical y los métodos lingüísticos matemáticos (algebraicos y analíticos), estos son poco conocidos,

por lo que veremos una introducción a los métodos de lingüística matemática y en particular de Inferencia Gramatical que es precisamente el área dedicada a encontrar las estructuras

Gramática

*Representación lingüística de la
estructura de un lenguaje*

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

Inferencia Gramatical

El problema que ataca la Inferencia Gramatical consistir básicamente en encontrar la Gramática (Estructura) que describe a un lenguaje dado a partir de ejemplos de oraciones del Lenguaje.



Inferencia gramatical

Las herramientas de la Inferencia Gramatical
trabajan con la estructura de las oraciones
buscando encontrar una estructura general
(o reglas sintácticas)
a partir de estructuras particulares
(u oraciones).

Muchos de los métodos de inferencia gramatical se basan en las operaciones básicas de
Factorización,
Distribución y
Recursividad

por lo que a continuación se explicaran estas operaciones.

Factorización Lingüística

La factorización lingüística consiste en la búsqueda de cadenas que se repiten varias veces en un texto y verlas como un factor dentro del texto.

Algunas propiedades matemáticas de los sistemas lingüísticos

www.fgalindosoria.com/linguisticamatematica/propiedades_matematicas_sistemas_linguisticos/prop_mate_sistemas_linguisticos.pdf

Factorización Lingüística

La factorización lingüística consiste en la búsqueda de cadenas que se repiten varias veces en un texto y verlas como un factor dentro del texto.

Algunas propiedades matemáticas de los sistemas lingüísticos

www.fgalindosoria.com/linguisticamatematica/propiedades_matematicas_sistemas_linguisticos/prop_mate_sistemas_linguisticos.pdf

Recordemos que la factorización algebraica consiste en encontrar los factores comunes en una expresión algebraica y sacarlos de la expresión, como se ve a continuación:

$$\underline{a}b + \underline{a}c = a(b+c)$$

$$3x + 3y = 3(x + y)$$

$$a(b * c) + a(e/f) = a(b*c+e/f)$$

En la factorización lingüística,
si por ejemplo se tienen las siguientes oraciones:

Juan es hermano de Pedro y
Juan estudia en UPIICSA

se puede detectar que en las dos oraciones se encuentra
presente la palabra **Juan** y que un párrafo equivalente
sería:

Juan es hermano de Pedro y estudia en UPIICSA

Si se observa lo que se ha hecho es detectar que la
palabra **Juan** era común a las dos oraciones por lo que
se factorizó (o sea que se sacó como factor común) y se
obtuvo un párrafo donde sólo aparece una sola vez.

Distribución Lingüística

La distribución lingüística es la operación inversa de la factorización

Algunas propiedades matemáticas de los sistemas lingüísticos

www.fgalindosoria.com/linguisticamatematica/propiedades_matematicas_sistemas_linguisticos/prop_mate_sistemas_linguisticos.pdf

Distribución Lingüística

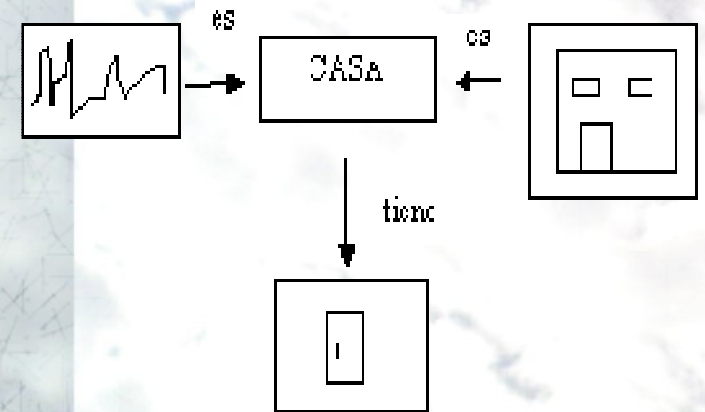
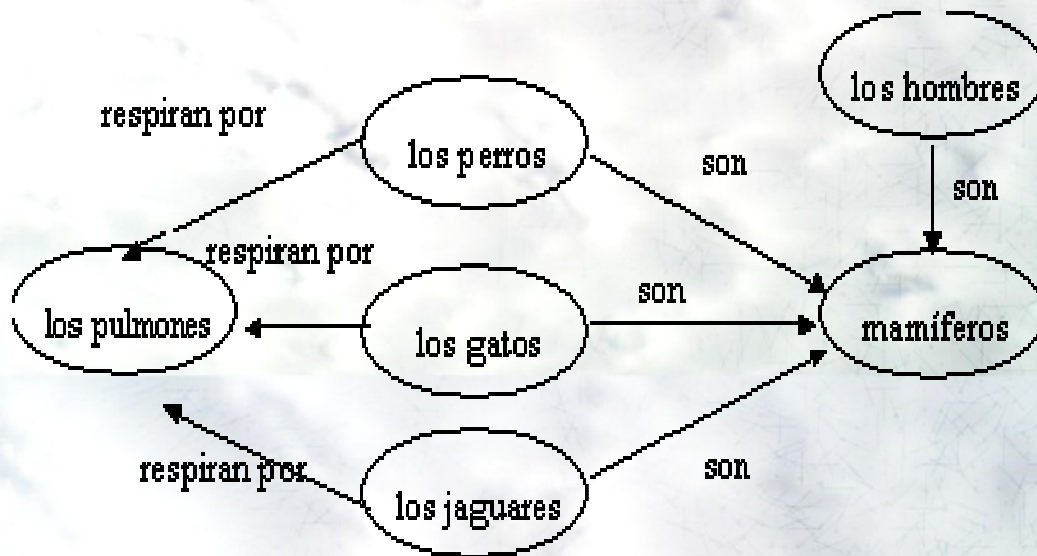
La distribución lingüística es la operación inversa de la factorización y para entenderla veremos un ejemplo basado en problemas de representación del conocimiento.

Existen varios mecanismos para representar el conocimiento, pero uno de los más usados son las redes semánticas, donde una red semántica está formada por hechos o reglas de la forma objeto relación objeto (o r o),

UPIICSA del IPN, 1991, 1992 México

Redes Semánticas Ampliadas Jesús Olivares Ceja

Red Semántica Ampliada (RSA, la información que se registra en sus nodos, puede ser: etiquetas, figuras, conceptos, procesos, etc. La semántica de cada nodo está dada por las relaciones que tiene con los demás



Sistema Evolutivo para Representación del Conocimiento

http://www.fgalindosoria.com/eac/evolucion/libro_sistemas_evolutivos/VII6-SER.pdf

www.fgalindosoria.com/informaticos/investigadores/Jesus_Manuel_Olivares_Ceja/serc/serc_tesis_jesus_olivares.pdf

Jesús Olivares en su tesis de licenciatura
"Sistemas Evolutivos para Representación de Conocimiento"
desarrollada en la UPIICSA del IPN,
planteo un mecanismo evolutivo para construir un base de
conocimientos basada en una Red Semántica Aumentada
aplicar una serie de técnicas de Inferencia Gramatical
basadas en la distribución lingüística,
para transformar una oración compuesta de múltiples elementos
en un conjunto de oraciones semánticamente equivalentes a la
original,
donde cada oración final es de la forma **oro**
(**objeto relación objeto**),
por lo que, son fácilmente representadas con una red semántica

por ejemplo a partir de la oración:

Juan es hermano de Pedro

se detecta que **Juan** es un *objeto*, **es hermano de**
es una *relación* y **Pedro** es otro *objeto* por lo
que de esta oración se obtienen directamente
los componentes de una red semántica

Por ejemplo :

Juan estudia en el Poli, trabaja en el INEGI y vive en Lindavista.

Juan	es un objeto	o0
estudia en el	es una relación	r1
poli	es un objeto	o1
,	es un separador	+
trabaja en el	es una relación	r2
INEGI	es un objeto	o2
y	es un separador	+
vive en	es una relación	r3
Lindavista	es un objeto	o3

queda representada como:

$o0 (r1o1 + r2o2 + r3o3)$

El paréntesis que esta después de o0 refleja que tiene relación con todos los demás elementos de la oración.

Si se toma

$$o0 (r1o1 + r2o2 + r3o3)$$

y se le aplica la operación de distribución del álgebra queda:

$$o0 (r1o1 + r2o2 + r3o3) = o0r1o1 + o0r2o2 + o0r3o3$$

y si sustituimos el significado de los diferentes elementos tenemos:

$$\begin{array}{ll} o0r1o1 & \text{Juan estudia en el poli} \\ + & , \\ o0r2o2 & \text{Juan trabaja en el INEGI} \\ + & \text{y} \\ o0r3o3 & \text{Juan vive en Lindavista.} \end{array}$$

o sea que a partir de una oración compleja se obtuvieron tres oraciones simples del tipo oro que mantienen el significado de la oración original

Cuando nos comunicamos factorizamos
Cuando almacenamos distribuimos

Jesús Olivares Ceja

Las gramáticas no son el único mecanismo para representar la realidad y en particular para representar estructuras, a continuación mostraremos otro **mecanismo basado en matrices**, donde también **es relativamente fácil encontrar** los componentes, y en particular **la estructura de múltiples sistemas**, como por ejemplo **los de una red neuronal**

MATRICES EVOLUTIVAS

Matrices Evolutivas y Dinámica
Dimensional

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

En el caso de nuestro modelo matricial de las redes neuronales

desde el principio se presentó el problema de obtener los valores de las neuronas

y es ahí donde a mediados de los 80's,

Cuitláhuac Cantú, al estar trabajando sobre reconocimiento de imágenes llegó en forma independiente a una representación matricial prácticamente equivalente a la encontrada para las redes neuronales, pero con capacidades evolutivas.

A continuación se comenta la idea general (aunque muy básica) del método de Cuitláhuac Cantú

se parte de que
originalmente la matriz está vacía
y lo que hace el sistema es
buscar una imagen,
como no encuentra nada
la coloca en el primer renglón,
cuando llega la segunda imagen,
si son similares la reconoce y la acumula con la
primera
si no son similares entonces la coloca en el
siguiente renglón
y así sucesivamente.

cada imagen se representa como un vector,

de tal forma que por ejemplo,
si se tiene un gato, un perro y un ratón,
cada uno de ellos se almacena como
un vector

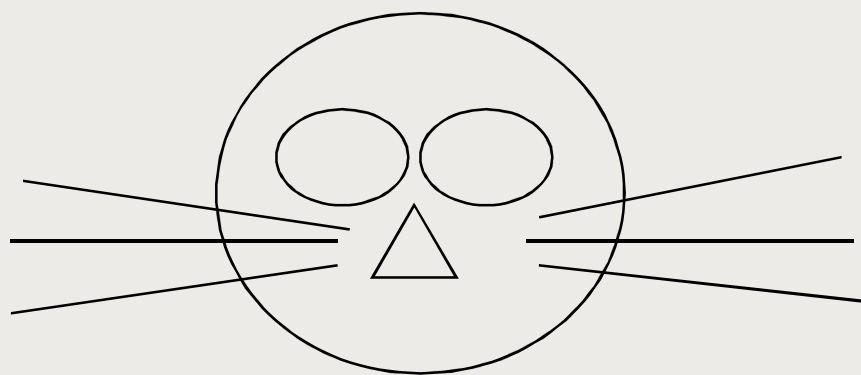
y entre los tres forman una matriz
como la siguiente

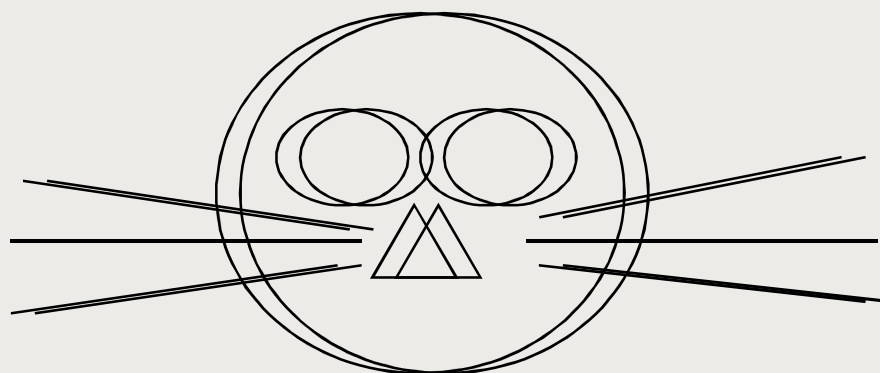
a1	a2	a3	a4	a5	a6	a7	a8	a9	
1	0	1	0	0	1	0	1	0	gato
0	1	0	1	0	1	1	0	1	perro
1	1	0	0	1	0	0	1	1	ratón

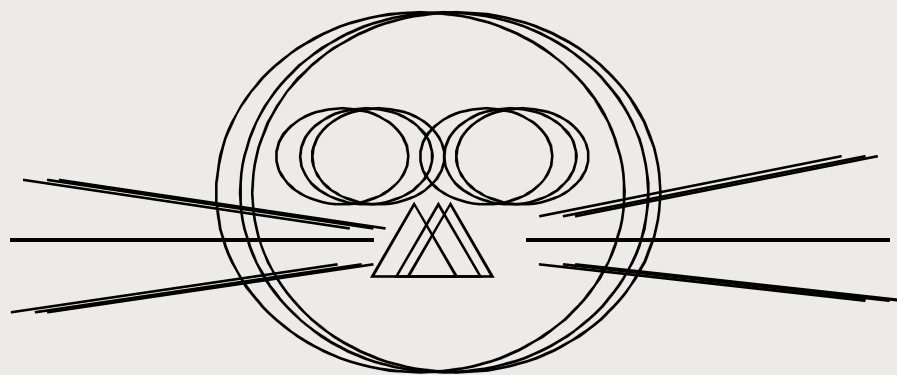
Si se perciben varios gatos,
en lugar de almacenar cada gato en un
vector independiente
se suman los vectores que representan
cada uno de los gatos y el resultado se
toma como la representación de la
estructura general del gato
y se almacena en la matriz.

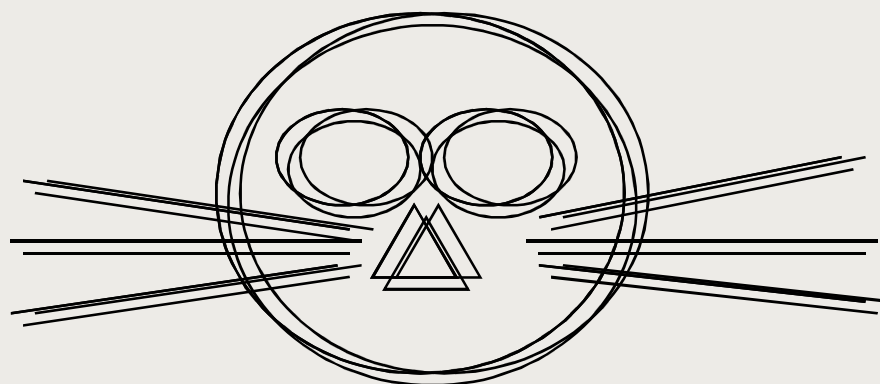
(5 1 4 0 0 5 0 4 1 20) gato

Lo anterior es equivalente a que cada gato se dibujara en
un acetato y posteriormente se superpusieran los acetatos,
con lo que las características repetitivas del gato quedan
más recalcadas y equivale a números mayores en el vector
que lo representa.









Por otro lado si se tienen varios gatos, perros y ratones, la matriz seria de la forma:

a1	a2	a3	a4	a5	a6	a7	a8	a9	h	
5	1	4	0	0	5	0	4	1	20	gato
0	3	0	3	0	2	3	2	1	14	perro
1	1	0	0	1	0	0	1	1	5	ratón

cada renglón representa un tipo de objeto y h es un valor donde se acumula el número de puntos en el vector

(En este caso se puso h como la suma de los valores del vector por facilidad del ejemplo, sin embargo existen otros métodos para asignar este valor).

Si se desea reconocer un nuevo objeto, se representa también como un vector, se multiplica por la matriz, se ve en que renglón se obtuvo el máximo valor y se le asocia a ese renglón el objeto.

Por ejemplo si tomamos la matriz anterior y llega el objeto 1 0 1 0 0 1 0 1 0, se multiplica por la matriz y les restamos el valor de h quedando:

-2 para el gato, -10 para el perro y -3 para el ratón.

De donde se propone que el objeto reconocido es un gato.

Como siguiente punto el sistema acumula el nuevo vector en la matriz, con lo que, en el ejemplo anterior el nuevo vector se suma al renglón del gato y la matriz queda:

a1	a2	a3	a4	a5	a6	a7	a8	a9	h	
6	1	5	0	0	6	0	5	1	24	gato
0	3	0	3	0	2	3	2	1	14	perro
1	1	0	0	1	0	0	1	1	5	ratón

Con lo que, el sistema evolutivo esta transformando permanentemente su imagen de la realidad.

Como se puede observar este método reúne la característica de que esta evolucionando en forma natural y encontrando la imagen acumulada (y por tal, la imagen promedio),

con lo que no existe un proceso previo de aprendizaje y otro de aplicación, sino que por el proceso natural de conocer y reconocer las imágenes va evolucionando.

Existen muchísimas aplicaciones y estructuras de las matrices evolutivas, como las que se muestran a continuación

Aplicación de los Sistemas Evolutivos a la Composición musical
Desarrollado por Itztli (Horacio Alberto García Salas)

Musical Composer Based on Detection of Typical Patterns in a Human Composer's Style

Horacio Alberto García Salas, Alexander Gelbukh

Sistema de Mimetismo basado en gramáticas para ocultamiento de Información

Fátima Margarita Lechuga Blanco y Mario César Lima Rodríguez

Mundo artificial basado en Sistemas Evolutivos
Desarrollado por Paola Neri Ortiz

Ejemplos

Musical Composer Based on Detection of Typical Patterns in a Human Composer's Style

Horacio Alberto García Salas, Alexander Gelbukh

Laboratorio de Lenguaje Natural,. Centro de Investigación en Computación,. Instituto Politécnico Nacional

Sistema Evolutivo para el Diagnóstico de Fallas en Maquinas Rotatorias
Eduardo De la Cruz Sánchez, Luis C. Longoria Gándara, Rodolfo A. Carrillo Mendoza

Sistema Evolutivo de Reconocimiento de Formas en dos Dimensiones
Diana Karla García García, Sergio Salcido Bustamante, Alfonso Ventura Silva

Sistema evolutivo reconocedor de huellas digitales

Paola Neri Ortiz, Celia Pérez, Araceli Tlamanca

Universidad Anahuac de Xalapa

Aplicación de los Sistemas Evolutivos en el Análisis de Espectros de Rayos Gamma

Luis E Torres Hernández, Luis C. Longoria Gándara, Antonio Rojas Salinas

REFORNED: Reconocimiento de Imágenes por Medio de Redes Neuronales, Sistemas Evolutivos y Distribuidos

Alejandría Salazar Torres

SAEVIUS DEUS, que significa "La ira de Dios", videojuego con Inteligencia Artificial

Desarrollado por Alfredo Ramos Flores, Jesús Rodríguez Salazar, Jorge Israel Toledo

Alvarado, José Rafael Vega Trujillo

ESpertCOM Sistema Experto Evolutivo en Banca Múltiple

José de Jesús Acosta Marrón y Rodrigo Arenas Arriola

El Sistema Experto ESpertCOM utiliza un par de matrices de dos dimensiones que permanentemente están cambiando. Originalmente, la matriz evolutiva está vacía, por lo que la base de conocimientos que representa también lo está.

Conforme detecta nuevos síntomas y reglas aumentan los renglones y columnas de la matriz

Sistema Evolutivo Reconocedor de Textos

Jesús Manuel Olivares Ceja

Algunos Trabajos de Grado y Titulación relacionados con Matrices Evolutivas

Doctorado

Horacio Alberto García Salas, "Modelo Generativo de Composición Melódica con Expresividad"

Tesis para obtener el Grado de Doctor en Ciencias de la Computación

Laboratorio de Procesamiento de Lenguaje Natural, Centro de Investigación en Computación CIC, Instituto Politécnico Nacional IPN, México, Mayo del 2010

<http://www.gelbukh.com/thesis/Horacio%20Alberto%20Garcia%20Salas%20-%20PhD.pdf>

Maestría

Fausto Pavel Priego Pérez, "Reconocimiento de Imágenes del Lenguaje de Señas Mexicano"

Tesis de Maestría en Ciencias de la Computación

Centro de Investigación en Computación (CIC) del Instituto Politécnico Nacional (IPN), México, 16 de Enero del 2013

Profesional

Horacio Alberto García Salas, "Aplicación de los sistemas evolutivos a la composición musical", IPN-UPIIICSA, México, 5 de Marzo de 1999

http://www.fgalindosoria.com/informaticos/investigadores/Itztli/secm/aplicacion_sev_composicion_musical.pdf

José de Jesús Acosta Marrón y Rodrigo Arenas Arriola, "ESpertCOM Sistema Experto Evolutivo en Banca Múltiple", IPN-ESCOM, México, Mayo del 2001

El Sistema Experto ESpertCOM utiliza un par de matrices de dos dimensiones que permanentemente están cambiando. Originalmente, la matriz evolutiva está vacía, por lo que la base de conocimientos que representa también lo está. Conforme detecta nuevos síntomas y reglas aumentan los renglones y columnas de la matriz.

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

Maria Paola Neri Ortiz, "Mundo Artificial basado en sistemas evolutivos", IPN-ESCOM, México, Junio de 2001

http://www.fgalindosoria.com/informaticos/investigadores/Paola_Neri_Ortiz/bichosevolutivos/TT1doc.htm

Fátima Margarita Lechuga Blanco y Mario César Lima Rodríguez, "Sistema de mimetismo basado en gramática para ocultamiento de información", IPN-ESCOM, México, Junio del 2002

http://www.fgalindosoria.com/linguisticamatematica/sistema_de_mimetismo_basado_en_gramaticas/index.htm

Matrices Evolutivas y Dinámica Dimensional

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

Un ejemplo donde se pueden aplicar las matrices evolutivas es en el área de los sistemas expertos

Si partimos de un sistema experto de diagnostico formado por reglas de la forma

$S_1 \ S_2 \ S_3 \dots S_n \Rightarrow D_i : T_i$

donde S_j son los síntomas, D_i los diagnósticos y T_i los tratamientos
(un conjunto de síntomas generan un diagnostico y un tratamiento)

El sistema experto puede representarse como una matriz, cada columna corresponde a un síntoma y cada renglón a una regla del experto.

	S1	S2	S3	...	S _n
r1	1	0	0	...	0
r2	0	1	1	...	0
...	...	...	...	...	...
r _m	1	0	1	...	1

De tal manera que si aparece un 1 en el renglón i y la columna j , quiere decir que la regla i necesita el síntoma j para cumplirse.

Por ejemplo, si se tiene la siguiente matriz:

	S1	S2	S3	S4	S	S6
regla1 3.2	0	4.0	6.5	0	0	
regla2 2.0	1.3	0	0	1.5	0	
regla3 0	3.0	0	0	4.2	2.0	

y llega un problema con los síntomas

(1.0 0 1.5 0 0 0.7)

entonces se multiplica la matriz por el vector quedando el resultado:

regla1 9.2

regla2 2.0

regla3 1.4

de donde se propone la regla1 como el resultado del sistema.

Una de las ventajas de representar al sistema experto mediante una matriz es que ahora se puede ver y atacar con múltiples herramientas incluyendo entre otras la lógica difusa, redes neuronales, teoría de la medida y sistemas evolutivos.

En particular podemos aplicar las técnicas de matrices evolutivas

Se parte de que la matriz está originalmente vacía. *Mev0 o Matriz Evolutiva 0* que no tiene dimensiones, renglones, columnas, capas, o valores en la matriz

cuando llegan los primeros síntomas y no encuentra nada almacena el vector en el primer renglón y asigna el diagnostico (le puede pedir el diagnostico al experto humano).

Por ejemplo, si llegan los síntomas
(1.2 0 0 2.3 0 0.7)

el sistema busca en la matriz el diagnostico,
pero como está vacía no los encuentra,
entonces pregunta por el diagnostico
y genera una matriz con un renglón.

	S1	S2	S3	S4	S5	S6	diagnostico
r1	1.2	0	0	2.3	0	0.7	D1

Si llegan ahora los síntomas
(0.7 0 0 0 3.2 0),
como la colisión con el vector almacenado
es baja, pregunta por el diagnostico,
si el nuevo diagnostico es diferente crea
un nuevo renglón,
quedando la matriz:

	S1	S2	S3	S4	S5	S6	
r1	1.2	0	0	2.3	0	0.7	D1
r2	0.7	0	0	0	3.2	0	D2

Si llega una cadena de síntomas con alta colisión con D1, como por ejemplo:

(1.0 0 0 1.8 0 1.2)

el sistema emite el diagnostico D1 y acumula el nuevo síntoma sobre la matriz anterior en el renglón 1:

	S1	S2	S3	S4	S5	S6		# de acumulados
r1	2.2	0	0	4.1	0	1.9	D1	2
r2	0.7	0	0	0	3.2	0	D2	1

Al tener el vector acumulado se están reforzando los síntomas y en forma natural cambian los pesos.

El sistema siempre se transforma.

Si llega una cadena de síntomas con alta colisión los suma al renglón diagnosticado.

Si llega una cadena y el experto indica que corresponde a un diagnostico presente, la nueva cadena se acumula a este diagnostico.

Si llega una nueva cadena con un síntoma no conocido crea una nueva columna y a partir de ese momento reconoce el nuevo síntoma

Si llega una nueva cadena no reconocida crea un nuevo renglón.

Si se toma el vector acumulado y cada síntoma lo dividimos entre el número de acumulados tenemos una regla promedio, por lo que, lo que estamos encontrando es la regla promedio asociada a un diagnostico y mediante el mecanismo evolutivo esta regla se está depurando y afinando cotidianamente,

o sea que, en forma natural y permanente se están encontrando y afinando las reglas del sistema.

El sistema siempre evoluciona y cambia su estructura permanentemente

Si llega una nueva **cadena con un síntoma no conocido** **crea una nueva columna en la gramática** y a partir de ese momento reconoce el nuevo síntoma

Si llega una nueva **cadena no reconocida** **crea un nuevo renglón en la gramática.**

lo que estamos diciendo es que **cada síntoma no reconocido genera una nueva variable que lo representa**, si la nueva variable es linealmente independiente con respecto a las otras, **esa variable representa una nueva dimensión del sistema**. Esto es un ejemplo de la **dinámica dimensional** o sea sistemas que **permanentemente están cambiando su número de dimensiones**

Espacios Transfinito Dimensionales y Dinámica Dimensional: Un Universo Fractal
www.fgalindosoria.com/transfinitoydinamicadimensional/

Matrices Evolutivas y Dinámica Dimensional
www.fgalindosoria.com/eac/evolucion/mev/

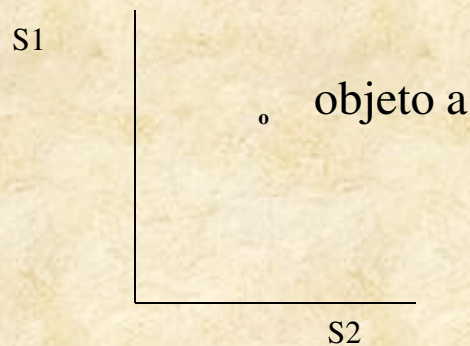
Una Representación N-Dimensional de los Sistemas Evolutivos

[Matrices Evolutivas y Dinámica Dimensional](http://www.fgalindosoria.com/eac/evolucion/mev/)
<http://www.fgalindosoria.com/eac/evolucion/mev/>

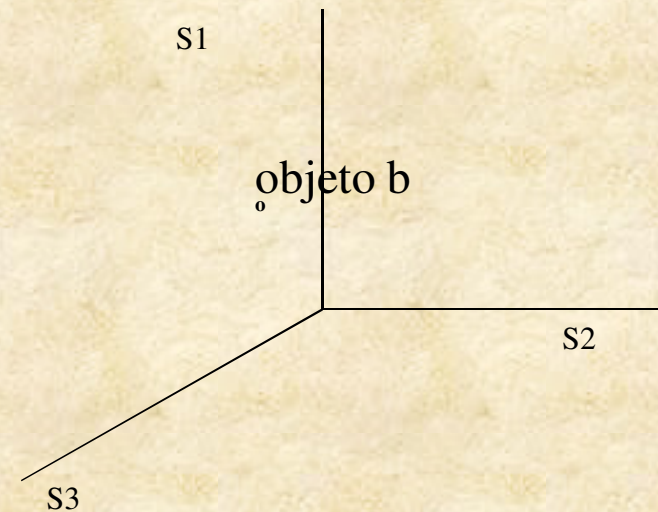
Espacios Transfinito Dimensionales y Dinámica Dimensional
Un Universo Fractal

<http://www.fgalindosoria.com/transfinitoydinamicadimensional/>

Si analizamos un poco mas a los sistemas evolutivos vemos que cada imagen, regla o cadena de síntomas se representa mediante un conjunto de n variables, si estas variables son independientes entre si entonces cada objeto (imagen, regla o cadena de síntomas promedio) representado por las n variables se puede visualizar como un punto en un espacio de n dimensiones.



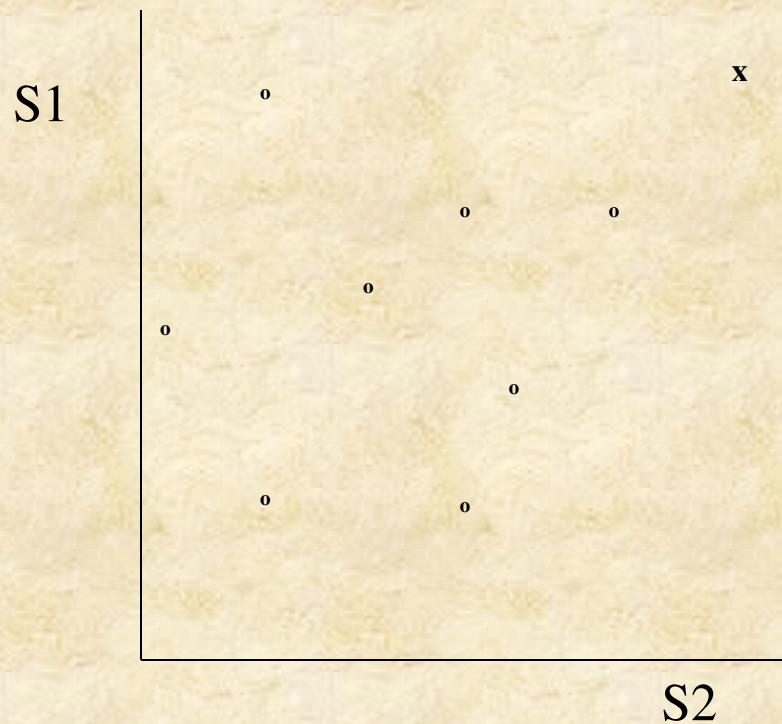
El objeto a se representa con dos variable independientes, por lo que se puede visualizar como un punto en 2 dimensiones



El objeto b se representa con tres variable independientes, por lo que se puede visualizar como un punto en 3 dimensiones

Si un objeto c se representa con cuatro variable independientes, entonces se puede visualizar como un punto en 4 dimensiones, y en general un objeto n que se representa con n variable independientes se puede visualizar como un punto en n dimensiones

O sea que cada vector de la matriz evolutiva representa un punto en un espacio multidimensional y todas las reglas forman una nube de puntos. De donde, por ejemplo, el proceso de encontrar el diagnostico asociado a un vector de síntomas X equivale a encontrar el punto que tiene distancia mínima con este vector.

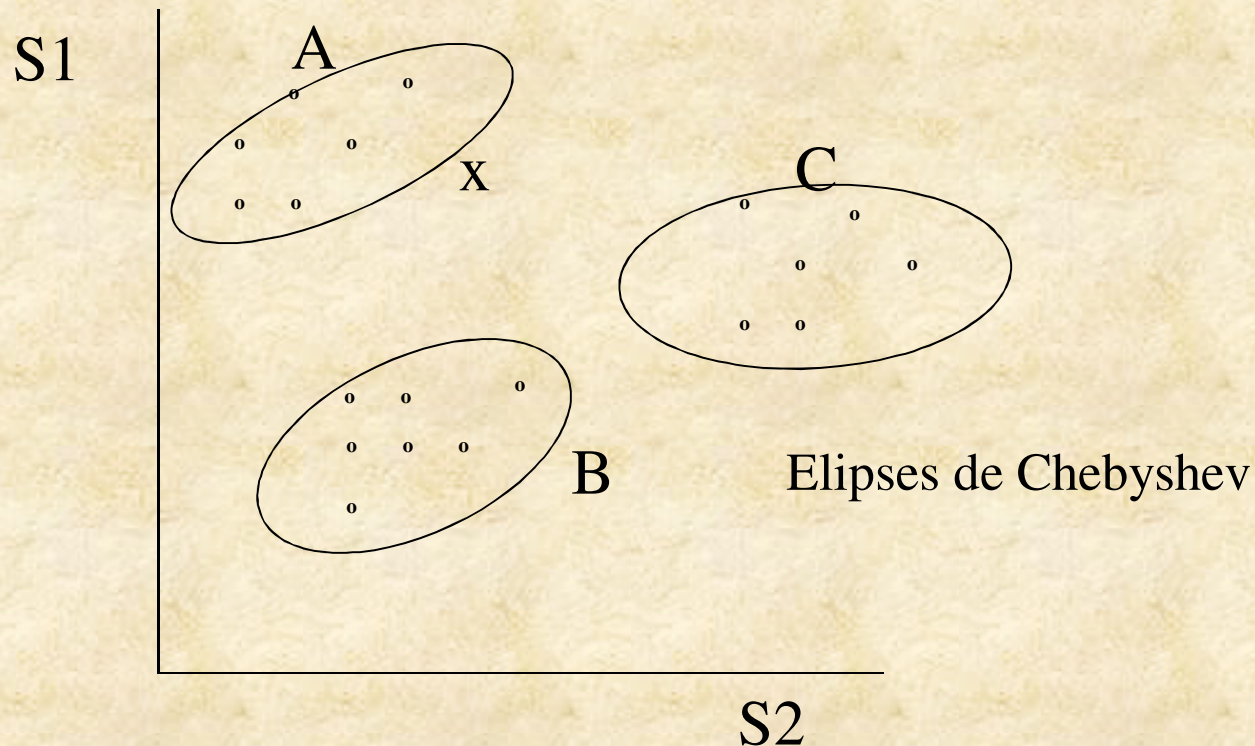


Nota: por facilidad visual y de explicación manejaremos espacios con 2 variables, pero la idea se aplica directamente a espacios de n variables

Espacios N-Dimensionales Evolutivos

Análisis de Cúmulos

básicamente toma una muestra significativa de objetos similares, los representa como puntos en un espacio n-dimensional, encuentra la media y la varianza de estos puntos y cuando llega un objeto nuevo al sistema, mide la distancia entre el punto nuevo y el punto medio, si esta es pequeña lo reconoce como un objeto similar y si no lo desecha.



es común que cuando se aplica esta técnica, los patrones ya están prefijados, ya sea porque explícitamente fueron asignados o porque se obtuvieron mediante un proceso de aprendizaje y quedaron fijos e inmutables.

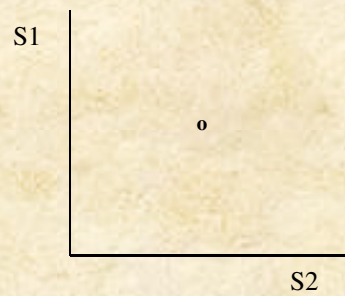
Ahora bien, en el caso de las matrices evolutivas también cada imagen, regla o cadena de síntomas promedio se representa como un punto en un espacio de n dimensiones, o sea que, todas las reglas forman una nube de puntos, pero con la característica de que *la matriz evolutiva permanentemente se esta modificando*

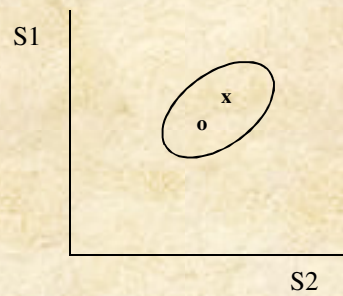
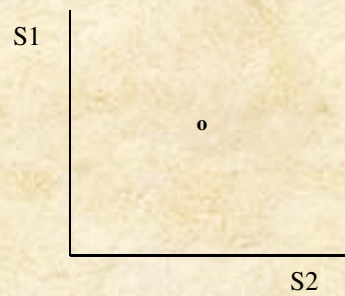
Originalmente la matriz evolutiva esta vacía, no tiene ni renglones ni columnas y representa un espacio que también lo esta,

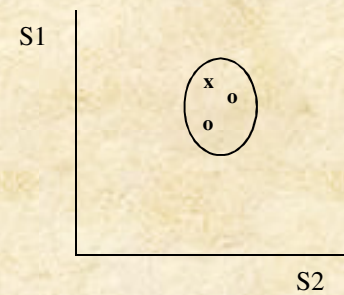
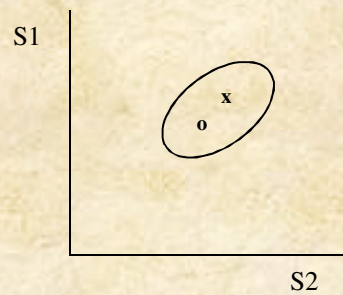
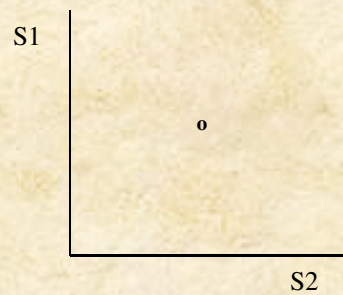
más adelante cuando llegan las primeras reglas o imágenes surgen los primeros puntos, renglones y columnas

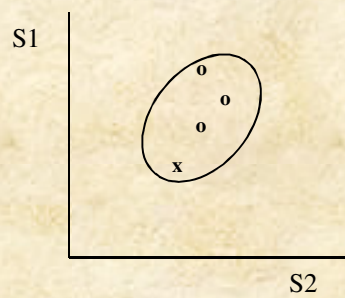
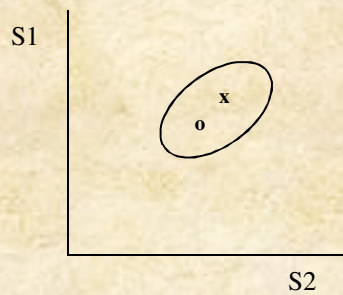
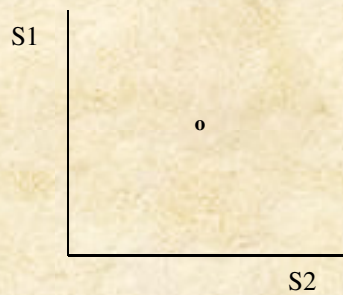
*pero estos no están fijos,
cuando una regla de la matriz evolutiva se modifica
el punto que la representa también cambia de
posición,*

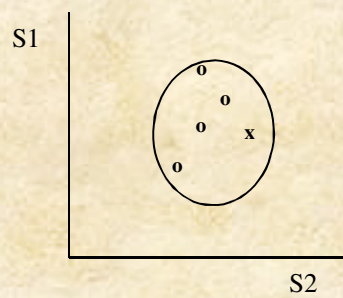
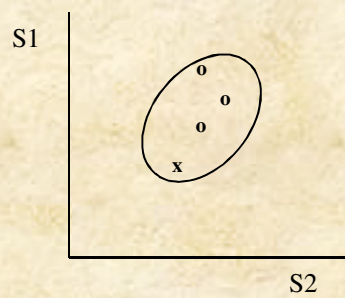
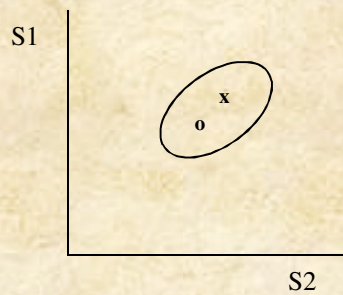
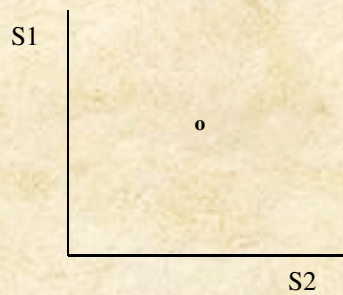
*o sea que,
en forma natural y permanente el espacio n-
dimensional se esta afinando y evolucionando.*

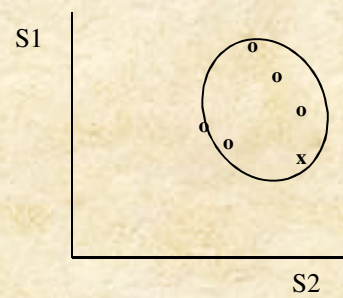
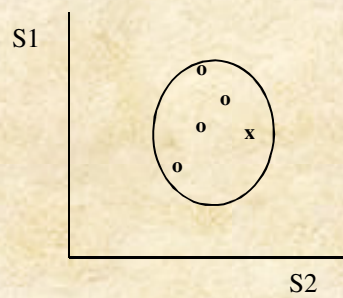
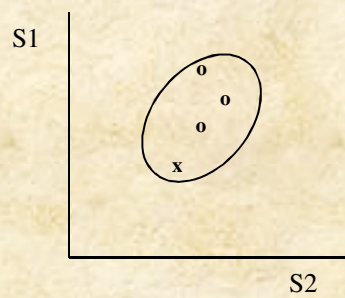
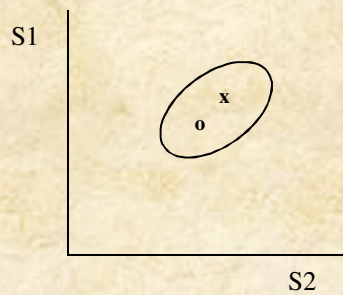
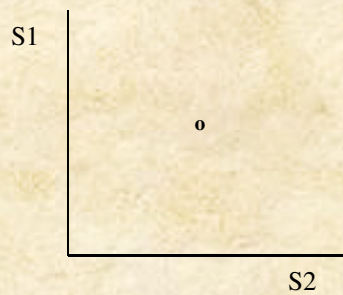




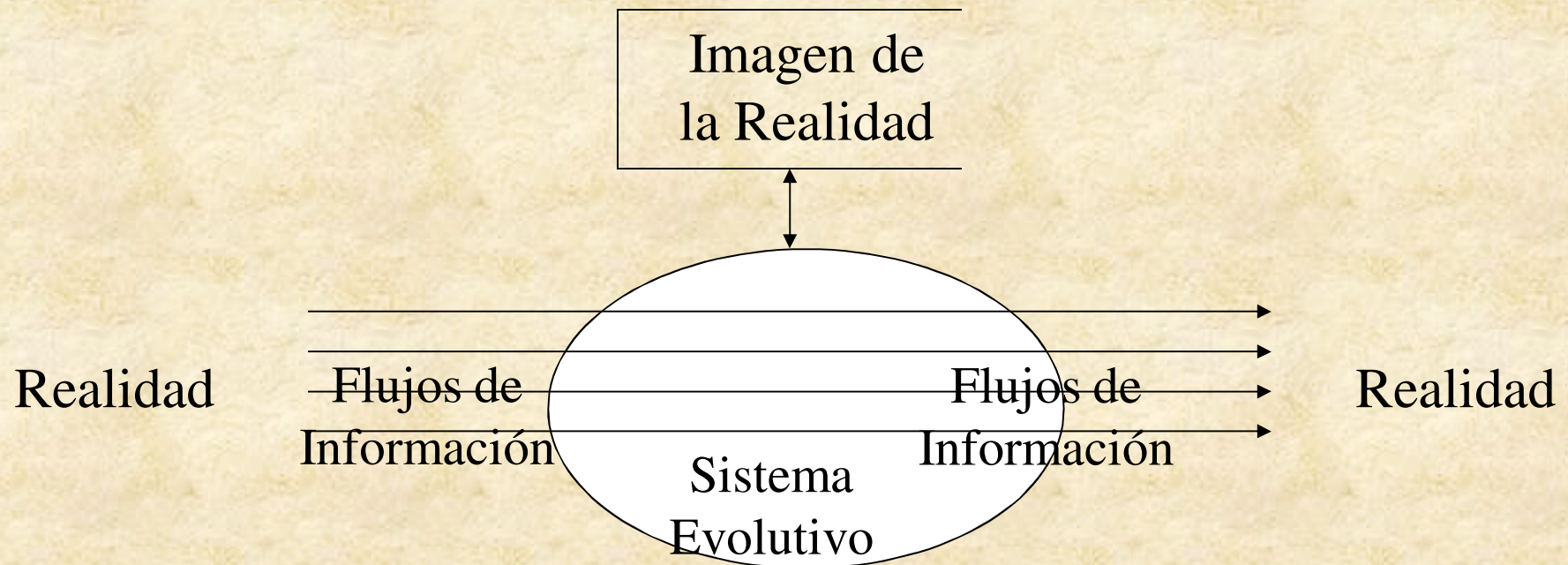








Con lo que el sistema evolutivo está transformando permanentemente su imagen de la realidad y no existe un proceso previo de aprendizaje y otro de aplicación, sino que por el proceso natural de fluir la información, el sistema evoluciona.



Por los flujos de información el Sistema Evolutivo se transforma

Sistema Evolutivo 1993

Los sistemas evolutivos y las matrices evolutivas en particular evolucionan a partir de los flujo de materia, energía e información que los cruzan,

en el caso mas general se asume que originalmente existe una Matriz Evolutiva 0 o Mev0 que no tiene dimensiones, renglones, columnas, capas, o valores en la matriz

conforme el sistema empieza a interactuar con el universo, la matriz empieza a evolucionar y dependiendo de cada caso a generar nuevos elementos, renglones, columnas, capas, dimensiones, y propiedades.

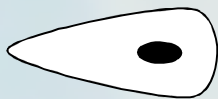
las matrices evolutivas son ejemplo claro de la dinámica dimensional,
permanentemente están cambiando sus dimensiones,
este es un ejemplo de la fuerza de los sistemas evolutivos y de las matrices evolutivas en particular,
ya que en los sistemas tradicionales, no se pueden aumentar ni siquiera el número de renglones o columnas y en las Mev en ciertos casos es indispensable poder modificar las columnas renglones, capas, dimensiones

Evolución de la realidad

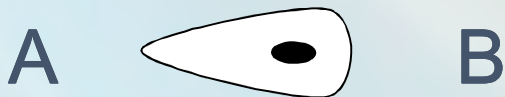
Percepción

¿Cómo los organismos o sistemas perciben, construyen su imagen de la realidad, se transforman, transforman su entorno?

Operador percibe

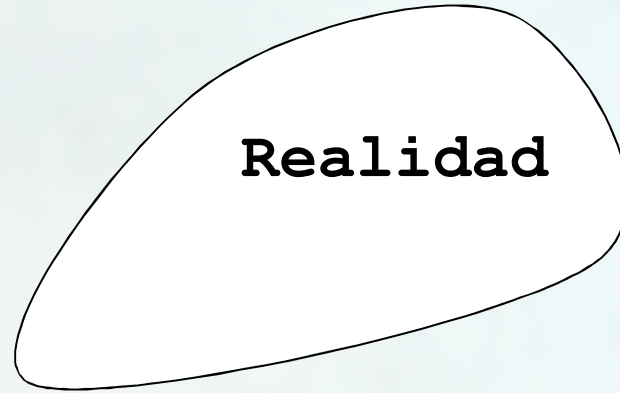
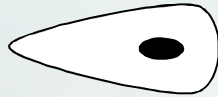


A percibe a B



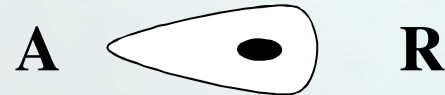


A

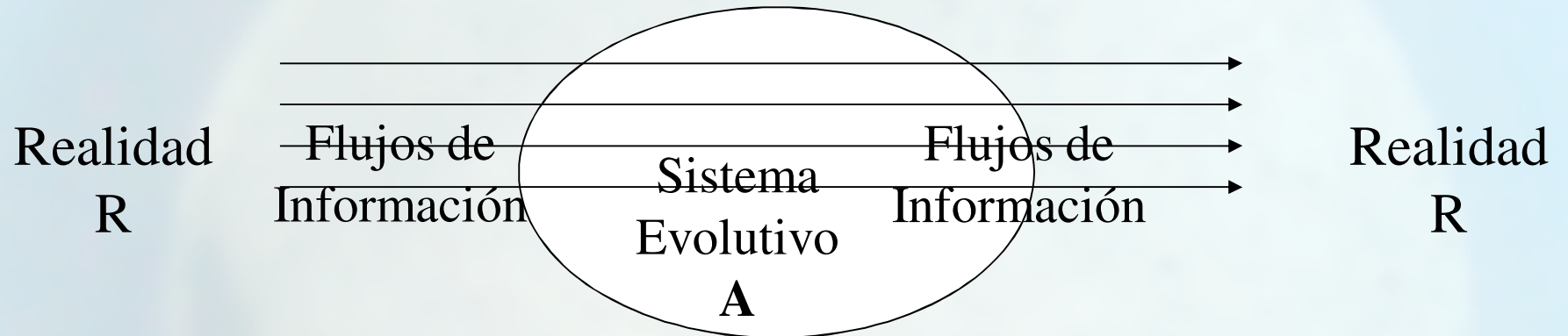


R

A percibe la realidad **R**,



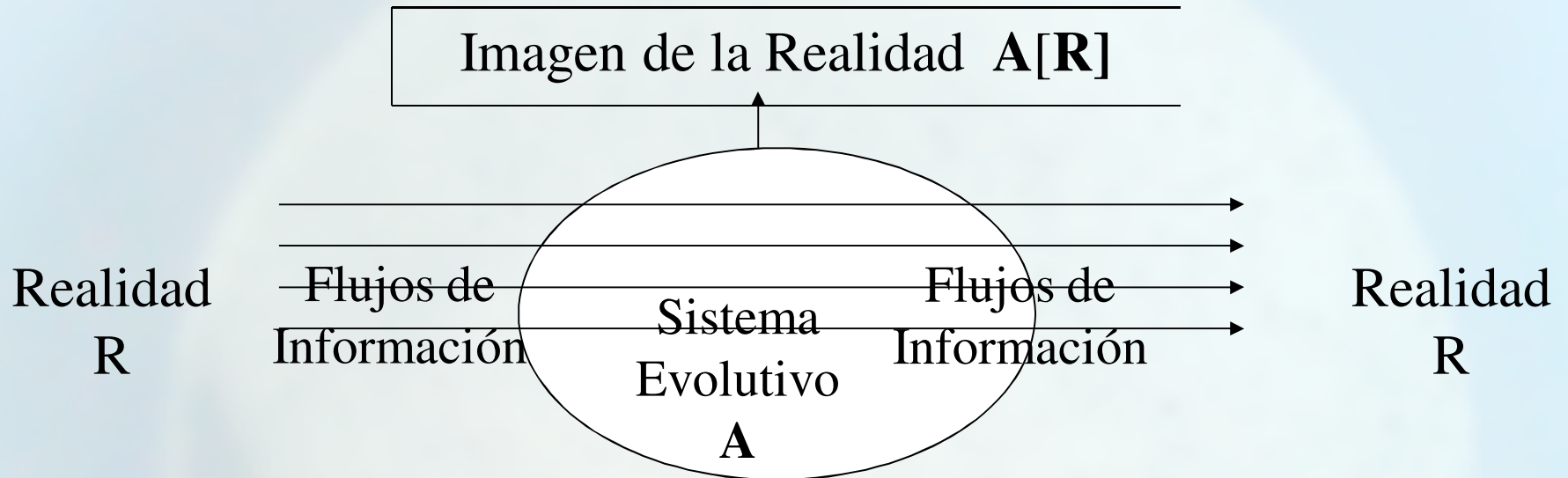
Evolución de la imagen de la realidad



El sistema A permanentemente esta percibiendo la realidad R

A  **R**

Evolución de la imagen de la realidad



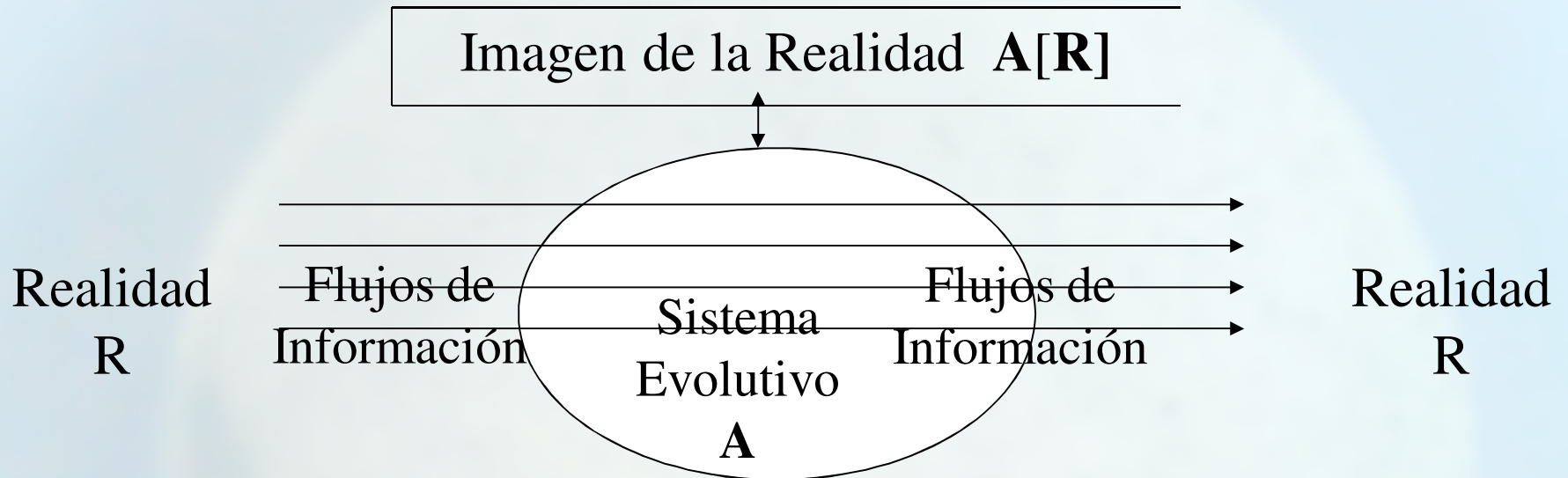
El sistema A permanentemente esta percibiendo la realidad R

$A \text{ } \left(\text{◀} \bullet \right) R$

A construye su imagen de la realidad R , $A[R]$

$(A \text{ } \left(\text{◀} \bullet \right) R) \rightarrow A[R]$

Evolución de la imagen de la realidad



El sistema A permanentemente esta percibiendo la realidad R

$A \triangleleft R$

A construye su imagen de la realidad R, $A[R]$

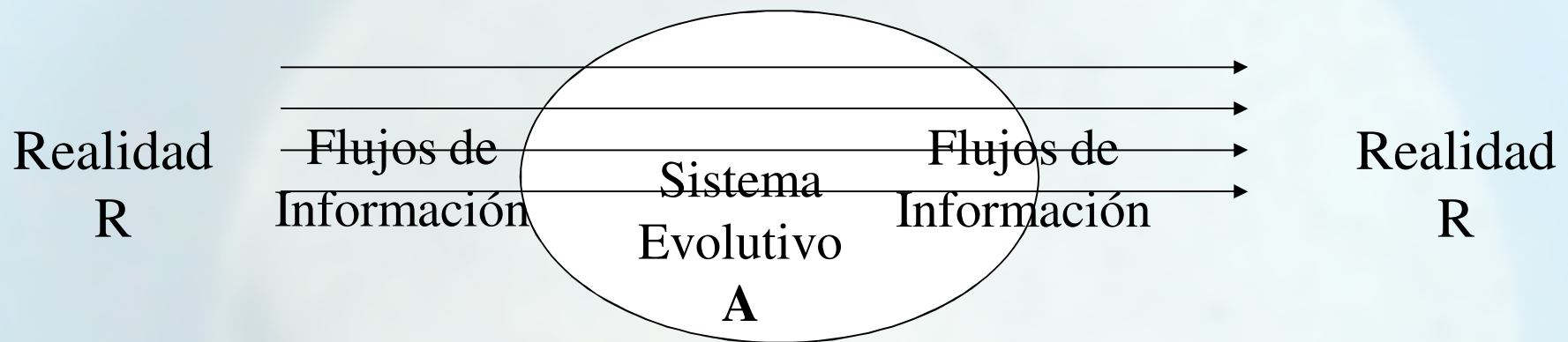
$(A \triangleleft R) \rightarrow A[R]$

La imagen de la realidad $A[R]$ permanentemente esta evolucionando

$A[R] \oplus (A \triangleleft R) \rightarrow A[R]$

Evolución del sistema

Por los flujos de información el sistema evoluciona.



El sistema **A**
permanentemente esta percibiendo la realidad **R**

A  **R**

y permanentemente esta evolucionando

$$\boxed{\mathbf{A} \oplus (\mathbf{A} \text{  \mathbf{R}) \rightarrow \mathbf{A}}$$

La Realidad R permanentemente esta evolucionando

A es parte de la Realidad R

la realidad R

permanentemente esta percibiendo a el sistema A

R  A

La Realidad R integra una imagen de A **R[A]**

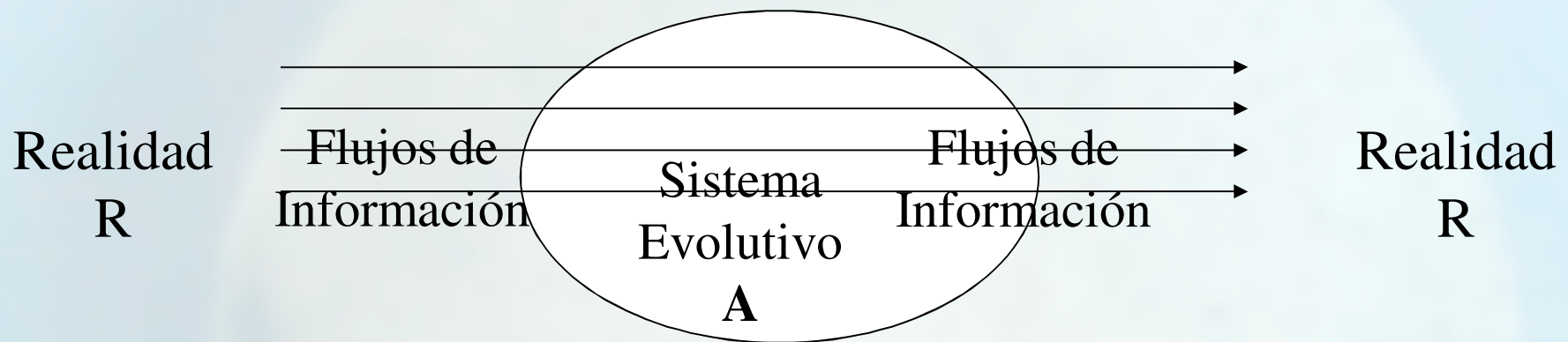
A modifica a R

R permanentemente esta evolucionando

$$\mathbf{R} \oplus (\mathbf{R} \text{  \mathbf{A}) \rightarrow \mathbf{R}$$

Evolución del sistema

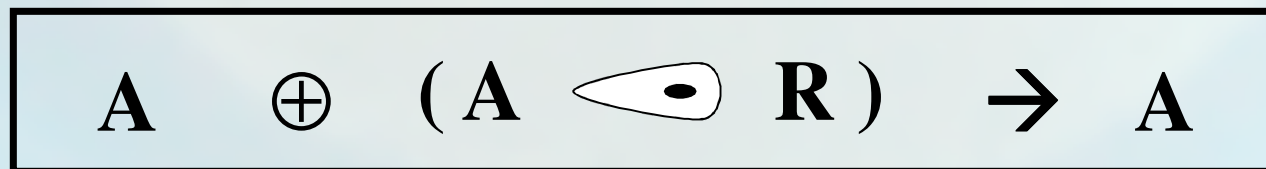
Por los flujos de información el sistema evoluciona.



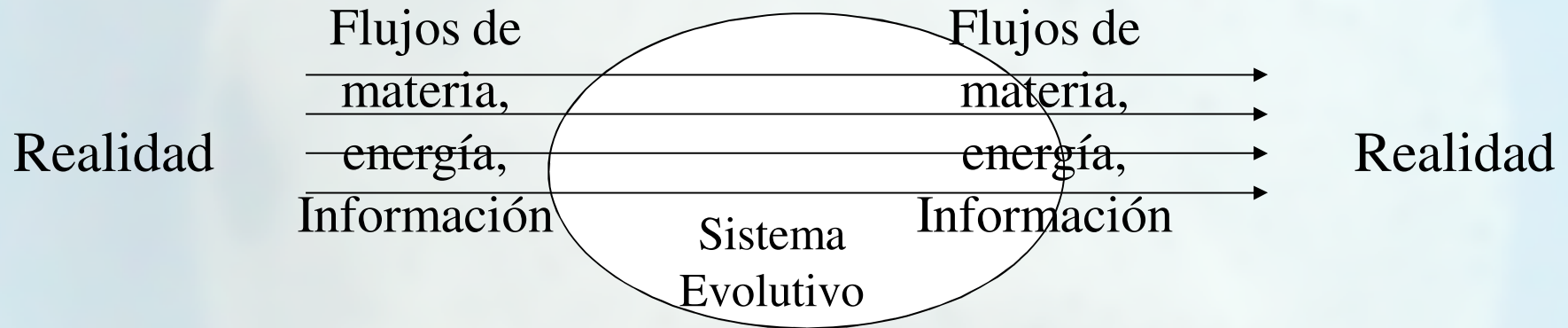
El sistema **A**
permanentemente esta percibiendo la realidad **R**

A  **R**

y permanentemente esta evolucionando



Por el hecho de que fluyen la materia, energía, información o algunos otros factores esenciales se da la evolución.



Por los flujos de materia, energía, información el
Sistema evoluciona

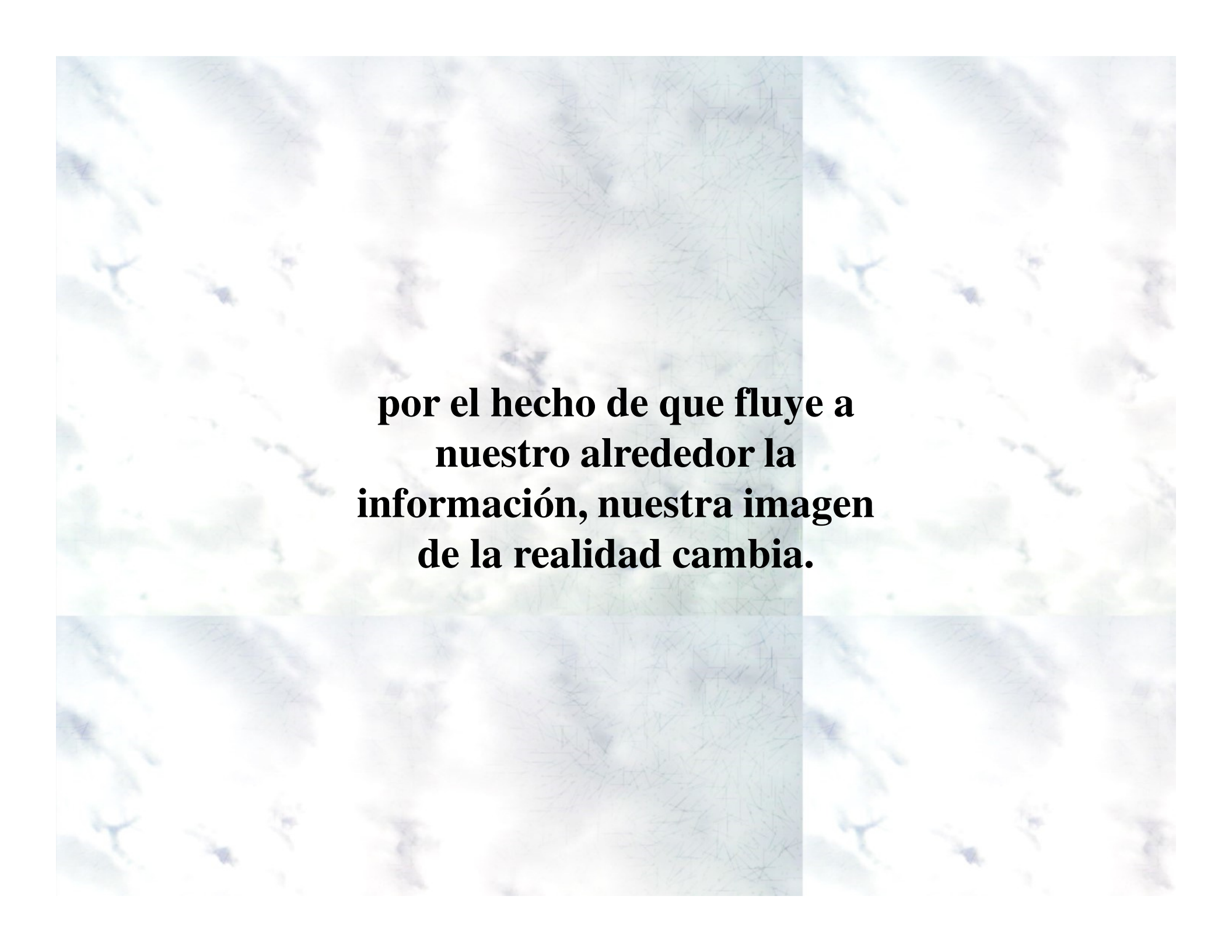
Evolución 1995

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

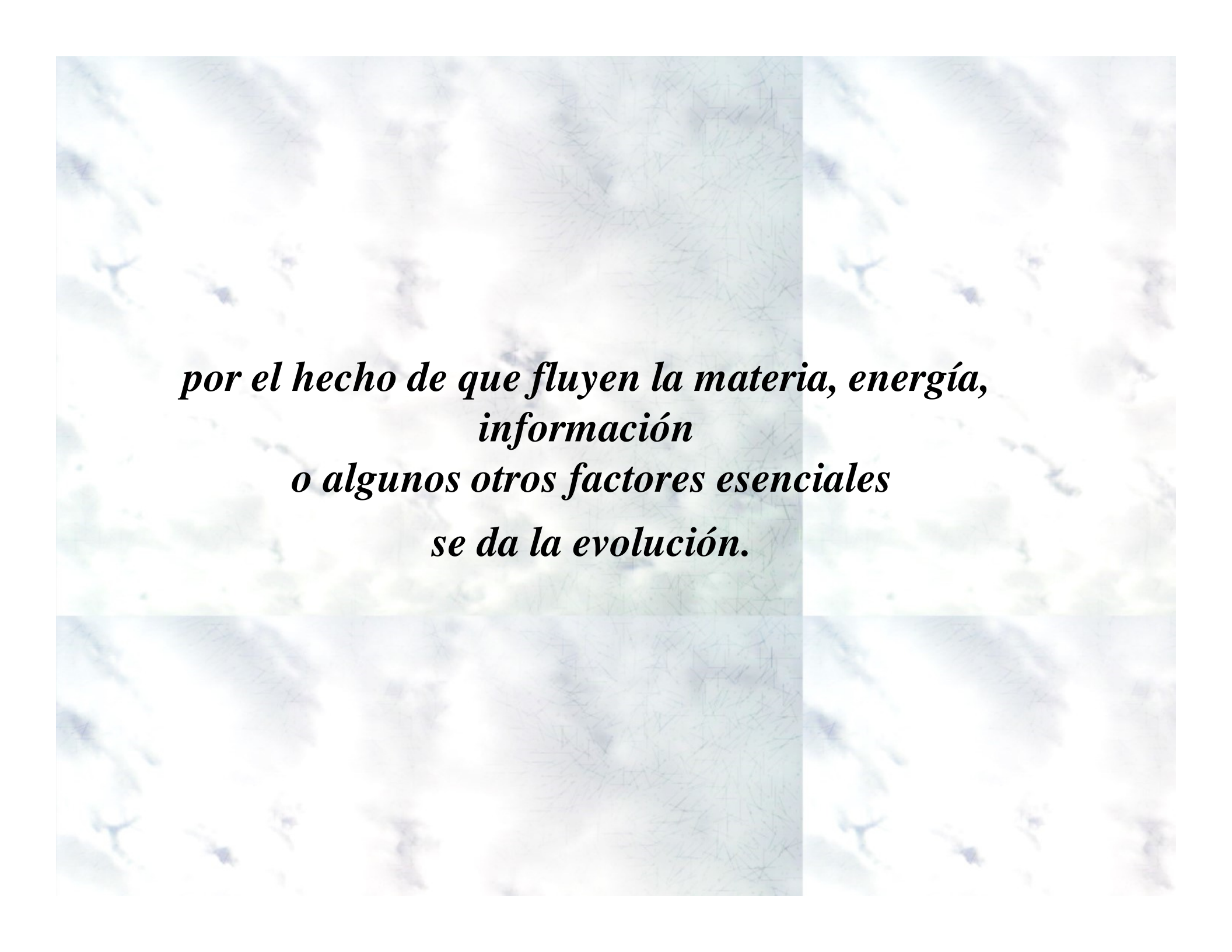
EVOLUCIÓN

Evolución 1995

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



**por el hecho de que fluye a
nuestro alrededor la
información, nuestra imagen
de la realidad cambia.**



*por el hecho de que fluyen la materia, energía,
información
o algunos otros factores esenciales
se da la evolución.*

La Evolución es un fenómeno universal e intrínseco a la naturaleza se presenta como resultado de la interacción de un organismo con su medio, con otros organismos, consigo mismo o cuando diferentes organismos y sociedades interactúan mutuamente en procesos de coevolución, dando como resultado que la organización o sistema (natural, artificial, social, etc.) se transforme y en su caso modifique el ambiente.

*el cambio se da como resultado de la
interacción fractal que se da entre los
componentes de un sistema, entre los
sistemas
y entre todo esto y el enorme caldo de
materia, energía e información
donde se encuentra inmerso
ya que al haber interacción fluye algo
entre los que interactúan y esto
propicia la evolución.*

**la evolución,
el crecimiento,
la vida,
el aprendizaje,
el pensamiento,**

***la transformación de nuestra imagen de la realidad,
los procesos de descomposición,
el desarrollo y transformación de las empresas, sociedades,
organizaciones, países, galaxias y universos,
etc.***

**son manifestaciones de un mismo proceso general de
transformación o cambio**

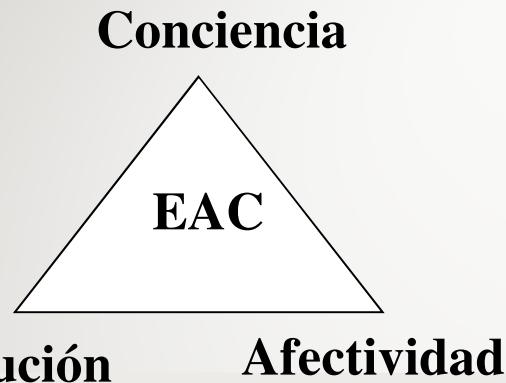
**Al que por facilidad denominamos
*Evolución***

El hecho de que un sistema evolucione no significa que sea "mejor" en un sentido absoluto (mas grande, mas rico, mas humano, mas inteligente, etc.)

sino que se adapta mejor a su ambiente y aumenta sus posibilidades de supervivencia.

Un sistema evolutivo tiende a mejorar su interrelación con el medio y en general es “hermoso” y natural dentro de su contexto.

Con este trabajo espero dar una idea de la magnitud del área y propiciar que se puede empezar a plantear el surgimiento de una
Ciencia o Disciplina de la Evolución, Interacción o Cambio,
donde se pueda *estudiar la Evolución en sus diferentes manifestaciones y encontrar sus características (fundamentos, reglas, leyes, patrones, etc.) generales e independientes de las manifestaciones particulares en que se presenten.*



EVOLUCIÓN Y SISTEMAS EVOLUTIVOS

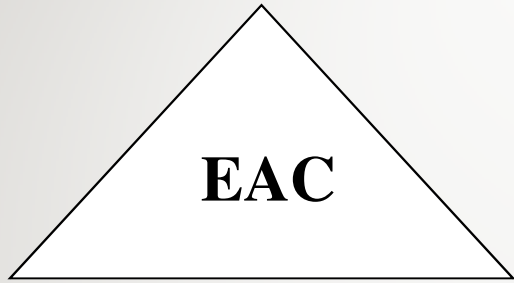
<http://www.fgalindosoria.com/eac/>

www.fgalindosoria.com/eac/evolucion/

La evolución, el crecimiento, el aprendizaje, la vida, la transformación de nuestra imagen de la realidad, el pensamiento, los procesos de descomposición, el desarrollo y transformación de las empresas, sociedades, organizaciones, países, galaxias, universos, etc., son manifestaciones de un mismo proceso general de transformación o cambio al que por facilidad llamamos evolución.

21 de Septiembre del 2002, Octubre del 2009, Julio del 2012, Febrero del 2017

Consciencia



Evolución

Afectividad

Seminario de Investigación Sistemas Evolutivos, Afectivos y Conscientes

www.fgalindosoria.com/seac2017/seac.pdf

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

Fernando Galindo Soria

fgalindo@ipn.mx

www.fgalindosoria.com

Paola Neri Ortiz

paolaneriortiz@yahoo.com

www.eventbrite.com/o/paola-neri-ortiz-12877179662

Auditorio Sor Juana (edificio C)

Tecnológico de Estudios Superiores de Ecatepec (TESE)

Del Lunes 13 al Jueves 16 de Marzo

de 2017 de 12:00 a 17:00 horas de la Ciudad de México

<http://www.fgalindosoria.com/seac2017/>

Algunas ligas las encuentra en
www.fgalindosoria.com/eac/evolucion/

Manejador Léxico (Enfoque Lingüístico)

El Manejador Léxico es un programa que recibe como entrada una oración en algún lenguaje y obtiene las unidades léxicas presentes en la oración, el tipo de cada unidad léxica y la estructura general u oración canónica.

Por ejemplo, en la oración

Calcula la Regresión de X

se encuentran las Unidades Léxicas mostradas en la tabla.

Calcula	A1	acción
la	i	ignora
Regresión	a2	acción
de	i	ignora
X	d1	dato

Calcula la Regresión de X
a1 i a2 i d 1

A

a1 i a2 i d 1

se les conoce como oración canónica

en general cuando se sustituyen los elementos de una oración por una representación que permita visualizar la estructura de la oración se obtiene una oración canónica.

Originalmente el Manejador Léxico no tiene ningún conocimiento acerca del lenguaje a utilizar y solamente cuenta con la capacidad para encontrar los diferentes tipos de unidades léxicas de un lenguaje.

cotidianamente manejamos una palabra como una cadena de caracteres (letras, números, signos) separados por espacios, por ejemplo en la oración “el perro ladra” tenemos tres palabras: “el”, “perro”, “ladra”,

el problema es que estamos manejando elementos semánticos y los elementos semánticos no necesariamente son cadenas separadas por blanco,

por ejemplo “Buenos Aires” es un elemento semántico que se refiere a una ciudad, “buenos” es otro elemento y “aires” es otro,

en el manejador es conveniente tratar de manejar elementos semánticos y no solo palabras,

a la unidad con significado semántico propio se le llama unidad léxica

la unidad léxica corresponde en muchos casos a una palabra, pero existen otros muchos casos en los cuales eso no es cierto

La operación general que permite encontrar los componentes de un objeto o los elementos de un sistema, como por ejemplo las palabras de una oración, los objetos de una imagen, las empresas que forman parte de un corporativo, etc. se conoce como

Desconcatenación

y consiste en encontrar y separar, los elementos de la realidad, universo, espacio, campo, sistema, organización, objeto, organismo, bicho, o ente que se este estudiando.

La desconcatenación esta relacionada con el análisis de sistemas, la lingüística matemática y con las operaciones de búsqueda.

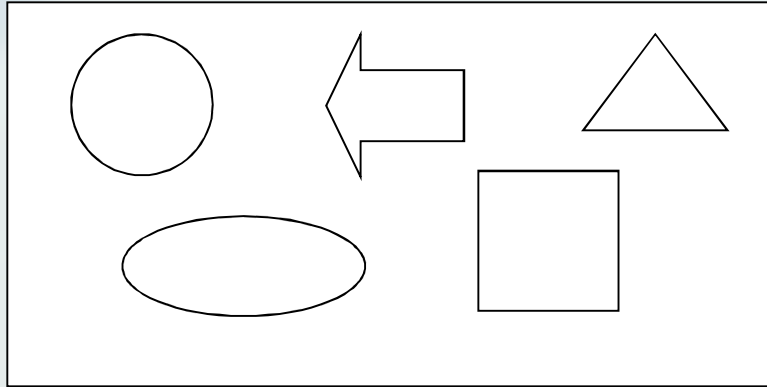
La desconcatenación es una operación que básicamente busca los elementos que componen un sistema u organización, por ejemplo si se tiene el texto

el perro mordió al gato

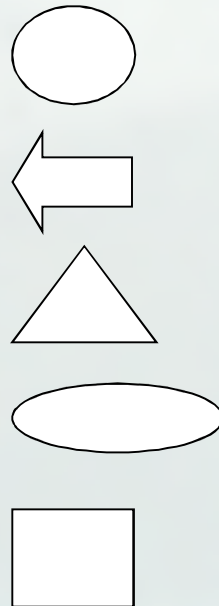
la desconcatenación encontraría que esa oración esta formada por las palabras

el
perro
mordió
al
gato

O por ejemplo si se tiene la Imagen



El desconcatenador encontraría los siguientes elementos:



Partimos de que en un universo dado existen elementos básicos e "indivisible" tipo átomos o cuantos, por ejemplo en el caso de las imágenes podrían ser los píxeles de la imagen, en el caso del código genético las proteínas, en el caso del texto de un documento los caracteres, en el caso de una sociedad, las personas que la forman, etc.

Esos elementos básicos se combinan para conformar unidades semánticas mínimas, que son los elementos funcionales de ese universo, conocidas como unidades léxicas o sea unidades con significado mínimo en ese universo; dependiendo del problema estas unidades pueden ser átomos o moléculas, genes, objetos en una imagen, cadenas de palabras con significado propio, componentes funcionales del cuerpo, como por ejemplo brazos, piernas, empresas dentro de un corporativo, etc.

Desconcatenación mediante búsqueda de cadenas.

La búsqueda de cadenas consiste en tomar una cadena grande, por ejemplo

mgasdfglsmgasdfghjkasmhjd fg

y buscar dentro de ella cadenas que se repiten.

Existen muchas formas de encontrar cadenas dentro de cadenas, las dos variantes mas generales del método son:

Tomar un objeto o sistema y buscar la cadena mas grande que se repita en él, y sustituirla por una etiqueta X1,

tomar la siguiente cadena que se repita y sustituirla por una cadena X2
y así sucesivamente mientras existan cadenas,

Otro método parte al revés,

se buscan las cadenas mas pequeñas, mayores que uno y a todas las cadenas iguales se les asigna la misma etiqueta, luego si se encuentra una combinación de etiquetas repetida varias veces

por ejemplo la cadena de etiquetas X8X3X14 aparece repetida varias veces, se supone que toda esa combinación también es por si sola una unidad léxica y se le asigna su propia etiqueta.

Tomar un objeto o sistema y buscar la cadena mas grande que se repita en él, y sustituirla por una etiqueta X1,
tomar la siguiente cadena que se repita y sustituirla por una cadena X2
y así sucesivamente mientras existan cadenas que se repitan

Por ejemplo en `mgasdfglsmgasdfghjkasmhjdffg`
`mgasdfg` es la cadena mas grande que se repite
`mgasdfglsmgasdfghjkasmhjdffg`

Por lo que se marca como una nueva cadena

`O` → `X1lsX1hjkasmhjdffg`

`X1` → `mgasdfg`

Si se sigue analizando las cadenas se encuentra que la cadena `dfg` es la siguiente cadena mas grande que se repite

`O` → `X1lsX1hjkasmhjdfg`

`X1` → `mgasdfg`

Por lo que se genera la siguiente regla

`O` → `X1lsX1hjkasmhjX2`

`X1` → `mgasX2`

`X2` → `dfg`

Otro método parte al revés,
se buscan las cadenas mas pequeñas, mayores que uno y a todas las cadenas iguales se les asigna la misma etiqueta, luego si se encuentra una combinación de etiquetas repetida varias veces
por ejemplo la cadena de etiquetas X8X3X14 aparece repetida varias veces, se supone que toda esa combinación también es por si sola una unidad léxica y se le asigna su propia etiqueta.

Por ejemplo **mgasdfglsmgasdfghjkasmhjdffg**

Tiene varias cadenas pequeñas

mgasdfglsmgasdfghjkasmhjdffg

Por lo que sustituyendo queda:

O → **X1 X2 X3** g l s **X1 X2 X3** g **X4** k **X2** m **X4 X3** g

X1 → **mg**

X2 → **as**

X3 → **df**

X4 → **hj**

Si seguimos analizando, se detecta que existe cadenas que se repiten como **X1 X2** o **X3 g**
por lo que se sigue sustituyendo quedando

O → **Y1 Y2** l s **Y1 Y2 X4** k **X2** m **X4 X3** g

X1 → **mg**

X2 → **as**

X3 → **df**

X4 → **hj**

Y1 → **X1 X2**

Y2 → **X3 g**

Aun mas la cadena **Y1 Y2** también se repite
por lo que introducimos la regla

Z1 → **Y1 Y2** Y la cadena queda

O → **Z1** l s **Z1 X4** k **X2** m **X4 X3** g

**en Ciencia, Ingeniería, Filosofía, Informática,
Lingüística, Computación, etc.**

se llevan a cabo procesos donde se buscan los

Elementos,

Relaciones (Estructuras),

Significados

de los campos de problemas del área

El concepto de sistema

Teoría de Sistemas

Karl Ludwig von Bertalanffy

Un sistema es un conjunto de elementos
relacionados entre si

Elementos
y estructuras

inteligencia

Para Jean Piaget, la inteligencia tiene dos atributos principales: la *organización* y la *adaptación*.

El primer atributo, la organización, se refiere a que la inteligencia está formada por estructuras o esquemas de conocimiento, cada una de las cuales conduce a conductas diferentes en situaciones específicas.

.....

La segunda característica de la inteligencia es la adaptación, que consta de dos procesos simultáneos: la asimilación y la acomodación

http://es.wikipedia.org/wiki/Teor%C3%ADa_Constructivista_del_Aprendizaje

asimilacion ingreso de nuevos **elementos** a los esquemas mentales preexistentes,.

acomodación modifica esquemas (**estructuras cognitivas**)

Asimilación (psicología)

La **asimilación** (del Lat. *ad* = hacia + *similis* = semejante) es un concepto **psicológico** introducido por **Jean Piaget** para explicar el modo por el cual las personas ingresan nuevos **elementos** a sus esquemas mentales preexistentes, explicando el crecimiento o cambio cuantitativos de éste. en este caso no existe modificación en el esquema sino sólo la adición de nuevos elementos.

[http://es.wikipedia.org/wiki/Asimilaci%C3%B3n_\(psicolog%C3%ADa\)](http://es.wikipedia.org/wiki/Asimilaci%C3%B3n_(psicolog%C3%ADa))

Acomodación

La **acomodación** o **ajuste** es un concepto **psicológico** introducido por **Jean Piaget**. Por acomodación se entiende el proceso mediante el cual el sujeto modifica sus **esquemas (estructuras cognitivas)** para poder incorporar a esa estructura cognoscitiva nuevos objetos. Esto puede lograrse a partir de la creación de un nuevo esquema, o la modificación de un esquema ya existente para que el nuevo estímulo pueda ingresar en él. Por esta razón suele considerarse este mecanismo como un cambio cualitativo en el esquema. A partir de estos procedimientos, que Piaget denomina **funciones cognitivas** se establece el proceso de adaptación y equilibrio cognitivo entre esquema y medio del organismo.

<http://es.wikipedia.org/wiki/Acomodaci%C3%B3n>

Lingüística Matemática

Noam Chomsky

Las oraciones están formadas por:

Unidades Léxicas (Palabras, elementos)

Sintaxis (estructura)

Semántica (Relaciones entre diferentes cosas)

Teoría de Sistemas

Karl Ludwig von Bertalanffy

Un sistema es un conjunto de elementos relacionados entre si

Elementos y estructuras

la **inteligencia**, Jean Piaget

la **adaptación**, consta de dos procesos simultáneos: la asimilación y la acomodación

Jean Piaget

asimilación ingreso de nuevos **elementos** a los esquemas mentales preexistentes,.

acomodación modifica **esquemas (estructuras cognitivas)**

Lingüística Matemática

Noam Chomsky

Léxico (Palabras, elementos)

Sintaxis (estructura)

Semántica (Relaciones)

**Ejemplos de herramientas para encontrar los
elementos, entidades, partes,
análisis, desconcatenación, análisis léxico,
teselación, etc.**

**Ejemplos de herramientas para encontrar
relaciones, estructuras, sintaxis
análisis sintáctico**

**Ejemplos de herramientas para encontrar
significado
análisis de contenido, exegesis, semántica**

un Sistema Evolutivo debe tener la capacidad de encontrar los componentes del sistema de información a partir del lenguaje utilizado en el área problema

Un sistema lingüístico tiene involucrados al menos

*Datos
elementos o componentes léxicos (unidades léxicas)*

*Procesos
componentes semánticos (significado),*

*estructura
sintaxis*

por lo que a continuación veremos en general como encontrar e integrar esos componentes a partir del, lenguaje utilizado en el área problema