

Juan Bautista Chavarría-Chaves

# Incursión de Claudio Gutiérrez en informática, bases de datos, modelaje y planificación universitaria, con especial referencia a los años 1971-1973: el proyecto SIMULA

**Resumen:** Aquí se describe una obra particular de Claudio Gutiérrez, visionario filósofo costarricense, cuyo accionar abarcó muchos campos, incluyendo el de planificación académica universitaria, al cual hizo importantes contribuciones. Una que él desarrolló con gran visión al principio de la década de los años 1970 fue el proyecto SIMULA. Primero se describe el modelo que elaboró bajo este proyecto y el tipo de resultados que permitió a las autoridades universitarias establecer políticas para la institución. Seguidamente se relatan las condiciones y limitantes de la época, en cuanto a capacidad informática y en cuanto a disponibilidad de datos.

**Palabras clave:** Planificación universitaria, modelos, simulación.

**Abstract:** This article is about a particular work of the late Claudio Gutiérrez, a visionary and well-known Costa Rican philosopher, with broad productive activity in many fields, including academic university planning, to which he made important contributions. The project which he termed SIMULA, developed in the early 1970's is described here. An overview of the computer model used, along with its main characteristics and results, useful in the process of political decision taking for a university, are presented first. Then, a general glimpse is

given about the limiting conditions in terms of computing and data availability of those years.

**Keywords:** University planning, computer models, simulation.

## Introducción

Claudio Gutiérrez fue un gran pensador y visionario, con una extensa labor académica. Su formación y accionar abarcó muchos campos: Historia, derecho, filosofía, economía, informática, inteligencia artificial, planificación universitaria. Su obra escrita comprende temas de lógica, epistemología, informática y sociedad, humanismo, así como ensayos sobre la realidad nacional. También escribió poesía, algunos cuentos y hasta logró adelantarse adivinando lo que vendría y escribir a tiempo su autobiografía, antes de la llegada de una larga e incapacitante enfermedad. Y de seguro esta lista es solo una pincelada de todo lo que hizo.

Poseedor de una inteligencia extraordinaria, no sólo podía abarcar los aspectos teóricos y conceptuales, sino que era capaz de adentrarse en los detalles y, por ejemplo, sentarse y escribir el código de un programa específico para una computadora. Pensar, hacer, iban de la mano con mucha naturalidad en él. Fue un excelente profesor universitario, según el testimonio de los



que tuvimos el privilegio de tomar algún curso con él. Los que además tuvimos la fortuna de trabajar algún tiempo a su lado, hallamos en él un verdadero maestro que nos guió y enseñó con mucha sabiduría, llevándonos incluso a realizar cambios importantes en nuestra formación universitaria y profesional, lo que convierte su recuerdo en cariño y aprecio. Fue el filósofo que enseñó programación de computadoras a dos jóvenes estudiantes que trabajaban con él, uno era exseminarista y enamorado de la lógica y el otro un aprendiz de Biología.

Este artículo pretende ser una pequeña ventana a una obra particular suya, que él desarrolló con gran visión al principio de la década de los años 1970: el proyecto SIMULA. Su trabajo involucró aspectos de informática, programación, bases de datos, modelaje, simulación y planificación, cuando fue asesor académico de la Oficina de Planificación Universitaria (OPLAU) de la Universidad de Costa Rica (UCR). Eso obligó a desarrollar diversas tareas que, para la UCR de esos años marcaron trayectorias importantes en cuanto a planificación, organización de datos y formas de gestión. Su contribución, en aquel momento, al desarrollo de esos campos en la UCR parece acrecentarse hoy día, con la perspectiva del tiempo, y dadas las condiciones y herramientas disponibles en aquel entonces.

Primero se describe el modelo SIMULA ideado por Claudio Gutiérrez, el cual buscaba hacer una representación simbólica de los aspectos más sobresalientes, desde el punto de vista académico, de la estructura y el funcionamiento de la UCR.

En cuanto a herramientas, un personaje muy importante en este recuento histórico fue Matilde, la primera computadora que tuvo la UCR y compañera inseparable en el proyecto durante esos años. Por eso aquí también se hace una descripción de cómo era ella, con sus virtudes y limitaciones. En esa época los recursos computacionales eran bastante limitados, como se intenta reflejar en este documento, pero Claudio Gutiérrez logró resultados sorprendentes.

Otro aspecto esencial de mencionar es el tipo de datos, su organización y disponibilidad en aquella época y el esfuerzo para poder convertirlos en información útil para la toma de

decisiones. Eso también se aborda aquí. Esos dos aspectos, el de la herramienta computacional y el de disponibilidad de datos, pueden darnos una perspectiva histórica de las condiciones en que se desarrolló el proyecto.

## El modelo SIMULA

Como se mencionó antes, Claudio Gutiérrez era por aquel entonces el asesor académico de la oficina de planificación universitaria de la universidad de Costa Rica, la cual había sido recientemente fundada.

Su proyecto estrella desembocó en el modelo que desarrolló a partir de la información de bancos de datos: estudiantes, profesores, cursos y planta física. Luego le incorporó datos financieros. El objetivo era dotar a las autoridades universitarias de alto nivel con información y herramientas útiles que contribuyera a establecer políticas para la institución.

Vio claro que las ciencias de la administración (planificación, en particular) debían pasar a jugar un papel mayor en la administración de instituciones de educación superior (Gutiérrez 1973). La herramienta principal para procesar los datos reunidos sería una computadora.

En la Universidad de Costa Rica existía una que fue adquirida en el año 1968 (Feoli 2018) y bautizada como Matilde. Pocas personas estaban capacitadas para usar esa computadora y Claudio Gutiérrez cuenta, en sus memorias (2010), que guiado por su curiosidad sobre la posibilidad de probar teoremas de lógica con una computadora, empezó a asistir como oyente al curso en que se enseñaba el lenguaje de computación FORTRAN, de orientación muy matemática e ingenieril. Aprendió así a programar y poco después, cuenta él jocosamente, ya estaba pasando sus primeras noches con Matilde, cuando lograba que se la prestaran después de las jornadas regulares. No solo aprendió uno de sus lenguajes, sino que también a operarla él mismo. Sus aventuras de lógica con FORTRAN y Matilde debieron ser un productivo entrenamiento, aunque llegase a descubrir que ese lenguaje no era el apropiado para ese tipo de aplicaciones de la lógica.

Poco después de su experiencia con FORTRAN, Claudio tuvo la oportunidad de aprender el lenguaje ensamblador de Matilde, gracias a unas tutorías que recibió con el Ing. Mario Feoli y ahí parece haberse convencido de que ese era el lenguaje apropiado que andaba buscando, por ser más cercano al corazón de Matilde (Mario Feoli 2023, comunicación personal). Al crearse OPLAU e iniciar sus labores como asesor académico, empezó a reclutar personas para su idea de desarrollar un programa de computadora para predecir el número de estudiantes, profesores y aulas que requería la UCR. Recuerdo que fue en marzo de 1971 cuando empecé como asistente suyo (a medio tiempo) a codificar datos de estudiantes entre anaqueles de la Oficina de Registro de la universidad. Más adelante, ese mismo año ya me estaba enseñando como programar y también luego reclutó otra persona, que fue alumno aventajado suyo en los cursos de lógica de la escuela de filosofía. Esta persona, entrenada por el mismo Claudio Gutiérrez, resultó ser muy hábil programador y fue quien escribió la mayor parte del código de SIMULA.

Fue en ese periodo 1971-1973 que logró desarrollar su proyecto que denominó «Un Simulador Integral del Movimiento de una Universidad Latino Americana» (SIMULA).

El modelo no pretendía generar políticas institucionales sino más bien simular resultados posibles ante diferentes políticas que se probaran, y de esa forma adelantar las probables consecuencias para la institución. Se pretendía averiguar qué pasaría si se tomaban ciertas decisiones respecto al número de nuevos alumnos, relación estudiantes-profesor, intensidad de uso de aulas, política de salarios, etc (Gutiérrez 1973). Se obtenían proyecciones hasta por 7 años.

Un breve recorrido por los componentes del modelo nos permite señalar los siguientes (Gutiérrez 1972):

a. El programa de computador (código) que procesa los datos y que brinda los resultados. Este programa está organizado en varios módulos o subprogramas cada uno ensamblado independientemente. Como estrategia se escribió código siguiendo la idea de programación por procesos pre-definidos (PPP), ideado expresamente por

Claudio Gutiérrez y codificado por el hábil programador mencionado antes, en el lenguaje ensamblador de Matilde.

- b. Ciertas constantes requeridas para el proceso. Por ejemplo, coeficientes de mantenimiento de edificios y reposición de equipo, las cuales se incorporaban al programa de computador.
- c. Matrices de coeficientes específicos para la sede central y sedes regionales. Estas matrices de coeficiente se ensamblaban anualmente, a partir de los datos del último año. Un detalle curioso de destacar es la forma en que se invocaba esto, por ejemplo, se le hacía peticiones al computador de la siguiente manera (todo en mayúscula):

#### UNIPLAN SIMULE FUTURO UCR SEGÚN MATRICES SEDE CENTRAL

- d. Los parámetros para las ejecuciones del programa (corridas). Eran de varios tipos:
  1. El año histórico base y el año guía de proyección. Por ejemplo, la sede central tenía 1965 como año base, que fue cuando se inició la recolecta de datos. La sede San Ramón tenía el año 1968, la de Turrialba 1971 y Liberia 1972.
  2. Política de admisión que indicaba el número estudiantes admitidos o por admitir el primer año para cada uno de los cursos lectivos.
  3. Índices de costos, referidos a cambios en los salarios y en los costos de construcción.
  4. Políticas según unidad académica para la relación profesor crédito y para la política de aumento diferencial de sueldos de un profesor de tiempo completo.

Los puntos a, b anteriores venían a representar la parte que podemos llamar fija del modelo; el punto c representaba la historia de la institución, mientras que el punto d era la parte que se quería proyectar o simular, es decir el futuro de la institución.

En cuanto al punto c, es de interés destacar que, por ejemplo, en el año 1973, condensaba la historia de la UCR durante el periodo 1963 a 1971. Eso era así debido al rezago que se producía porque la oficina de registro requería de

un cierto tiempo para registrar y oficializar los resultados de los estudiantes y la OPLAU requería otro tiempo para codificarlos. En cuanto a profesores y aulas, la historia comprendía los años 1970 a 1972 dado que no existían registros previos. Algo similar ocurriría con lo de planta física y uso de aulas.

## Verificación del modelo de simulación

El modelo fue sometido a pruebas de validez, sobre todo validez en la historia, aplicándolo a años anteriores y contrastando con los resultados ya conocidos. También se hizo pruebas de validez en lo más reciente (por ejemplo, para el año que estaba en curso), con información que no estaba completa o registrada en los archivos pero que era suficiente o que se podía consultar puntualmente en las dependencias encargadas

de la universidad. Se aplicaron además pruebas de agregación y desagregación para detectar si los posibles sesgos estadísticos mantenían su congruencia al desagregar los datos. Las pruebas anteriores permitieron hacer algunas correcciones y ajustes para mejorar los cálculos que proporcionaba el modelo. En cuanto a los niveles de error se procuró mantenerlos entre 5 y 10% (Gutiérrez 1973).

## Proyecciones del modelo SIMULA y su uso

Una ejecución o «corrida» del modelo producía un informe para cada unidad académica y un total para todo el campus. Lo esencial de los resultados ofrecidos por el modelo pueden verse en esta ejecución para la sede regional de San Ramón. (Gutiérrez 1973, apéndice B).

MUESTRA DE RESULTADOS DEL PROGRAMA PRINCIPAL							
UNIPLAN SIMULA FUTURO U.C.R. SEGUN CINTA MATRICES REGIONAL SAN RAMON	SISTEMA OPERATIVO OPLAU UCR ** CLAGUT						
ESTUDIOS GENERALES	1973	1974	1975	1976	1977	1978	1979
Política de Admisión	458	573	716	895	1.118	1.398	1.747
Política de Salarios	1.000	1.091	1.136	1.192	1.227	1.273	1.318
Política de Uso de Aulas	1.000	1.100	1.200	1.300	1.333	1.333	1.333
Aumento Relación Prof./Cred.	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Aumento Dif. de Salarios	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Créditos Ofrecidos	8.684	10.865	13.576	16.970	21.198	26.507	33.125
Graduados	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Profesores	9,12	11,41	14,25	17,82	22,26	27,93	34,78
Auditorios 250 Alumnos	0,28	0,40	0,54	0,73	0,93	1,17	1,47
Aulas para 100 Alumnos	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Aulas para 75 Alumnos	1,73	2,39	3,25	4,41	5,64	7,06	8,82
Aulas para 50 Alumnos	0,28	0,40	0,54	0,73	0,93	1,17	1,47
Aulas para 25 Alumnos	1,73	2,39	3,25	4,41	5,64	7,06	8,82
Laboratorios	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Oficinas para Profesores	10	12	15	18	23	28	35
Costo Profesores	281.534	384.279	499.724	650.222	843.153	1.093.650	1.415.082
Costo Administrativo	00	00	00	00	00	00	00
Costo Dif. Area Docente	76.872	102.213	138.365	162.230	192.760	253.328	329.320
Costo Dif. Area Aux. y Equipo	110.934	147.504	199.675	234.114	278.172	365.578	475.242
Subtotal de Costos	469.340	633.996	837.764	1.046.566	1.314.085	1.712.556	2.219.644

Aquí el año guía de las proyecciones fue el año 1973. Los detalles y significado para cada uno de los rubros deben consultarse en la respectiva publicación.

El modelo SIMULA tuvo una continuación. La Oficina de Planificación (OPES) del Consejo

Nacional de Rectores (CONARE) produjo una versión actualizada (CONARE-OPES 1978.)

Ahí se indica que:

El replanteamiento del modelo original resulta en una versión más sofisticada desde el punto de vista estadístico y recoge una

vasta experiencia en la utilización del modelo como indicador programático para la elaboración del presupuesto de la universidad de Costa Rica; constituye por tanto un nuevo aporte del Dr. Gutiérrez a las técnicas de planificación de la educación superior.

Esta segunda versión de SIMULA fue replanteada por un grupo de trabajo de OPES y autorizada por el Dr. Gutiérrez. Se usó una computadora IBM-360-40, bastante más potente que Matilde, con 192 mil posiciones de memoria y un tamaño de palabra de 32 bits. Se programó combinando los lenguajes FORTRAN IV y COBOL. No se utilizó el lenguaje ensamblador de esa otra computadora.

## Computadoras y lenguajes de la época

En la época que nos ocupa existían pocas computadoras disponibles en el país. En el sector de gobierno se puede mencionar la de la Oficina técnica mecanizada del Ministerio de hacienda, la de Estadística y Censos y alguna otra. La programación de la única computadora que existía en la UCR en esa época se hacía principalmente usando FORTRAN II, SPS y lenguaje de máquina. FORTRAN, que aún existe hoy día para programación muy especializada, sobre todo en supercomputadores, era el lenguaje de «alto nivel» para aplicaciones cuantitativas en matemática, estadística, ingeniería, geología y de investigación en general.

Los lenguajes de alto nivel, como FORTRAN en aquel entonces o como Python, C+ y Java hoy día, son más fáciles de leer, escribir y por tanto de programar. Están diseñados para ser independientes del hardware y tienen una sintaxis flexible. Pero resultan lentos para aplicaciones de tipo administrativo en que se manejan bases de datos de cierta magnitud en adelante.

SPS era una versión «mnemónica» (o lenguaje ensamblador, como se conoce en computación) del lenguaje de máquina, es decir, programación de «bajo nivel». Los lenguajes de bajo nivel, como el lenguaje de máquina y el ensamblador, están más cerca del hardware y pueden tener control directo sobre la computadora. Eso los

hace más veloces y pueden aprovechar mejor el espacio de memoria disponible. Pero son más difíciles de aprender, entender y programar.

Claudio Gutiérrez, considerando la poca capacidad de memoria de Matilde y el tipo de aplicación que desarrollaría, intuyó con mucha claridad que sería más eficiente usar SPS para el desarrollo de su proyecto. Pero eso planteó algunos retos. SPS era propio de las máquinas IBM-1620 y había que conocer a fondo su estructura y manejar mucho detalle. Por ejemplo, definir en cual posición de memoria (de las 40 mil que tenía la computadora), se empezaría a alojar las cantidades que representaría los parámetros, variables y datos que se iban a manejar. Para el programador implicaba también más trabajo (más código que escribir). Muchas operaciones rutinarias, para las cuales ya alguien había programado una función o una subrutina en FORTRAN, no existían en SPS. Por ejemplo, a mí me correspondió programar una subrutina para calcular algo tan elemental como la raíz cuadrada. Sin embargo, la gran ventaja estaba en la mayor velocidad de SPS y en la versatilidad del manejo de la escasa memoria. De manera que la escogencia nos lanzó a los brazos de Matilde y eso exigía entenderse bien con ella, lo cual implicaba dominar su lenguaje.

## El lenguaje de Matilde

Antes de referirnos al lenguaje de Matilde, es útil repasar algunos aspectos básicos del lenguaje escrito, dado que se requería una traducción de éste al lenguaje artificial de ella, de alguna manera.

Los humanos nos comunicamos entre nosotros utilizando algún lenguaje que denotaremos como natural. Aquí el término natural es abordado en un sentido restringido a los signos y palabras y hace referencia al modo en que codificamos instrucciones utilizando signos que son fáciles de emplear para nosotros por coincidir con nuestro sistema de escritura.

Aquí vamos a referirnos al lenguaje escrito del idioma inglés (su alfabeto), básicamente porque era el de los creadores de Matilde y la documentación acompañante facilitaría hacer

una traducción al lenguaje cibernético y entendernos con ella. Pero también requerimos de números decimales y símbolos que denoten operaciones con ellos, así como operadores lógicos y quizá otros. ¿Cuál sería ese lenguaje? ¿Cuántos caracteres podría tener?

El alfabeto inglés está compuesto por 26 letras: 5 vocales (A, E, I, O, U) y 21 consonantes (B, C, D, F, G, H, J, K, L, M, N, P, Q, R, S, T, V, W, X, Y, Z). Aquí solo se consideran mayúsculas porque era una limitación que tenía Matilde. Tenemos entonces 26 caracteres que consisten en letras.

Agreguemos ahora los números del sistema decimal: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. También se requieren caracteres de puntuación: . , ; : (punto, coma, punto y coma, dos puntos). Otros caracteres necesarios son los requeridos en operaciones matemáticas y lógicas.

*Aritméticos:* + - \* / ^ (suma, resta, multiplicación, división, exponente).

*Relacionales:* < = > (menor que, igual, mayor que).

*Operadores lógicos:* ~ & | (negación, conjunción, disyunción).

*Otros:* ( ) [ ] { } (paréntesis, corchetes, llaves).

Hay otros símbolos (caracteres) que se pueden o deben agregar, pero la idea es solo ilustrativa. Supongamos que con esto es suficiente para una comunicación escrita razonable. En total llevamos 57 caracteres contabilizados.

Con estos símbolos o caracteres de lenguaje natural escrito nosotros los humanos formamos palabras, frases, oraciones, cifras, ecuaciones, etc. Pero resulta que Matilde, en lo más profundo de su ser solo entendía que le dijeran No (0) o Sí (1) en cada instante de comunicación. Entonces su alfabeto se reducía a dos caracteres: 0, 1 (sistema numérico binario). Cada una de sus respuestas ella la acomodaba en un «bit» o casilla. Ese bit o casilla podía estar ocupado entonces por 0 o por 1, según la respuesta. Pero ella también podía formar caracteres y «palabras» cibernéticas, gracias a que su creador le permitía juntar 6 posiciones o casillas contiguas en un «byte». Entonces, una sola casilla es llamada «bit» (binary digit), pero a 6 casillas contiguas se le llama «byte».

Un bit: un espacio, celda o casilla.



Un byte: 6 casillas contiguas.



Es en este punto donde se puede empezar a explicar cómo se traducía del lenguaje «artificial» de Matilde a instrucciones expresadas de un modo más cercano al lenguaje natural de los humanos que deseábamos interactuar con ella.

Sucedía que Matilde en realidad solo usaba 4 casillas (4 bits) de cada byte, para representar números decimales (bases 10). Los otros 2 bits los usaba para otros menesteres que se mencionan más adelante.

Uso de solo 4 bits (de los 6)



Solo que había una manera muy peculiar de «traducir» cada una de esas 4 casillas. Lo que se hacía era acomodar potencias de 2 en cada una y hacerlo en el siguiente orden:  $2^3$ ,  $2^2$ ,  $2^1$ ,  $2^0$ .

Luego se sumaban los resultados de las potencias de 2 de las casillas que obtenían respuesta positiva (un 1), para representar el dígito en sistema decimal (un número de 0 a 9).

Números propios de Matilde → Traducción → Números para humanos  
 (Sistema binario: 0, 1) (Sistema decimal: 0, 1, 2, ... 9)

				Traductor									
				Suma valores de potencias									
				Bit:	6	5	4	3	2	1			
				Potencia:			2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>			
				Valor:			8	4	2	1			
Binario (Matilde)				Valor potencias de 2				Decimal (humano)					
0	0	0	0				0	0	0	0	0		0
0	0	0	1				0	0	0	0	1		1
0	0	1	0				0	0	0	2	0		2
0	0	1	1				0	0	0	2	1		3
0	1	0	0				0	4	0	0	0		4
0	1	0	1				0	4	0	0	1		5
0	1	1	0				0	4	2	0	0		6
0	1	1	1				0	4	2	1	1		7
1	0	0	0				8	0	0	0	0		8
1	0	0	1				8	0	0	0	1		9

Estas cuatro casillas eran suficientes para representar los decimales del 0 al 9 y aún quedaban posibilidades hasta representar seis símbolos más. Es decir, con cuatro casillas y potencias de 2 surgen posibilidades para representar hasta 16 caracteres (todas las combinaciones posibles de ceros y unos), pero, ya antes se había determinado que se requerían al menos 57 caracteres. ¿Y las letras y todos los otros caracteres, como representarlos?

Aquí podemos decir que Matilde hizo trampa para lograrlo. Lo que hacía era juntar dos bytes contiguos y formar un «doble byte», lo que quizá puede llamarse una «palabra sintética» (cibernética, claro) de 12 bits, aunque luego

solo utilizara 4 bits de cada byte para la parte decimal). Eso le permitía generar hasta 16 x 16 = 256 combinaciones posibles, y aún más si hacía uso del quinto y sexto bit de cada mitad del doble byte (los bits C, F). El bit F (Flag) servía para definir el signo de un dígito, una dirección indirecta o el fin de un campo numérico. El bit C (Check) era para determinar la «imparidad» del byte, algo que tenía que ver con la precisión y exactitud de los cálculos. (Germain 1965). Pero no ahondaremos más sobre estos bits.

Omitiendo ahora los detalles del cálculo del traductor, se ilustra con algunos ejemplos como queda entonces la traducción con el uso de «doble byte» q convierte a caracteres.

Bit de cada byte	Byte 2						Byte 1						Doble byte	Carácter
	6	5	4	3	2	1	6	5	4	3	2	1		
			2 <sup>^</sup> 3	2 <sup>^</sup> 2	2 <sup>^</sup> 1	2 <sup>^</sup> 0			2 <sup>^</sup> 3	2 <sup>^</sup> 2	2 <sup>^</sup> 1	2 <sup>^</sup> 0		
	C	F	8	4	2	1	C	F	8	4	2	1		
							1	0	0	0	0	0	00	0
							0	0	0	0	0	1	01	1
							0	0	0	0	1	0	02	2
							1	0	0	0	1	1	03	3
														...
							0	1	0	0	1	1	F3	-3
														...
	0	0	0	1	0	0	0	0	0	0	0	1	41	A
	1	0	0	1	1	0	1	0	1	0	0	1	69	Z
														...
	1	0	0	0	0	0	0	0	0	0	1	1	C3	.
	0	0	0	0	0	1	1	0	0	0	0	0	1C	+
	0	0	0	0	1	0	0	0	0	0	0	1	21	/
	1	0	0	0	1	1	1	0	0	0	1	1	33	=
Bit del doble byte:	12	11	10	9	8	7	6	5	4	3	2	1		

A esta traducción, que desemboca en la última columna de la tabla anterior (Carácter), se le denominaba «Binario Codificado Decimal» (BCD). En computación se acostumbra a denominar «palabra» al número de bits requerido para representar un carácter. Con una «palabra» (DobleByte) de Matilde se podía entonces formar un carácter: dígito, letra, símbolo especial, por ejemplo 3, A, +, /, ... También números negativos (-3) si se activaba con 1 el bit F del primer byte.

Es razonable decir que Matilde tenía un tamaño de palabra de 12 bits. En comparación, el sistema de una computadora portátil hoy día tiene una longitud de palabra de 64 bits.

Cada carácter requiere una «posición de memoria» para ser almacenado. Matilde fue construida con cuarenta mil posiciones de memoria (memoria de trabajo o RAM). Hoy día

cualquier computadora portátil tiene más de un millón de posiciones de memoria RAM.

## Conversando con Matilde

Hablemos ahora de como se le daba una orden a Matilde. En lo que nos ocupa esto se podía hacer de dos maneras. La primera sería en lo que llegó a denominarse «lenguaje de máquina» (usando los códigos numéricos). La segunda sería usando un «ensamblador» que pasara las ordenes numéricas a símbolos que facilitara recordarlas. Ella poseía un ensamblador denominado SPS (Symbolic Programing System). Este sistema se diseñó de forma que se disponía de 12 caracteres (cada uno era un «doble byte» de 12 bits) para darle una orden a Matilde.

Los dos primeros caracteres denotaban el operador; por ejemplo «TF» es la orden de pasar



el contenido de una posición de memoria a otra. Los siguientes 6 caracteres denotaban el operando 1 (posición de memoria receptora) y los 6 restantes el operando 2 (posición de memoria donante).

Antes presentamos primero (a la izquierda) el código binario y a la derecha el código codificado en decimal. Podemos imaginar que Matilde «decía» algo (en binario) y el humano lo traducía a sistema decimal (con ayuda del traductor). Ahora interviene otro actor que es el código del ensamblador y la dirección de la traducción se invierte: primero «habla» el humano

programador (por medio del SPS) y esto se traduce a decimal y luego a binario, de manera que Matilde sepa que hacer.

Traducción: Código ensamblador → Código decimal → Código binario

Para decirle a Matilde que moviera el contenido de la variable X que estaba en las posiciones de memoria 20000 a 20008, al espacio de la variable Y, que estaba en las posiciones 20032 a 20040, se usaba el operador TF (Transmit field) . Para sumar X+Y se usaba el operador A (Add). En ambos casos el resultado quedaba en el campo Y.

SPS	BCD	Binario
TF Y, X	26 20040 20008	00000 000010 000000 000110 ....
A Y, X	21 20040 20008	00000 000010 000000 000001 ....

Seguidamente se muestra un fragmento de un programa en SPS (cortesía de José Ángel Rojas).

TONOCD TF	A1, J	12572 26 17908 15812
TFM	D1, FLECHA-2	12584 16 17923 J8043
MM	J, 2	12596 13 15812 -0002
A	D1, 99	12608 21 17923 00099
TONO01 SM	D1, 2	12620 12 17923 -0002
SM	A1, 1	12632 12 17908 -0001
BZ	TONOCD-1, 6	12644 46 1257J 01200
TFM	B1, DIRFLE-5	12656 16 17913 J8083
MM	A1, 5	12668 13 17908 -0005
A	B1, 99	12680 21 17913 00099
TF	C1, B1, 11	12692 26 17918 1791L
SM	C1, 17	12704 12 17918 -0017
MM	D1, 20, 67	12716 13 1792L -0020
A	C1, 99	12728 21 17918 00099
AM	C1, 1, 610	12740 11 1791Q 000-1
B7	TONO01	12752 49 12620 00000
DS	5	12763 00005

En esta lista de órdenes ya no se muestra el código binario, pero se agrega la posición de memoria (antes del operador) donde inicia cada orden o comando. Podemos decir que así eran las conversaciones con Matilde, hace ya más de 50 años.

Esas experiencias incluyeron primero el lenguaje artificial de Matilde, después, dominar

el uso de operadores aritméticos y lógicos y sus reglas. Posteriormente vinieron la programación de computadoras con los diagramas de flujo, y la escritura de código ensamblador.

Un detalle curioso que vale la pena contar era la predilección de CLAGUT (como gustaba él abreviar su nombre) por las direcciones indirectas del SPS de Matilde. Él siempre lo

destacaba como algo ingenioso en programación y le gustaba que pudiéramos ponerlo en práctica en el código que se escribía. Esa posibilidad de crear direcciones indirectas permitía encadenar y lograr secuencias interesantes y útiles en las órdenes a Matilde y ahorra tiempo de computación y espacio de memoria. Una instrucción en particular o varias desde otros puntos del programa, podían hacer referencia a una dirección de la memoria de la computadora, pero esta no contenía un dato sino más bien lo que contenía era otra dirección de memoria donde quizá estaba el dato. Una orden posterior podía modificar el contenido de esa dirección y así alterar la secuencia del programa. ¿Una versión rudimentaria de código modificándose a sí mismo? Quizás ahí empezó Claudio Gutiérrez a desarrollar su gusto posterior por la inteligencia artificial.

## Los datos

Una de las contribuciones importantes del proyecto fue el acervo de datos que se reunió, se codificó y que se digitalizó. Los datos existían en papel y muchos estaban disperso por toda la Universidad.

Como lo percibió Claudio Gutiérrez, «El desbalance que a menudo existe entre los requerimientos de decisión y la información disponible se hace cada vez más patente conforme recursos son limitados y la demanda por servicios se expande» (1973). Pero la tarea en realidad era un gran reto en una época en que prácticamente no existía información digitalizada (al menos en CR) y había que empezar casi de cero, primero para conformar bases de datos, procesarlas para obtener estadísticas útiles y luego lograr proyecciones para la planificación del quehacer académico. Y ese era el caso en particular de la UCR.

Información importante existía, por ejemplo, en los expedientes de estudiantes de la Oficina de Registro, o en archivos sobre cursos y profesores en las diferentes unidades académicas y en la oficina de personal. También había información sobre planta física y aspectos financieros en otras oficinas. Salvo por los expedientes estudiantiles, que constituían información de tipo registro y centralizada, el

resto eran datos bastante dispersos en toda la universidad y sin uniformidad en su forma de recolección. Pero todo estaba en papel. Había que iniciar seleccionando lo verdaderamente relevante y diseñar un formato para luego codificar esos datos de manera que fueran aptos para un procesamiento electrónico. Otra información importante para el quehacer académico había que generarla o completarla, por medio de encuestas (o alguna otra técnica estadística) a los jefes de oficina u otros funcionarios. Los datos reunidos para el proyecto se describen brevemente a continuación:

- BIRA (banco de información del rendimiento académico) era quizá la base de datos más importante del modelo; contenía el rendimiento académico de los estudiantes divididos en cortes según año de ingreso. Ello permitía construir una matriz de sobrevivencia académica, al derivarse cuantos estudiantes iban quedando después del primer año de ingreso, segundo año, y sucesivos. Además, se iba conociendo el avance en su carrera y cuantos llegaban a graduarse al término de cuantos años. Esos números se estandarizaban al hacerlos relativos a un ingreso inicial de mil estudiantes (una cohorte). Un dato muy importante en esta base era el de créditos aprobados según curso y unidad académica. Se registraba el carnet (que indicaba el año de ingreso) las siglas del curso, la carrera que seguía y los créditos aprobados o no. También se registraba el género de manera que era posible obtener resultados para mujeres y hombres, aparte y en conjunto. Posteriormente, en la ejecución del modelo de simulación se incorporaba coeficientes de deserción calculados por Instituto de Investigaciones Psicológicas de la universidad.
- BIPA (banco información de profesores) contenía los datos del número de profesores en equivalentes a tiempo completo en cada unidad académica y algunos otros datos relativos a sus profesores.
- El banco de información de cursos tenía el número de horas (teóricas y prácticas) los créditos, el nivel en la carrera y otros datos.

- El banco información de planta física brindaba información sobre disponibilidad de auditorios, laboratorios y aulas de diversas capacidades, según unidad académica.
- Otros datos utilizados por el modelo hacían referencia a costos y otras características. Entre estos datos estaba el sueldo promedio de un profesor para cada unidad académica, costo promedio de la oficina amueblada del profesor, costo de un administrativo, costos de laboratorios, etc.
- La información sobre planta física y aspectos financieros de la institución era manejada por otras secciones asesoras de OPLAU. El mismo Claudio Gutiérrez coordinaba con dichas instancias para el uso de los datos.

## Conclusión

Con el conjunto de programas de computadora de SIMULA era posible predecir la carga de trabajo en créditos que soportaba cada unidad académica de una institución de educación superior en los años que proyectaba el modelo. También proyectaba aspectos administrativos. Esto fue sin duda una ayuda en la elaboración de presupuestos y en la planificación de diversos proyectos entre ellos el de apoyo económico al personal docente para realizar estudios en el exterior, lo cual fue base para un trabajo en que se fijaron prioridades para desarrollo académico de los profesores. (CONARE-OPES, 1978).

Un aspecto esencial en la toma de decisiones para cualquier institución es generar información de su accionar y que esta sea razonablemente de buena calidad. Además, es necesario que esté disponible. Como se mencionó antes en el caso de la universidad de Costa Rica existía información, incluso alguna de muy buena calidad, otra no existía y el resto estaba dispersa por diferentes oficinas de la institución. Una contribución aquí del doctor Gutiérrez fue lograr juntar mucha de esa información en bases de datos y ponerla electrónicamente a disposición de las autoridades universitarias, en una época en que esto no era fácil de llevar a cabo. La simulación del

funcionamiento de una universidad (SIMULA) integró esas bases. El consejo Universitario usaba algunos de sus resultados para definir políticas, por ejemplo, cuantos estudiantes nuevos admitir al año siguiente.

También se logró hacer estimaciones de «coeficientes de transferencia» de un año a otro de cohortes estudiantiles. Algo así como curvas de sobrevivencia de matrícula: cuantos estudiantes de un mismo año de ingreso iban quedando al avanzar en su carrera. ¿Se hace esto hoy día para las diferentes carreras?

Situaciones dignas de resaltar fueron su investigación pionera sobre modelos de simulación en planificación educativa, su empeño en dominar con destreza el uso de una computadora y la acertada escogencia de un lenguaje de programación para su proyecto. No se usó un lenguaje especializado de simulación de los que se podía mencionar para aquel entonces. Ni siquiera usó un lenguaje general de alto nivel como FORTRAN, que estaba disponible en Matilde en aquel momento. Pudo haber intentado modelos econométricos que eran quizá los más elaborados y conocidos. Sin embargo, Claudio Gutiérrez parece haber estado muy conscientes de que lo que tenía entre manos era más cercano a procesos discretos y necesitaba adaptarse a las particularidades de la UCR. Podríamos decir que lo usado para SIMULA fue más bien programación heurística para solución de problemas en que intervienen decisiones humanas (Meier 1969).

SIMULA, aunque basado en la universidad de Costa Rica, era lo suficientemente general para representar una institución de educación superior. Su algoritmo era sencillo pero su aplicación muy laboriosa y ahí fue donde se sacó ventaja del uso de la computadora. Puede decirse que se basaba en las tendencias de las distribuciones de los estudiantes con respecto a las carreras y su avance en el tiempo.

Resumiendo, podría decirse que Gutiérrez y su proyecto SIMULA reunió cantidades de datos, los codificó, los digitó, construyó resúmenes y estadísticas (coeficientes), desarrolló algoritmos muy eficientes en una computadora de aquella época (con una capacidad limitada de memoria) y logró proyecciones que simulaban el funcionamiento de una universidad.

## Referencias

- CONARE-OPES. 1978. *Sistema SIMULA, versión actualizada*. San José, Costa Rica: Consejo nacional de rectores, Oficina de planificación de la educación superior.
- Feoli, Mario. 2018. *Matilde, un ícono. Testimonios de la historia del primer computador de la Universidad de Costa Rica*. San José, Costa Rica: Editorial UCR.
- Gutiérrez, Claudio. 1972. *Un modelo de simulación para planificación universitaria*. San José, Costa Rica: Universidad de Costa Rica, Oficina de planificación universitaria.
- Gutiérrez, Claudio, 1973. *Un simulador integral del movimiento de una universidad latinoamericana: guía para su aplicación*. San José, Costa Rica: Universidad de Costa Rica, Oficina de planificación universitaria.
- Gutiérrez, Claudio. 2010. *El ancho panorama: memorias*. San José, Costa Rica: Editorial UCR.
- Germain, Clarence B. 1965. *Programación IBM 1620. Cuarta reimpresión*. México, D.F.: Limusa-Wiley, S.A.
- Meier, R.C., Newell, W.T., Pazer, H.L. 1969. *Simulation in business and economics*. New Jersey: Prentice-Hall, Inc.
- Juan Bautista Chavarría-Chaves** (juan.chavarria@ucr.ac.cr). Profesor retirado, Universidad de Costa Rica. Bachiller en Estadística de la Universidad de Costa Rica y Máster en Biomatemática de la Universidad de Washington, Seattle. Dentro de sus publicaciones se encuentran: Cubero-Pardo, P., Chavarría-Chaves, J.B., & Romero-Chaves, R. 2021. Distribución espacial y variables explicativas de capturas de *Thunnus albacares* (Perciformes: Scombridae) y especies no objetivo por la flota internacional de cerco en el Pacífico de Costa Rica. *Revista de Biología Tropical* 69, 1: 245-261; Bonilla, Roger E., Chavarría, Juan B. 2014. Mortalidad entre los inmigrantes nicaragüenses en Costa Rica: una aplicación de la regresión geográfica ponderada. *Rev.Mate.Teor.Aplic.* 21, 2: 283–294.

Recibido: 30 de octubre, 2023.  
Aprobado: 6 de noviembre, 2023.