

PROGRAMA ANALITICO

LINEAMIENTOS GENERALES

Para darle un sentido histórico a los particulares recortes pedagógicos de la asignatura se debe comenzar por la primera asignatura que comenzó a dictarse a partir de 1976 bajo la denominación de Computación y Cálculo Numérico, con un Programa basado en los conceptos de los Métodos Numéricos (McCracken y Dorn, 1976) y la implementación de los correspondientes algoritmos (Rice y Rice, 1973) en el lenguaje de programación científica denominado FORTRAN IV (Mc Cracken, 1967). Este programa se mantuvo, salvo por la forma de dictado que desde 1981, se dividió en teóricos de Cálculo Numérico y prácticas de Laboratorio de Programación en el Centro de Cálculo de la U.N.C., hasta la modificación integral de los planes de estudio ocurrida en 1988, a partir de lo cual se divide en dos materias. Computación ubicada en el primer año se ocupa de los conceptos básicos de la Informática y en particular de Programación de Computadoras y Métodos Numéricos en tercer año de los respectivos temas de aplicación en las diferentes ingenierías. Este cambio además estuvo acompañado de un cambio en el lenguaje de programación ya que se adoptó el lenguaje PASCAL (Sanchis, 1980) (Joyanes, 1990) y tanto la formulación de programas así como su diseño se vieron influenciadas por las ideas de la Programación Estructurada (Braunstein y Giogia, 1986) y los Algoritmos y Estructuras de Datos (Wirth, 1986).

Simultáneamente a los cambios originados por las corrientes europeas, si bien influenciados por éstas, se produce en Estados Unidos una difusión masiva del Sistema Operativo Unix en los ámbitos universitarios y del lenguaje con el que se desarrolló que es el C y sus antecesores (Kernigham y Ritchie, 1985) que también sirve al desarrollo de los Sistemas Operativos DOS y Windows de Microsoft. Este lenguaje fue incorporado al dictado durante 1991, y posteriormente fue adecuado durante 1994 a las ideas de Programación Orientada a Objetos en lenguaje C++ (Barclay y Gordon, 1994). La tendencia a la orientación a objetos se ha profundizado con el lenguaje JAVA, que posee una base sintáctica en los lenguajes del tipo C.

Durante la década de los 90 se mantuvo el equilibrio entre los conceptos muy básicos sobre Algoritmos y un lenguaje de programación claramente Imperativo, pero nunca pudo implementarse la asignatura basada en el paradigma de objetos en forma completa por falta de tiempo.

Durante éste mismo período la asignatura de Métodos Numéricos acompañó los cambios en los lenguajes, incluyendo en parte la enseñanza de una Planilla de Cálculo como modo de presentar una herramienta orientada a determinados problemas de ingeniería. Se redactó un manual sobre soluciones a los métodos numéricos con Excell (Gil Montero, 1999), pero como en el caso de la orientación a objetos tampoco se dispuso del tiempo suficiente para su dictado completo, no obstante marcó la necesidad de contar con herramientas de software interactivas y de uso inmediato con capacidades de graficación.

Desde 2001 se produce una nueva bifurcación con dos orientaciones muy pronunciadas, por un lado la profundización de los conceptos de la Ciencia de la Computación en términos de Algoritmos y Estructuras de Datos, que permitió

darle a la asignatura una formalización, similar a las de otras asignaturas de la matemática, en lo que hace al Diseño de Algoritmos (Galvez y otros, 1993) de los que carecía (Braunstein y Giogia, 1986), y además ya es un hecho indiscutible que los conocimientos sobre informática se han hecho imprescindibles como herramienta de Simulación de Sistemas en todas las ciencias y las ingenierías, en particular en las de Electrónica y Computación donde ya no se conciben como temas o disciplinas separadas.

Se destaca que hubiera correspondido estudiar previamente una asignatura de Matemática Discreta (Comellas, 2002), pero hasta el momento las Escuelas de Ingeniería en general y la de Electrónica y Computación en particular no han aceptado la propuesta del Departamento de Computación de su incorporación a los planes de estudio, más que nada por la gran sobrecarga de asignaturas, especialmente del campo de la matemática que ya soportan las carreras. Esta incorporación de la Matemática Discreta será obligatoria al momento de la Acreditación de la carrera de Ingeniería en Computación ya que ha sido incorporada en las propuestas de CONFEDI y de RedUNCI (2005)

La otra dirección importante es la elección, no ya de un lenguaje de programación sino de una herramienta de gran difusión en el desarrollo de aplicaciones interactivas de métodos numéricos y de graficación en ingeniería, denominado MATLAB (Hanselman, 1996). Éste intérprete fue desarrollado en el lenguaje C y es en su funcionalidad de programación, enteramente similar en su sintaxis a la de aquel .

Esta elección se ha basado también en la dificultad de la enseñanza, a estudiantes de primer año, de herramientas informáticas y lenguajes de programación para aplicarlas a Problemas de Ingeniería que desconocen casi completamente, inclusive carecen de conceptos de física básica, lo que impide el uso de material didáctico orientado a problemas (Biran y Breiner, 1999) y en relación a los Métodos Numéricos (Nakamura, 1997). Se hizo necesario adaptar los textos de MATLAB a la enseñanza de la programación y los algoritmos para lo cual se desarrolló un Apunte Operacional, y que es un texto ejecutable en el intérprete como Literatura Computacional (van Dyke, 1987) lo que permite unir en un mismo texto las explicaciones teóricas significativas con el código correspondiente. Éste material se desarrolló en 2000 con la finalidad de ser utilizado como transparencias de ejemplos de problemas y ejemplos resueltos de cierta longitud y complejidad, lo que no puede lograrse en un tiempo razonable, con uso exclusivo del pizarrón.

Durante 2001, 2002 y 2003 se ha incrementado el uso del Software Libre en ingeniería, en particular la existencia del émulo de MATLAB denominado GNU-OCTAVE (Eaton, 1997) lo que permitió utilizar dicho software sin restricciones de licencias de ningún tipo. Actualmente se está estudiando en la cátedra otro lenguaje de emulación denominado SCILAB (Gómez, 1999), que si bien es de propiedad del INRIA su uso es gratuito pero no es libre el uso del código fuente y sus prestaciones son superiores en algunos casos al del mismo MATLAB actual.

En este año 2007, con el objetivo de unificar criterios de dictado y evaluación, se propone utilizar como único lenguaje de programación el lenguaje C++ ya que de acuerdo a las experiencias previas del dictado de dos lenguajes (Matlab y C++) ha resultado altamente inconveniente y definitivamente imposible de cumplir en el término de un cuatrimestre con un nivel de adquisición de conocimientos que pueda considerarse como una competencia adquirida.

Uno de los aspectos centrales del Método Científico y en particular el de la Física, que es una de las ciencias básicas que sustentan los distintos diseños y obras de las ingenierías, es la experimentación, es decir, la posibilidad de repetir experiencias de una manera controlada, de forma tal que se pueda poner en evidencia la corrección de las Teorías Formales en las que se sustentan.

Esta interpretación de la experimentación es el que debe animar al estudiante ante los problemas de la ingeniería ya que las herramientas informáticas adecuadas le permitirán poner a prueba su propia comprensión de las teorías, mediante la observación de los resultados controlados por la lógica de la especificación de programas de simulación y su puesta en funcionamiento mediante su implementación computacional.

La ubicación de la asignatura de Informática en el primer año de los diferentes planes de estudio de las ingenierías, pone a prueba a quienes ingresan al sistema universitario, frente a un sistema de aprendizaje basado en el autocontrol del propio conocimiento, lo que requiere ante todo dejar de creer en los textos sin un pensamiento crítico propio y por ende le exige poner a prueba y experimentar con la realidad acotada o simulación que es posible construir mediante la matemática, la física y la informática.

Otro aspecto, que a lo largo de varios años de enseñanza de la informática se ha puesto en evidencia, es la gran similitud entre con el aprendizaje de idiomas extranjeros, y también con el entrenamiento deportivo. En todos los casos la construcción de estructuras de pensamiento que permiten hablar o jugar en forma automática, requieren de un considerable tiempo dedicado a la práctica. Baste recordar que no es posible hablar ningún idioma extranjero mientras no se puede pensar en él y esto se logra luego de varios años de estudio o convivencia.

En síntesis, para entender, y por ende aprender, no basta con aceptar como buenas las relaciones lógicas, por correctas que éstas sean, que momentáneamente parecen evidentes y casi de sentido común cuando son enunciadas por el profesor en el dictado de una clase o en una lectura circunstancial, es imprescindible dudar, poner a prueba, experimentar, para lo cual nada más gratificante y creativo que construir una solución algorítmica a un problema y comprobar que funciona!

METODOLOGIA DE ENSEÑANZA

Las etapas de construcción y elaboración de conocimientos son sustentadas mediante la exposición dialogada como estrategia didáctica y el empleo de proyección de diapositivas, transparencias, pizarrón y proyector multimedia como materiales didácticos. Todos los materiales de estudio, incluyendo sistema de consultas, preguntas frecuentes, e-mail, evaluaciones, etc., se disponen en el sistema informático de aprendizaje del Departamento de Computación (Laboratorio de Enseñanza Virtual - LEV. <http://lev.efn.uncor.edu>)

La fase de ejercitación y aplicación de los contenidos de la signatura, se fundamenta tanto en el desarrollo teórico como en el práctico del presente curso. Se realizan dos tipos diferenciados de actividades en coordinación con el desarrollo de la autonomía de aprendizaje, consistentes en la solución de problemas acotados y en la elaboración de un proyecto informático integrador realizado en equipo. En estas instancias el trabajo individual y grupal, permite la conformación de ideas y el establecimiento de relaciones entre el conocimiento

adquirido y situaciones nuevas planteadas desde otras problemáticas de la misma disciplina.

El dictado se realizará en 16 clases de 5hs 25min (reloj) consistentes en la presentación teórica de los temas por parte del docente, las que no podrán superar 3hs:00min en cada sesión.

La presentación de temas prácticos se realizará preferentemente, en el caso de disponer de equipos de computación, en el marco de tareas de laboratorio, previamente asignadas por el docente coincidentes con el tema teórico previo, asumiendo el docente el rol de tutor y mediante evaluaciones formativas en cada clase.

En el caso de no disponer del laboratorio se realizarán ejercitaciones y simulaciones de escritorio que permitan poner de manifiesto los objetivos de la asignatura.

El proceso de elaboración del proyecto integrador será seguido mediante entregas parciales pautadas en el LEV, así como la devolución de las evaluaciones parciales.

Programación de actividades y bibliografía recomendada

Clase	Tema	Capítulo*
1	Unidad 1	1
2	Unidad 2	2
3	Unidad 2	3
4	Unidad 2	3
5	Unidad 3	4
6	Unidad 3	4
7	Unidad 3	5
8	Evaluación parcial - P1	

Clase	Tema	Capítulo*
9	Unidad 4	6
10	Unidad 4	7
11	Unidad 5	11
12	Unidad 5	12
13	Unidad 6 -P2	13
14	Unidad 7	8
15	P3	
16	Parcial de recuperación	

* Corresponde con el Capítulo del material bibliográfico básico:

Bronson, G. (2007). C++ para Ingeniería y Ciencias.(2da. edición) International Thomson Editores, México.

CALENDARIO - 2008

INICIO PRIMER CUATRIMESTRE LECTIVO 10/3 - 2008

Semana	Clase	Tema	Capítulo*
10/3 a 15/3	1	Unidad 1	1
17/3 a 19/3		Unidad 1	1
20/3 y 21/3	Feriados semana santa (se recomienda no iniciar tema nuevo)		
24/3 a 29/3	2	Unidad 2	2
31/3 a 01/4	3	Unidad 2	3
2/4	Feriado Miércoles		
03/3 a 05/4	3	Unidad 2	3
07/4 a 12/4	4	Unidad 3	4
14/4 a 19/4	5	Unidad 3	5
21/4 a 25/4	6	Unidad 3	5

01/5 Feriado Jueves (inamovible) Pasar los parciales al sábado

28/04 a 03/05 1era evaluación parcial (U1-U2-U3) ON-LINE en el LEV

05/5 a 10/5	7	Unidad 4	6
12/5 a 17/5	8	Unidad 4	7
19/5 a 24/5	9	Unidad 5	11
25/5	Feriado lunes(inamovible)		
26/5 a 31/5	10	Unidad 5	12
02/6 a 06/6	11	Unidad 5	13
09/6 a 13/6	12	Unidad 6	13

14/6 sábado 2da evaluación parcial (U4-U5)

16/6	Feriado lunes (móvil)		
17/6 a 21/6	13	Unidad 6	13
23/6 a 27/6	14	Unidad	

28/6 sábado 3ra evaluación parcial (U6-U7)

30/6 lunes Recuperación 3ra evaluación parcial (U6-U7)

FIN PRIMER CUATRIMESTRE LECTIVO 27/6 - 2008

INICIO SEGUNDO CUATRIMESTRE LECTIVO 04/08 - 2008

Semana Clase Tema Capítulo*

04/08 a 08/08 1 Unidad 1 1

11/08 a 15/08 2 Unidad 2 2

18/08 Feriado lunes (movil)

19/08 a 22/08 3 Unidad 2 3

25/08 a 29/08 4 Unidad 2 3

01/09 a 05/09 5 Unidad 3 4

08/09 a 12/09 6 Unidad 3 4

15/09 a 16/09 7 Unidad 3 5

17/09 Feriado miércoles

18/09 a 19/09 7 Unidad 3 5

22/09 a 26/09 8 1era evaluación parcial unificada (U1-U2-U3) ON-LINE en el LEV

29/09 a 03/10 9 Unidad 4 6

06/10 a 10/10 10 Unidad 4 7

13/10 a 17/10 11 Unidad 5 11

20/10 a 24/10 12 Unidad 5 12

27/10 a 31/10 13 Unidad 6 13

01/11 sábado 2da evaluación parcial unificada (U4-U5)

03/11 a 07/11 14 Unidad 7 8

10/11 a 14/11 15 Unidad 7 8

17/11 a 21/11 16 Repaso

22/11 sábado 3ra evaluación parcial unificada (U6-U7)

29/11 sábado Recuperación 3ra evaluación parcial unificada (U6-U7)

FIN SEGUNDO CUATRIMESTRE LECTIVO 21/11

EVALUACION

Evaluaciones Formativas

- Las actividades de Laboratorio consistirán en la puesta en funcionamiento de los Ejercicios y Problemas de especificación de programas que acompañan al enunciado de los temas conceptuales.
- Se utilizará el lenguaje informático asignado para la formulación, resolución e implementación de programas compilados directamente sobre la arquitectura del computador.
- Se desarrollará un Proyecto Integrador conformado por un equipo de estudiantes.
- Ambas actividades se considerarán como de realización necesaria para la acreditación del porcentaje de asistencia total. El estudiante pondrá a disposición de los profesores los trabajos realizados, mediante el uso del Laboratorio de Educación Virtual (LEV).

1era Evaluación Parcial de Acreditación

Tiene por objeto acreditar que el alumno ha alcanzado, individualmente, las siguientes metas de aprendizaje, en relación a las unidades 1, 2 y 3 del programa analítico:

- Deberá interpretar código en el lenguaje de programación indicado con relación los siguientes tipos de preguntas que se le presentarán sobre:
 - Errores sintácticos.
 - Resultados esperados de la ejecución del programa.
 - Estado de las variables en pasos intermedios de la ejecución.
 - Errores lógicos como tipos de datos erróneos, operaciones matemáticas y lógicas erróneas con respecto a la especificación en pseudocódigo, etc.

Características generales:

- La evaluación se realizará en las fechas y horarios indicados en el calendario, ON-LINE en el LEV. **Las consignas de evaluación estarán unificadas para todos los grupos con independencia del profesor asignado.**
- Consistirá en un cuestionario de tipo objetivo con opciones múltiples.
- La calificación del desempeño será de un porcentaje 0% al 100% y el peso relativo del 25% del total.

2da Evaluación Parcial de Acreditación

Tiene por objeto acreditar que el alumno a alcanzado, individualmente, las siguientes metas de aprendizaje, en relación a las unidades 4, y 5 del programa analítico y considerando los conceptos necesarios de las unidades anteriores:

- Interpretar el enunciado de la solución en pseudocódigo de un problema de ciencias básicas de la ingeniería o de información e implementarlo en el lenguaje de programación indicado.
- Interpretar algoritmos imperativos basados en el control de las acciones de la computadora y administrar la entrada y salida básica de datos y resultados.

- Ser capaz de definir funciones y utilizarlas.
- Ser capaz de utilizar conjuntos de datos mediante arreglos y punteros.

Características generales:

- La evaluación se realizará en las fechas y horarios indicados en el calendario. **Las consignas de evaluación estarán unificadas para todos los grupos con independencia del profesor asignado.**
- Consistirá en la especificación de algoritmos completos en el lenguaje asignado.
- La calificación del desempeño será de un porcentaje de 0% al 100% y el peso relativo del 35% del total.

3era Evaluación Parcial de Acreditación

Tiene por objeto acreditar que el alumno ha alcanzado, individualmente, las siguientes metas de aprendizaje, en relación a las unidades 6 y 7, considerando los contenidos de las anteriores unidades del programa analítico:

- Comprender los principios necesarios para generalizar las soluciones específicas de los problemas científicos y de ingeniería a modelos de simulación mediante herramientas informáticas basadas en los algoritmos matemáticos.
- Ser capaz de analizar, representar y resolver los problemas científicos y de ingeniería en el lenguaje formal de programación por procedimientos indicado.
- Habilidad para analizar un problema de información en términos de datos compuestos y persistentes.

Características generales:

- La evaluación se realizará en las fechas y horarios indicados en el calendario. **Las consignas de evaluación estarán unificadas para todos los grupos con independencia del profesor asignado.**
- Consistirá en la solución algorítmica e implementación en el lenguaje asignado de uno o más problemas de ingeniería o de sistemas de información.
- La calificación del desempeño será un porcentaje del 0% a 100% y el peso relativo del 40% del total.

Condición de regularidad

Para alcanzar la condición de ALUMNO REGULAR se deberán cumplir los siguientes requisitos excluyentes:

- Asistir al 80% de las clases teóricas y de laboratorio y aprobar las actividades correspondientes a las mismas.
- Aprobar dos exámenes parciales con un porcentaje de desempeño igual o superior al 60%.

Régimen de promoción

Aprobación de la materia:

Para lograr la promoción se deberán alcanzar los siguientes objetivos excluyentes:

- Asistir al 80% de las clases teóricas y de laboratorio y aprobar las actividades correspondientes a las mismas.
- Aprobar los tres exámenes parciales con un porcentaje de desempeño igual o superior al 60%.
- Obtener una calificación final con nota cuatro (4) o superior

Recuperación de parciales:

- Sólo se podrá recuperar uno de los tres exámenes parciales mediante la aprobación del examen parcial recuperatorio con un porcentaje de desempeño igual o superior al 60%.
- El examen parcial recuperatorio se realizará en una única fecha al terminar el dictado de la materia y versará sobre los temas del parcial no aprobado que corresponda. Es decir que se administrarán tres temas correspondientes a cada parcial y el estudiante podrá realizar el que le corresponda a su parcial no aprobado.

Calificación final:

La calificación es el promedio ponderado de las diferentes evaluaciones y su valor numérico se establece como:

$$\text{Nota Final} = \text{redondear}(\text{PorcentajeDesempeño 1er Parcial} / 10 * 0.25 + \\ \text{PorcentajeDesempeño 2do Parcial} / 10 * 0.35 + \\ \text{PorcentajeDesempeño 3er Parcial} / 10 * 0.40)$$

CONTENIDOS TEMATICOS

Unidad 1. Introducción a la informática

Introducción a la Programación. Solución de problemas y desarrollo de software. Algoritmos. Errores en programación. Hardware y conceptos de almacenamiento.

Unidad 2. Introducción a la especificación de programas imperativos

Herramienta de programación. Estilo de programación. Constantes y operaciones aritméticas. Variables y declaraciones. Tipos de datos. Procedimiento para el desarrollo de software. Operaciones de asignación. Formato de salida. Funciones de biblioteca. Entrada y salida estándar de información. Aplicaciones.

Unidad 3. Estructuras de control

Operadores lógicos y relacionales. Estructuras de decisión. La estructura de decisión simple. La estructura de decisión doble. Estructuras de decisión anidadas. La estructura de decisión múltiple. Estructuras de repetición. Las estructuras de repetición indefinidas. La estructura de repetición definida. Estructuras de repetición anidadas. Técnicas de programación estructurada. Aplicaciones.

Unidad 4. Funciones definidas por el usuario

Declaración de funciones y parámetros. Prototipos. Argumentos. Alcance de variables. Clases de almacenamiento de variables. Recursividad. Aplicaciones.

Unidad 5. Tipos de datos arreglos y punteros

Arreglos unidimensionales. Inicialización de arreglos. Arreglos bidimensionales. Arreglos como argumentos. Algoritmos de búsqueda y ordenamiento. Aplicaciones. Direcciones y punteros. Nombres de arreglos como punteros. Transmisión de direcciones.

Unidad 6. Estructuras de datos compuestos

Estructuras sencillas. Arreglo de estructuras. Estructuras como argumentos de función. Listas enlazadas. Asignación dinámica de estructuras de datos. Uniones.

Unidad 7. Entrada/salida de información

Lectura y escritura de archivos. Acceso aleatorio de archivos. Flujo de archivos como argumento de función. Excepciones y comprobación de archivos. Bibliotecas de entrada/salida. Aplicaciones.

1. LISTADO DE ACTIVIDADES PRACTICAS Y/O DE LABORATORIO

Actividades Prácticas

1.- Actividades de Laboratorio

El alumno realizará actividades de programación en el Laboratorio de Computación que se corresponden con los ejercicios propuestos como actividades de práctica.

- 1.- Sistema Operativo y Ambiente de Desarrollo.
- 2.- Especificación de programas imperativos.
- 3.- Especificación de unidades de programación.
- 4.- Especificación de datos compuestos de tipo arreglo.
- 5.- Especificaciones de datos compuestos.
- 5.- Comunicación de la información.

2.- Actividades de Proyecto y Diseño

Tiene por objeto acreditar que el alumno ha adquirido las siguientes habilidades y técnicas, relacionadas preferentemente a la totalidad de los contenidos de la asignatura:

- Aplicar la informática a un problema de ciencias básicas, de ingeniería o de sistemas de información, desde su formulación simbólico-matemática o de información hasta su implementación en un lenguaje informático.
- Adquirir la habilidad para la depuración de algoritmos y programas mediante una técnica basada en principios lógicos.
- Experimentar con diferentes criterios de diseño.
- Capacidad para el trabajo en equipo en la planificación y ejecución de un proyecto informático.

Características generales:

- El proyecto consistirá en el desarrollo de los algoritmos matemáticos y/o de información que den solución a un problema de ciencias o ingeniería.
- Se implementará la solución en el lenguaje definido y se probarán diferentes criterios de diseño y se presentarán todas las versiones de los archivos de código fuente. El diseño debe constar como mínimo de funciones y procedimientos que permitan definirlo como de arquitectura modular.
- La aplicación resultante deberá poderse ejecutar en un ambiente de Windows o de Linux sin errores sintácticos ni lógicos.
- Se documentará la presentación mediante una monografía sobre el tema, los criterios adoptados al respecto del diseño, como estructuras de datos, eficiencia algorítmica, interfaces con el usuario, etc.
- Constará de un manual de usuario o ayuda en línea
- Los grupos estarán constituidos por 4 alumnos como máximo.
- La presentación se realizará durante las clases de laboratorio correspondientes al último mes de clase.

2. DISTRIBUCION DE LA CARGA HORARIA

ACTIVIDAD	HORA S
TEÓRICA	32
FORMACIÓN PRACTICA:	
o FORMACIÓN EXPERIMENTAL	22
o RESOLUCIÓN DE PROBLEMAS	20
o ACTIVIDADES DE PROYECTO Y DISEÑO	10
o PPS	
	84

DEDICADAS POR EL ALUMNO FUERA DE CLASE

ACTIVIDAD	HORA S
PREPARACION TEÓRICA	24
PREPARACION PRACTICA	
o EXPERIMENTAL DE LABORATORIO	0
o EXPERIMENTAL DE CAMPO	0
o RESOLUCIÓN DE PROBLEMAS	36
o PROYECTO Y DISEÑO	24
TOTAL DE LA CARGA HORARIA	84

3. BIBLIOGRAFIA

Básica

- Bronson, G. (2007). C++ para Ingeniería y Ciencias (2da. edición) International Thomson Editores, México.

Recomendada

- Deitel, H. M., Deitel, P. J. (2003). Cómo programa en C++. Pearson Educación.
- Galve, J. et al. (1993). Algorítmica. Diseño y análisis de algoritmos funcionales e imperativos. Addison-Wesley Iberoamericana / Ra-Ma.
- Wirth, N. (1999). Algoritmos + Estructuras de Datos = Programas. Dossat.