

El proyecto interdisciplinario en la formación de profesores en Matemática

María Florencia Gonzalez^{1,2}; Carolina Gallo^{1,3}; Ana Lucía Saino^{1,4};
Adriana N. Magallanes^{1,5}

¹ Departamento de Matemática, Universidad Nacional de Río Cuarto.

² florenciagonzalez269@gmail.com ³ carolinagallo91@gmail.com

⁴ lucia.s1818@gmail.com ⁵ adriana.n.magallanes@gmail.com

Resumen

En este trabajo se presenta una experiencia realizada en la formación de estudiantes del profesorado en matemática. Se propone la planificación de un proyecto como estrategia pedagógica para la educación matemática con la intención de potenciar el trabajo interdisciplinario en la escuela secundaria. En el proyecto elaborado se busca resignificar el objeto matemático “divisor común mayor” a través de una tarea de programación. Esto excede la aplicación lineal de técnicas de una asignatura para resolver problemas de la otra, sino que exige un abordaje integral del problema desde ambas disciplinas, alrededor del cual ambas asignaturas pueden construir, resignificar y ejercitar conocimientos. Se exhibe el proyecto elaborado por las profesoras en formación y se enuncian algunas reflexiones en torno a este tipo de trabajo. Esta experiencia brindó una oportunidad para vivenciar la generación de un espacio de aprendizaje en el que surgieron discusiones, coflexiones, toma de decisiones colectivas; todo en un ambiente de respeto, confianza y valoración del pensamiento del “otro”.

Palabras clave: proyecto interdisciplinario; educación matemática; formación de profesores.

Introducción

El proyecto nace en el marco del Taller Interdisciplinario. Esta asignatura con modalidad taller se encuentra en el cuarto y último año del plan de estudios para el Profesorado en Matemática de la Universidad Nacional de Río Cuarto. Con este taller se pretende, en líneas generales, analizar diferentes interrogantes en relación a esta particular filosofía y marco metodológico (Ander-Egg, 1994; Carvajal, 2010) limitando este análisis al ámbito específico de la educación secundaria en nuestro país y poniendo el foco en la educación matemática (Almidón López Huancayo, 2017).

En la resolución N°188/2018 del Ministerio de Educación de la Provincia de Córdoba, se habla de la incorporación de “jornadas interdisciplinarias de integración de saberes”. Estas jornadas constituyen instancias de trabajo escolar colectivas obligatorias para todos los estudiantes, en la que los profesores aportan, desde los espacios curriculares que enseñan, a la problematización y comprensión de temáticas de relevancia social contemporánea para los estudiantes y saberes emergentes. Pero, en la formación de profesores, no es común encontrar espacios curriculares destinados a fortalecer capacidades que favorezcan la implementación de este tipo de experiencias interdisciplinarias.

Por otra parte, Valero (2017) considera que un modelo de aprendizaje basado en problemas (ABP) como una estrategia pedagógica, permite generar el espacio para operar dentro de ambientes interdisciplinarios.

Atendiendo a todo lo expuesto, como trabajo final para la aprobación del taller, los profesores en formación planifican un proyecto interdisciplinario para la educación matemática en el nivel secundario. Se procura formular nuevas alternativas en el ejercicio de la profesión; lo que requiere un análisis de las posibilidades y potencialidades de este tipo de filosofía y metodología en las actuales condiciones laborales de docentes de escuelas secundarias.

Durante el ciclo lectivo 2018, un grupo de estudiantes del profesorado en matemática y la docente responsable del taller se contactaron con un profesor en ciencias de la computación para comenzar a pensar en la elaboración de un proyecto interdisciplinario. En el primer encuentro, se definió el tema y se pudo reflexionar sobre la importancia de la formación específica e interés de los docentes involucrados en el proyecto en el momento de precisar la temática a trabajar. Al mismo tiempo fue posible vivenciar que las posibilidades y potencialidades del grupo de docentes y especialistas que se involucran

con el proyecto, es un condicionante que siempre está presente en el trabajo interdisciplinario, además de las posibilidades del grupo de estudiantes que serán los participantes del proyecto. De la primera jornada de trabajo con el docente de computación surge la idea de que una potencialidad del trabajo interdisciplinario entre matemática e informática es que uno debe modificar de alguna manera la matemática (resignificar, modificar el objeto matemático) para poder programar.

Las jornadas sucesivas de trabajo se utilizaron para definir con mayor precisión la problemática, y comenzar a explicitar los distintos aspectos del proyecto. Seguidamente se presenta una síntesis del proyecto elaborado por los estudiantes del profesorado y algunas de las reflexiones realizadas por ellos una vez concluido el proceso de escritura del mismo.

La fundamentación del proyecto

El proyecto se planifica para ser implementado en quinto año de una escuela secundaria con orientación en informática. Está pensado para potenciar la interdisciplinariedad entre dos áreas curriculares que conviven en esta orientación: matemática e informática.

Para hacer posible su implementación se requiere de conocimientos previos sobre programación e informática. Específicamente, se requiere que los estudiantes posean conocimientos de resolución de problemas usando algoritmos y sean capaces de codificar dichos algoritmos en algún lenguaje ejecutable. Aunque parte del proyecto está dedicado a recuperar, ampliar e introducir algunos de estos conocimientos, es necesario que los estudiantes estén familiarizados con técnicas de resolución de problemas propias de la programación como la representación de datos, división en sub-problemas, abstracción en funciones/procedimientos, codificación de algoritmos en lenguajes de programación y su transformación en programas ejecutables, y conocimiento sobre las herramientas de soporte para llevar a cabo esta tarea (compiladores e intérpretes).

Este proyecto recurrirá para su implementación a la codificación de algoritmos bajo el paradigma de programación imperativo, pero se introducirán conceptos subyacentes a la programación de forma independiente del lenguaje, de manera que el estudiante sea capaz de adaptarse en el futuro a nuevos lenguajes sin mayores inconvenientes. De esta manera, se fortalecerán los conceptos generales de algoritmos y resolución de problemas con ellos, aportando a los contenidos curriculares propios de la asignatura Informática.

El objetivo matemático es revalorizar la definición de divisor común mayor (DCM) como técnica para la búsqueda del mismo. Contar con plataformas informáticas nos brinda la posibilidad de sortear obstáculos asociados al uso de la definición como técnica para hallar el DCM (lentitud, poca eficiencia, repetitividad). Los estudiantes deberán escribir un programa que les permita hallar el DCM entre dos números naturales cualquiera. Esto obligará a poner a prueba y reforzar su comprensión sobre este objeto.

Otro de los objetivos es que los alumnos puedan adaptar los conocimientos de programación adquiridos a la resolución de problemas matemáticos simples y de mediana complejidad. Se utilizará un lenguaje de programación específico, Octave. Se selecciona este programa por ser libre y gratuito, y además porque las estudiantes del profesorado poseen conocimientos previos sobre dicho programa. De esta forma, el proyecto involucra e integra conocimientos específicos sobre matemática y programación, a la vez que permite un crecimiento conjunto (un avance tanto desde lo matemático como desde la informática), al unir el pensamiento propio de cada asignatura con el propósito de dar respuesta a un problema específico. Cada una aporta a la otra, y no se trata de una aplicación lineal en la que repasamos matemática al tener que programar lo que ya sabíamos hacer en papel, sino que cambia la concepción sobre el objeto ante la necesidad de adaptarlo al lenguaje y metodología propios de la programación.

Algunos detalles del Proyecto Interdisciplinario

Este proyecto se planifica con dos objetivos generales. El primero consiste en generar un clima de trabajo que rompa con la disciplinaria estricta y el segundo pretende promover el trabajo interdisciplinario entre matemática e informática. Se formulan además; los siguientes objetivos específicos: resignificar la definición de divisor común mayor y revalorizarla como técnica para la búsqueda del mismo; emplear la programación por ciclos para la resolución de problemas matemáticos y adaptar los conocimientos de programación adquiridos a la resolución de problemas matemáticos simples y de mediana complejidad. Para lograr los mencionados objetivos, se planifican seis clases; las que se describen a continuación:

La primer clase se propone con el objetivo de recuperar la definición divisor común mayor, a partir de nuevas tareas y las técnicas para calcularlo.

Se selecciona una tarea de divisor común mayor entre dos números con la intención de

que surjan las técnicas o la definición de estos conceptos que los estudiantes ya tienen disponibles. Los números involucrados en las tareas permitirán que los estudiantes puedan trabajar utilizando como recurso solamente lápiz y papel, poniendo a funcionar el teorema fundamental de la aritmética o la definición, esto último teniendo en cuenta una de las propiedades que creemos que sería útil a la hora de programar, generando la necesidad de utilizar esta propiedad para realizar un procedimiento más económico. Esta clase se planifica para ser desarrollada en 80 minutos aproximadamente y a cargo del profesor de Matemática. Se plantean la tarea 1 para ser resueltas en clase en los primeros 20 minutos.

Tarea 1: Resolver el siguiente problema, dejando por escrito los cálculos que realizas.

Andrés tiene una cuerda de 120 metros y otra de 96 metros. Desea cortarlas de modo que todos los tramos sean iguales pero lo más largos posible. ¿De qué medida deben ser los tramos?

En los siguientes 60 minutos se plantea una puesta en común donde surjan las definiciones o técnicas que tienen disponibles los estudiantes y han utilizado en la resolución de estos problemas. Para poder cortar ambas cuerdas en trozos iguales, la longitud de los trozos debe dividir la longitud de ambas cuerdas. Es decir, debe ser un divisor de 120 y de 96.

Además, esta longitud debe ser la máxima. Por tanto, debemos calcular el divisor común mayor (DCM) de las longitudes.

Se pondrán en discusión las diferentes resoluciones, con el objetivo de concluir en cuáles son las técnicas que se tienen disponibles, y cuáles son las ventajas de una sobre la otra, sobre todo que unas resultan más económicas que otras. Se repasa la definición de D.C.M. preguntando en una primera instancia qué es un divisor de un número (un número es divisor de otro si lo divide “en forma exacta”, es decir, si al realizar la división entera, el resto es 0) y luego, de que distintas maneras puedo hallar el D.C.M. entre dos números. Se espera recuperar, al menos, las siguientes definiciones:

- el mayor de los divisores comunes a ambos.
- mayor número entero que los divide a ambos sin dejar residuo.
- el divisor común a ambos que a su vez es divisible por todos los demás divisores comunes.

La clase 2, se propone con el objetivo de introducir y/o recuperar¹ diagrama de flujo y la programación por ciclos. Se propone para ser desarrollada en 80 minutos y a cargo del profesor de computación. Los siguientes conocimientos a desarrollar son los que se consideran necesarios; además de los conocimientos para la creación del programa que permita calcular el divisor común mayor. Los diagramas de flujo serán una instancia intermedia entre el trabajo manual y la programación, y los ciclos que se han seleccionado para estudiar serán claves en la escritura del programa.

Esta clase estará centrada en qué es un diagrama de flujo y cómo se construye (cuándo corresponde usar cada una de las distintas figuras y cómo se conectan entre sí). Se entiende por diagrama de flujo la representación gráfica de un algoritmo o proceso. Son los pasos a seguir, representados en un gráfico, que permiten ver cómo funciona un programa. Por otro lado, es necesario que los alumnos cuenten con los siguientes conocimientos de programación por ciclos: Ciclos a estudiar (en pseudocódigo); Mientras; Para i desde x hasta n , con paso z , repetir y Repetir hasta.

En el ciclo “mientras” el bucle se repite mientras la condición sea cierta, si al llegar por primera vez al bucle mientras la condición es falsa, el cuerpo del bucle no se ejecuta ninguna vez. En ciclo “para” una estructura de control muy común es el ciclo *FOR*, la cual se usa cuando se desea iterar un número conocido de veces, empleando como índice una variable que se incrementa (o decrementa). En el ciclo “repetir hasta” la estructura de control *repetir* se utiliza cuando es necesario que el cuerpo del bucle se ejecuten al menos una vez y hasta que se cumpla la condición.

La planificación de esta clase queda a cargo del profesor de computación.

La clase 3 se propone con el objetivo de introducir el lenguaje de programación de Octave. Se propone para ser desarrollada durante 160 minutos aproximadamente y también estará a cargo del profesor de computación.

Se mencionan a continuación los comandos que se espera que sean abordados:

Tabla 1: Comandos-Funcionalidad-Pseudocódigos (elaboración propia)

Comandos de Octave	Funcionalidad	Pseudocódigo	Tener en cuenta	Comandos de Octave	Funcionalidad	Pseudocódigo	Tener en cuenta
Input	Que el usuario	Ingrese	Se debe asignar a	min	Buscar el mínimo en	Mínimo	Si tengo un vector

¹ Esto depende de lo que hayan trabajado anteriormente en la asignatura.

	ingrese un valor de variable.		una variable		una matriz		no lo busca. Debo convertir el vector en matriz.
For i=1:n	Iterar n veces.	Para i desde 1 hasta n, repetir	Se termina con endfor. Debe tener instrucciones para repetir.	strcat	Unir cadenas de caracteres	Concatenar	
While	Iterar mientras una variable es verdadera	Mientras	Se termina con endwhile. Necesita una condición para evaluar.	num2str()	Convertir número en cadena	Número a cadena	El argumento es un número.
Switch	Ejecutar instrucciones de acuerdo al caso	Casos	Para cada caso instrucciones diferentes.	cell2mat()	Convertir un vector a matriz	Vector a matriz	El argumento es un vector.
Cell()	Armar un vector vacío	Celda / Vector	Se debe asignar a una variable y ese es el nombre del vector.	clc	Limpiar la ventana de variables		
Printf	Mostrar en pantalla ().	Mostrar en pantalla	Se puede utilizar strcat dentro de esta función.	clear	Limpian la ventana de comandos		
max	Buscar el máximo en una matriz	Máximo	Si tengo un vector no lo busca. Debo convertir el vector en matriz.	If / else / endif	Se da una condición, si se cumple, el programa ejecuta instrucciones si no ejecuta	Si entonces / sino / Fin del si	Hay que dar condiciones dicotómicas. Si las condiciones tienen más de

					otras.		dos casos posibles, se usa switch.
--	--	--	--	--	--------	--	------------------------------------

Cada función tiene sus argumentos, es fundamental el orden y el formato de estos. La cadena de caracteres se ingresa utilizando comillas a ambos lados. Por ejemplo: ‘Ingrese un número’, cuando está bien ingresado se marca en amarillo. Para hacer referencia a una celda de un vector, se pone el nombre del vector y entre llaves la posición.

Se utilizan los siguientes símbolos para suma (+); resta (-), multiplicación (*); división (/); conjunción (&&); disyunción (||); menor (<) mayor (>), menor o igual (<=); mayor o igual (>=); igual (=) y distinto(!= o ~=). Se deja a cargo del docente de computación la planificación y gestión de estas clases.

Las clases 4 y 5 se proponen con el objetivo de diseñar un programa que permita calcular el DCM entre dos números y utilizarlo para dar respuesta a nuevos problemas. Estas clases estarán a cargo de los profesores de Computación y Matemática. Se planifican para ser desarrolladas aproximadamente durante 320 minutos.

Al comenzar la clase 4 se plantea la tarea, para ser resuelta en grupo de 3 o 4 estudiantes, que solicita crear un programa que les permita calcular el divisor común mayor, para esto deben poner a funcionar e integrar los conocimientos adquiridos o recuperados en las clases anteriores.

Se presentan a continuación las principales tareas planificadas:

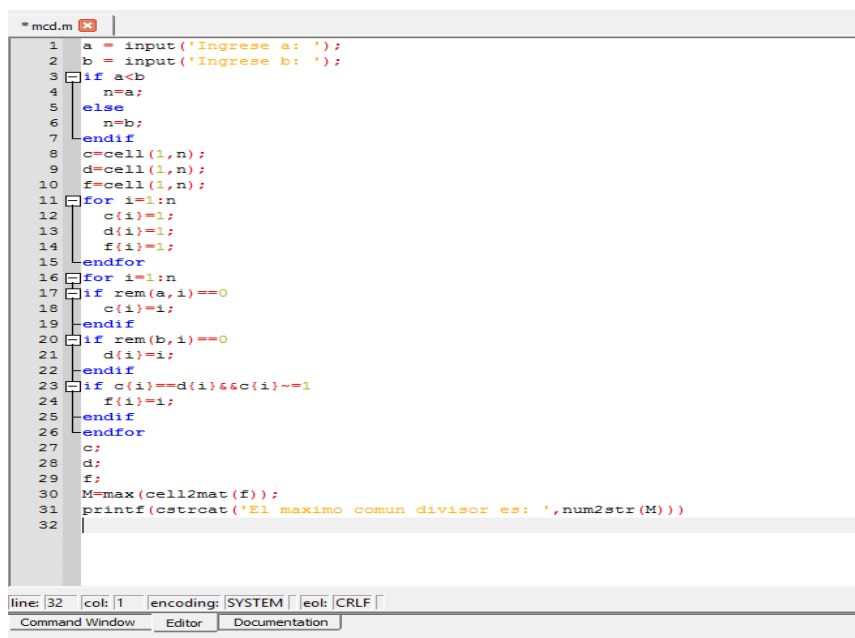
Tarea 2: Con lo trabajado en la clase de matemática y en la clase de informática creen un programa en Octave, que les permita encontrar el divisor común mayor entre dos números, y luego dar solución al siguiente problema, ingresando por pantalla los valores que correspondan. Antes de escribir el programa, realicen el diagrama de flujo correspondiente. Camila tiene una cuerda de 1512 centímetros y otra de 840 centímetros. Desea cortarlas de modo que todos los tramos sean iguales pero lo más largos posible. ¿Cuál debe ser la medida de cada tramo?

Observación: Para comprobar que el programa es correcto deben probar con distintos números y siempre debe darles el divisor común mayor entre esos números.

Pueden pensar en la siguiente pregunta: Si se cambian las medidas de las cuerdas, ¿el programa que crearon les sirve para dar respuesta a la pregunta, o deben modificar algo?

Sugerencia: Tener en cuenta las definiciones que recuperadas en clase 1 de divisor común mayor.

En la figura 1 se muestra uno de los programas que pueden escribir los estudiantes, basándose en una de las definiciones de divisor común mayor².



```
mcd.m
1 a = input('Ingrese a: ');
2 b = input('Ingrese b: ');
3 if a<b
4     n=a;
5 else
6     n=b;
7 endif
8 c=cell(1,n);
9 d=cell(1,n);
10 f=cell(1,n);
11 for i=1:n
12     c(i)=1;
13     d(i)=1;
14     f(i)=1;
15 endfor
16 for i=1:n
17     if rem(a,i)==0
18         c(i)=1;
19     endif
20     if rem(b,i)==0
21         d(i)=1;
22     endif
23     if c(i)==d(i)&&c(i)==1
24         f(i)=1;
25     endif
26 endfor
27 c;
28 d;
29 f;
30 M=max(cell2mat(f));
31 printf('El maximo comun divisor es: ',num2str(M))
32
```

Figura 1: Programa para calcular el divisor común mayor.

Existen tantas combinaciones para realizar un programa que calcule el D.C.M como grupos de estudiantes tengamos, ya que probablemente nunca dos programas sean exactamente iguales. En general, se pueden identificar 3 grandes grupos de programas que utilizan la definición de divisor común mayor (sin pasar por la factorización).

- Aquellos programas como el arriba mencionado, que, en primer lugar, buscan los divisores de a y b (por separado), luego, a partir de estas listas, encuentran los comunes a ambos, y de estos comunes obtienen el mayor.
- Aquellos que recorren los primeros números naturales hasta hallar todos los que dividen tanto a “a” como a “b”, y quedarse con el mayor (esto es, divido “a” y divido “b” por cada número natural menor a ambos). Si el resto para ambas divisiones es 0, tengo un divisor común.
- Aquellos que, al igual que el primer tipo, buscan los divisores de a y b (por separado) y luego encuentran los comunes a ambos, pero, en lugar de quedarse con el mayor de

² Siempre puede haber pequeñas diferencias entre un programa y otro, lo importante es que a la base de estos programas está la definición de divisor común mayor.

ellos, verifican, uno a uno, cual de todos es divisible por todos los demás.

Cada uno de estos programas responde a una forma distinta de definir DCM, que creemos son igual de enriquecedoras a la hora de resignificar este objeto.

Habiendo terminado el programa y controlando que funcione para dar respuesta a la tarea y para otros números, se entrega las siguientes tareas, con el objetivo de que los alumnos puedan poner a funcionar el programa para dar respuesta rápidamente a los problemas, notando así la potencia de este.

Tarea 3: Utilizando el programa que crearon anteriormente resolver los siguientes problemas.

Máximo quiere pintar una casa pequeña. Según sus cálculos, necesitará 12 litros de pintura naranja y 45 litros de pintura blanca. Pero quiere comprar botes de pintura que tengan la misma cantidad de litros y que esa cantidad de litros sea la mayor posible, ¿de cuántos litros debe ser cada bote?

Una tienda compra memorias USB de diferentes colores al por mayor. Para Navidad hizo un pedido extraordinario de 196 memorias azules y 252 verdes. Para guardar la mercancía de forma organizada, exigió que le enviaran las memorias en cajas iguales, sin mezclar los colores y conteniendo el mayor número posible de memorias.

Si se cumplen las exigencias de la tienda, ¿cuántas memorias habrá en cada caja?

La clase 6 se propone con el objetivo de compartir las producciones de los estudiantes, ponerlas en relación y debatir sobre la re-significación del divisor común mayor. Se planifica para ser desarrollada durante 80 minutos aproximadamente y a cargo de los profesores de informática y computación. Esta clase consistirá de una puesta en común entre los diferentes grupos, en la cual se expondrá los programas que cada grupo realizó. Se pondrán a discusión las diferencias y similitudes entre los distintos programas y las potencias o debilidades de uno sobre el otro. Luego, cada grupo deberá realizar un esquema sobre el procedimiento matemático que su programa utiliza para hallar el DCM. Se discutirá cómo queda explícita la definición en los programas realizados, y se preguntará a los estudiantes qué creen que aportó este proyecto a cada una de las disciplinas intervinientes.

Se resignifica el divisor común mayor operativizando su definición, y se hace de manera económica, lo que es una fortaleza de la programación sobre el trabajo con lápiz y pa-

pel. La descomposición en factores primos ya no aparece como técnica obligada y única para la resolución de este tipo de problemas.

Algunas Reflexiones

En este proyecto se propone una integración entre diversos tipos de conocimientos, métodos y formas de acción pertenecientes a diferentes disciplinas, los cuales potencian a las mismas y a su vez, pone en marcha una particular modalidad de enseñanza de la matemática. El conocimiento matemático surge para dar respuesta a un problema determinado, que no es posible abordar desde una única asignatura, sino que es necesario el aporte de cada una de las asignaturas involucradas. De esta manera, los conocimientos aparecen de manera que posibilitan la integración y relación entre diferentes disciplinas.

Durante el último mes de la cursada del taller interdisciplinar en el que tuvo lugar la elaboración del proyecto, fue posible evidenciar algunas de las características de un aprendizaje basado en problemas descriptas por Valero y Ravn (2017). Así por ejemplo, la participación activa de las estudiantes del profesorado al involucrarse en la formulación de un problema que pudiera ser abordado de manera interdisciplinaria; en definir una estrategia para la escritura, etc. Otra de las características que mencionan Valero y Ravn (2017) y que se pudo vivenciar fue la generación de un espacio de aprendizaje en el que surgieron discusiones, colexiones, toma de decisiones colectivas; todo en un ambiente de respeto, confianza y valoración del pensamiento del “otro”.

Se espera que este proyecto pueda ser disparador de otros trabajos interdisciplinarios, y vaya introduciendo, tanto en docentes como estudiantes, una nueva estrategia pedagógica para el trabajo áulico, acorde a lo que las propuestas curriculares más recientes plantean para el futuro educativo cercano.

Referencias bibliográficas

Ander-Egg, E. (1994). *Interdisciplinarietà en educación*. Buenos Aires: Magisterio.

Almidón López Huancayo, I.R. (2017). *El papel de la interdisciplinarietà en la enseñanza aprendizaje de la matemática*. Iberoamérica Divulga: Comunidad de Educadores para la Cultura Científica. Recuperado de <https://www.oei.es/historico/divulgacioncientifica/?El-papel-de-la-interdisciplinarietà-en-la-ensenanza-aprendizaje-de-la>

- Carvajal Escobar, Y. (2010). Interdisciplinariedad: desafío para la educación superior y la investigación. *Revista Luna Azul*, 31, 156-169. Recuperado de <http://www.scielo.org.co/pdf/luaz/n31/n31a11.pdf>
- Valero, P. y Ravn, O. (2017). Recontextualizaciones y ensamblajes: ABP y matemáticas universitarias. *Didacticae*, 1, 4-25.
- Volkheimer, W. (s/f). *Enfoque Interdisciplinario*. Mendoza: Conicet. Recuperado de <https://www.mendoza-conicet.gob.ar/portal/enciclopedia/terminos/EnfoInt.htm>