

# CONTROLADOR LOGICO PROGRAMABLE PARA SOLUCIONES DE AUTOMATIZACIÓN Y CONTROL EN LA PEQUEÑA Y MEDIANA EMPRESA

## PROGRAMMABLE LOGIC CONTROLLER FOR AUTOMATION AND CONTROL SOLUTIONS IN THE SMALL AND MEDIUM COMPANY

*Nicolás Molina Vuistaz<sup>(1)</sup>, Nahuel Nieva<sup>(2)</sup>, Alejandro Martinez<sup>(3)</sup>*

<sup>(1)</sup> Universidad Nacional de La Matanza  
nicomolina@unlam.edu.ar

<sup>(2)</sup> Universidad Nacional de La Matanza  
nahuel.nieva.1991@gmail.com

<sup>(3)</sup> Universidad Nacional de La Matanza  
a.j.martinez1412@gmail.com

**Resumen:**

Se describe el diseño de un Controlador Lógico Programable, de características funcionales destacadas, para soluciones de automatización y control en la pequeña y mediana empresa. Se exponen las características particulares del controlador, como lo son la modularidad, versatilidad y escalabilidad, combinados con una alta capacidad de cálculo y bajo consumo de energía. Además, se explican las distintas estrategias de evaluación del sistema, basadas en el análisis funcional de diversas prestaciones necesarias para la función planteada, tales como la determinación del tiempo de cómputo de filtros digitales y del consumo de corriente del núcleo para distintas velocidades de reloj.

**Abstract:**

The design of a Programmable Logic Controller is described, with outstanding functional features, for automation and control solutions, in small and medium companies. The special features of the controller are exposed, such as modularity, versatility and scalability, combined with a high calculation capacity and low power consumption. In addition, the different evaluation strategies of the system are explained, based on the functional analysis of various features necessary for the proposed function, such as the determination of the digital filter count time and the current consumption of the core for different clock speeds.

**Palabras Clave:** *controlador lógico programable, sistemas autónomos, filtros digitales*

**Key Words:** *programmable logic controller, stand-alone systems, digital filters*

**Colaboradores:** *Fernando I. Szklanny, Docente Investigador Universidad Nacional de La Matanza*

## **I. CONTEXTO**

El Controlador Lógico Programable surge como una consecuencia de las necesidades generadas en otros proyectos que lleva a cabo el grupo de investigación de lógica programable y procesamiento digital de señales perteneciente al Departamento de Ingeniería e Investigaciones Tecnológicas de la Universidad Nacional de La Matanza. Como resultado de estos otros proyectos, aparece la necesidad de normalizar los requerimientos de adquisición y control de distintas variables que intervienen en la mayoría de las aplicaciones de robótica y control industrial.

Por consiguiente, y de acuerdo con esas necesidades, se propuso el diseño de un Controlador Lógico Programable, de características funcionales destacadas, para soluciones de automatización y control en la pequeña y mediana empresa.

## **II. INTRODUCCIÓN**

En la actualidad, debido a los grandes avances tecnológicos en el campo de los microprocesadores [1], los diseñadores poseen un amplio abanico de excelentes alternativas de bajo costo para el desarrollo de soluciones electrónicas. Todas estas opciones permiten agregar a los sistemas en cuestión capacidades tales como alto poder de cálculo, manejo de gran caudal de datos, diversas interfaces integradas, bajo consumo energético, entre muchas otras. Estas características vuelven a los microcontroladores de hoy en día, herramientas idóneas para el control digital de sistemas orientados a robótica y sistemas autónomos en general. Dentro de este marco, se expone el desarrollo de un controlador programable económico, que reúne las características antes mencionadas en un diseño simple, pero a la vez flexible y escalable.

Puesto que los algoritmos destinados al procesamiento digital de las señales insumen el mayor tiempo de cómputo del procesador, es necesario evaluar la capacidad de cálculo del procesador, con la finalidad de certificar que el mismo será capaz de desempeñar su función de manera satisfactoria.

La herramienta seleccionada para tal fin es la medición del tiempo de cómputo de los algoritmos de filtros digitales [2], implementado con números de punto flotante de simple precisión. Este método resulta el más destacado debido a la importancia que sustenta la utilización de este tipo de filtros en sistemas de control como el descrito, usado para el procesamiento de diversas señales, como lo son las provenientes de sensores de posición, acelerómetros y giróscopos, a fin de eliminar el ruido en las señales o componentes de frecuencias no deseadas, etc. De forma paralela, se evaluó el consumo de energía del procesador para distintas frecuencias de operación del mismo.

## **III. DESCRIPCIÓN DEL SISTEMA**

El sistema controlador se diseñó en función de los requerimientos ya mencionados, además de disponer de varias interfaces de comunicaciones para la interacción del módulo con otros dispositivos, se incluyeron varios pines de entrada/salida para comandar diversos tipos de motores y recibir entradas de sensores de tipo codificador, de efecto hall, acelerómetros y giróscopos, entre otros. Sin embargo, se debe destacar que el mismo constituye un sistema de desarrollo flexible para diversas aplicaciones. Esto se logra mediante una interfaz de conexión que permite incorporar diversos módulos conectados en cascada. Es decir, una bornera de expansión permite ampliar las funciones del controlador de acuerdo con las necesidades de cada caso y provee la

escalabilidad y versatilidad requeridas para su utilización en aplicaciones de control de cualquier índole. A continuación, se describe brevemente los principales bloques del sistema (visibles en la figura 1):

#### A. Diagrama en bloques del Controlador Programable

El microcontrolador que se utiliza es el STM32F407VGT [3], de la firma STMicroelectronics, basado en un procesador de alto rendimiento ARM®Cortex®-M4 [4][5], de 32 bits y arquitectura RISC. El dispositivo puede operar hasta una frecuencia de 168 MHz, posee una unidad de punto flotante de simple precisión, un juego completo de instrucciones para el procesamiento digital de señales, dispone de un 1 Mbit de memoria del tipo Flash y 192 Kbit de memoria RAM, además de ofrecer una gran cantidad de periféricos. También implementa un set completo de instrucciones para procesamiento digital de señales (DSP) e incluye una unidad de protección de memoria (MPU), que mejora la seguridad del sistema para ciertas aplicaciones.

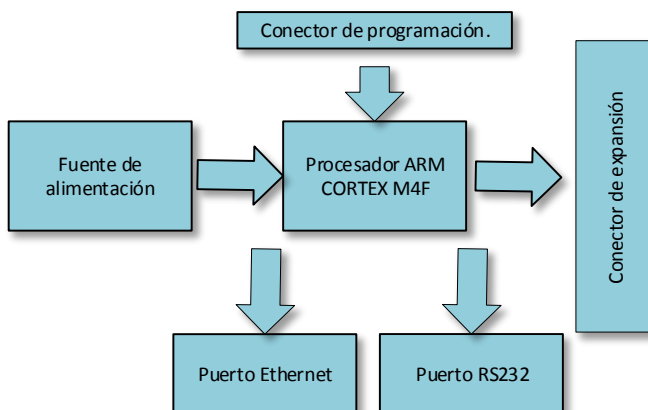


Fig. 1 Diagrama de bloques del controlador programable.

Algunas de las características típicas de DSP que se incluyen en este núcleo son [6]:

- Arquitectura Harvard.
- MAC de 16 y 32-bits de un solo ciclo de operación.
- Barrel Shifter.
- SIMD (aritmética de ciclo único).

#### B. Conector de programación

Para la programación del microcontrolador se utiliza una interfaz SWD (Serial Wire Debug) que permite la programación, revisión del código y visualización de variables en tiempo de ejecución.

#### C. Fuente de alimentación.

La fuente de alimentación del controlador está basada principalmente en dos reguladores integrados. El primero de ellos es un LM2596, que constituye un regulador de fuente conmutada de tipo step-down, que posee una salida de +5V y tolera tensiones de entrada de hasta +40V. Además, puede entregar una corriente de salida de hasta 3A (con una excelente regulación de salida), motivo por el cual es ideal para la extensión de la alimentación, por ejemplo, a un módulo de servomotores DC que se conecte en cascada. Luego, para la alimentación del microcontrolador se utiliza un regulador LM1117 que reduce la tensión desde +5V a +3.3V. Cabe destacar que, como medio de protección adicional de la fuente, se incorporó un fusible reinicializable PTC, que permite proveer protección circuital repetidas veces, aumentando la robustez del sistema frente a perturbaciones en la alimentación.

#### D. Bloque de expansión

Una de las principales características ya mencionadas de la placa controladora objeto del presente es su

escalabilidad, la que se logra con un conector de expansión que permite vincular diferentes módulos de propósito general, de entradas y salidas digitales y analógicas, módulos de comunicación industrial y módulos especiales destinados a realizar tareas específicas.

Concretamente, la bornera dispone de conexiones de +3.3V, +5V, GND y 20 líneas de propósito general configurables como entradas/salidas digitales y analógicas. Además, se pueden programar como salida y entradas de los módulos de comunicación (CAN, UART, SPI, I2C, USB) y los temporizadores (TIMs) del microcontrolador. Por otro lado, se incluyeron interfaces fijas de comunicación serial en borneras independientes, y las referencias de tensión también se protegieron con fusibles reiniciables.

#### E. Bloque del módulo RS-232

La interfaz de comunicación serial asíncrona se desarrolló en base a un dispositivo integrado MAX3232 que posee interfaces duales de transmisión y recepción, con tensiones adaptadas a niveles RS-232 en el conector DB-9. El circuito integrado opera en un rango de +3V a +5.5V con corrientes típicas de 300 uA y no necesita más que algunos capacitores externos, lo que reduce considerablemente el tamaño físico del circuito implementado, por lo que permite un mayor nivel de integración.

El módulo de comunicación serie, además de permitir la comunicación con otro dispositivo, también sirve para poder descargar actualizaciones del software del controlador [7], lo que permite realizar actualizaciones en forma sencillas sin tener que desmontar el equipo.

#### F. Módulo Ethernet

La placa incluye un conector RJ-45 para utilizar una conexión de tipo Ethernet para la vinculación del microcontrolador con otros dispositivos que soporten esta comunicación (por ejemplo, computadoras, laptops, etc.). Para tal fin, se utilizó un circuito integrado DP83848, el que constituye una interfaz de capa física de Ethernet de 10/100Mb diseñado para ambientes industriales, con elevados niveles de ruido e interferencias. Este dispositivo cumple con el estándar IEEE 802.3u en diversos rangos de temperatura (industrial, militar), lo que lo hace ideal para su operación en un sistema de control electromecánico. Debido a que soporta tanto 10BASE-T como 100BASE-TX se asegura la compatibilidad e interoperabilidad con todos los estándares Ethernet existentes.

#### G. Consideraciones en el diseño del circuito impreso

Para el diseño del circuito impreso se tomaron consideraciones de las guías de diseño de compatibilidad electromagnética propuestas por el fabricante del microcontrolador [8] y se incorporaron protecciones contra descargas electrostáticas. En la figura 2 puede observarse el módulo de control, la cual posee todas las características antes mencionadas.

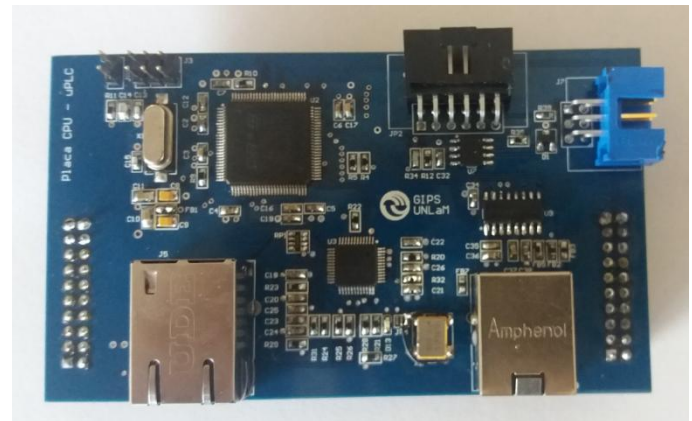


Fig. 2 Módulo de control



En la figura 3 puede observarse el módulo de entradas y salidas y en la figura 4 el montaje final del Controlador Lógico Programable.

#### IV. DESARROLLO DEL SOFTWARE

Dado que los procesadores ARM están muy difundidos, existen una innumerable cantidad de herramientas de desarrollo pagas o gratuitas y una gran cantidad de bibliotecas para el desarrollo de aplicaciones de todo tipo.

La biblioteca más importante es la CMSIS (Cortex Microcontroller Software Interface Standard) [9], que permite establecer una abstracción del hardware común a todos los fabricantes de procesadores ARM Cortex-M y que posee una cantidad importante de funciones altamente optimizadas.

En particular, existe un subconjunto de la biblioteca ya mencionada, denominada CMSIS-DSP, que cuenta con más de 60 funciones para el procesamiento digital de señales, de las cuales se pueden mencionar funciones básicas de matemática, operaciones con vectores y matrices, interpolación, manejo de números complejos, implementación de filtros digitales IIR, FIR, LMS, transformadas y el control del tipo PID.



Fig. 3 Módulo de entradas y salidas

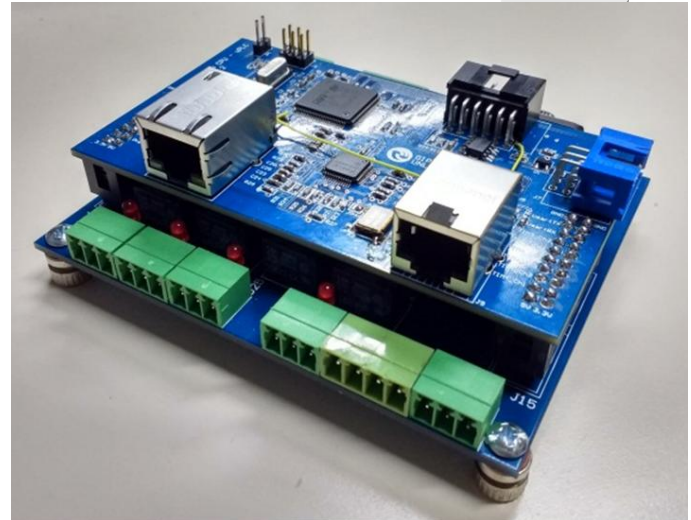


Fig. 4 Controlador Industrial Programable

Para la programación de tareas y funciones que debe realizar el controlador, se puede utilizar cualquier plataforma de desarrollo para procesadores ARM.

En particular cabe mencionar la herramienta uVision 5 de KEIL [10] y la herramienta CoIDE de Coocox [11], en combinación con el compilador GCC ARM Embedded [12].

Para el desarrollo de las funciones básicas destinadas a configurar los periféricos del controlador se dispone de la herramienta STM32CubeMX de la firma STMicroelectronic [13], que permite configurar los periféricos en forma gráfica.

Con la finalidad de disminuir los tiempos de diseño de aplicaciones con el controlador, se desarrolló una biblioteca para la configuración básica de los periféricos, la que, en conjunto con las bibliotecas de funciones de procesamiento digital de señales, permite el diseño de aplicaciones en forma rápida.

#### V. PRUEBAS DE RENDIMIENTO

Para evaluar la capacidad del cálculo del procesador, se evaluaron la implementación un filtro elimina banda.

Además, se analizaron los distintos niveles de

optimización del compilador propios de uVision 5 de KEIL.

Para el diseño del filtro elimina banda se utiliza un filtro digital de respuesta infinita IIR de sexto orden del tipo Butterworth, con topología en cascada, para eliminar la frecuencia de 50 Hz, con un ancho de banda de 4 Hz, a una frecuencia de muestreo de 1,4 KHz. Las operaciones se realizaron con números de punto flotante de precisión simple. En la figura 5 se visualiza la estructura básica del filtro.

Para el cálculo de los coeficientes de los filtros y la simulación se utilizó la herramienta NuHz [14]. En la figura 6 se observa la simulación de la respuesta en frecuencia del filtro elimina banda que proporciona la misma. En la figura se representan amplitud en trazo más oscuro y la fase en trazo más claro. En la misma, en el eje y se indica la amplitud en dB, en paso de 10db y en el eje x se indica frecuencia en escala logarítmica.

Para la verificación de la respuesta en el cálculo del filtro se utilizó la placa de desarrollo STM32F4 DISCOVERY [15], que posee un microcontrolador STM32F407VGT. En la tabla 1 se muestra el tiempo de cómputo que se insinse en el cálculo del filtro elimina banda.

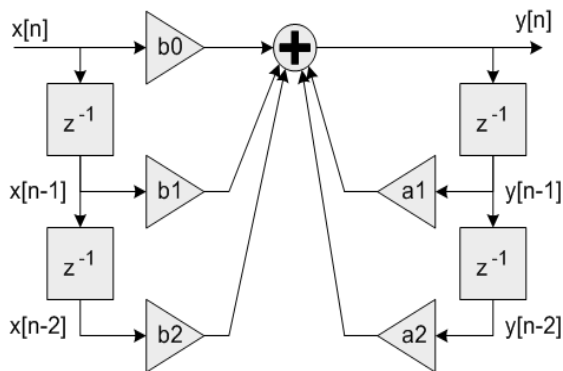


Fig. 5 Esquema básico de un filtro digital IIR biquadrático.

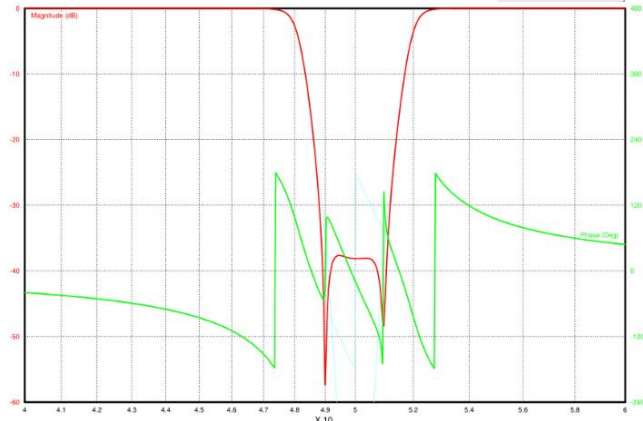


Fig. 6 Simulación de la respuesta en frecuencia del filtro elimina banda.

El filtro elimina banda presentó un muy buen rendimiento en el tiempo de cómputo, dado que solo se ocupa un 0.3% del tiempo entre dos interrupciones seguidas del conversor digital/análogo (se considera el mejor nivel de optimización obtenido).

Con respecto a la respuesta del ancho de banda se obtuvo un valor de aproximadamente 4 Hz, coherente con el valor esperado. En la figura 7 se observa la salida del filtro elimina banda, medida para una frecuencia de 50 Hz. A continuación, se detalla el algoritmo utilizado para el cómputo del filtro elimina banda:

```
float Filtro_orden6(float invar)
{
    float sumnum=0.0, sumden=0.0; int i=0, j=0;
    sumnum = .9659*invar;
    for (i=0;i<=5;i++){
        sumden=0.0;
        for (j=0;j<2;j++){
            sumden += states[i][j]*zden[i][j];
            sumnum += states[i][j]*znum[i][j];
            if (j<1) states[i][j] = states[i][j+1];
        }
        states[i][1] = (invar-sumden);
    }
    return sumnum;
}
```

Tabla 1: Tiempos de cómputo del filtro elimina banda.

Nivel de optimización	ciclos de maquina	tiempo de ejecución [microsegundos]
0	930	5,54
1	430	2,56
2	415	2,47
3	412	2,45

#### IV. RESULTADOS Y OBJETIVOS

El controlador lógico programable está destinado principalmente al procesamiento de señales provenientes de los diferentes sensores del autómatas, para lo cual se evaluaron las prestaciones de hardware requeridas para dichas tareas. Los resultados obtenidos se consideran de nivel excelente, sobre la base de las conclusiones que se resumen a continuación.

En base a las mediciones de cómputo, el controlador presentó un excelente rendimiento para una frecuencia de muestreo de alrededor de un 1KHz, dado que se ocupa menos del 1% del tiempo que se dispone para procesar la señal obtenida por el conversor digital analógico.

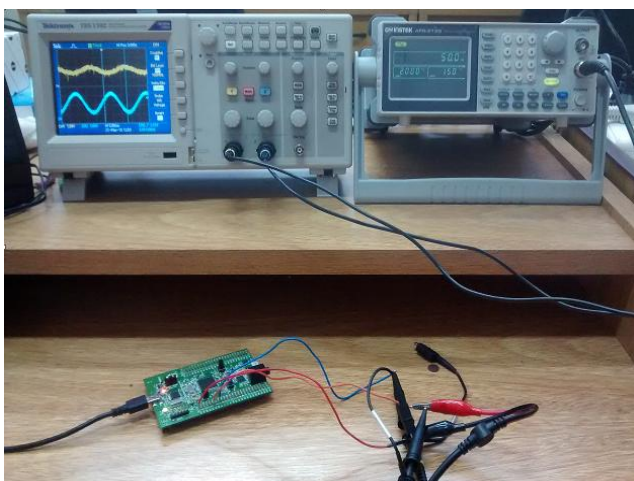


Fig. 7 Pruebas del algoritmo del filtro digital elimina banda.

Para los casos que se requiere una frecuencia de muestreo cercana a los 100 KHz, también presentó un buen rendimiento, pero se limita la cantidad de señales que se pueden procesar al mismo tiempo, dado que se dispone de un menor tiempo de cómputo. Para estos casos se puede combinar la utilización de filtros analógicos con los digitales. Esta combinación se puede lograr sin modificar el diseño del controlador programable gracias a que dispone de un puerto de expansión.

En cuanto al consumo se observa, como es de esperarse, que baja en forma considerable a medida que disminuye la frecuencia de reloj y a pesar de que el tiempo de cómputo del filtro aumenta en la misma proporción, para una frecuencia de operación de 2 MHz, aún se pueden computar filtros digitales con la limitación de la cantidad de señales que se pueden procesar al mismo tiempo.

En términos generales el controlador presenta una gran capacidad de cálculo, sobre todo si se configura el compilador para que utilice el nivel dos de optimización. Permite ajustar la velocidad de reloj para lograr bajos consumos y se puede complementar con diferentes módulos según se requiera gracias el puerto de expansión que dispone.

#### VII. REFERENCIAS

- [1] Santanu Chattopadhyay (1 January 2010). Embedded System Design. PHI Learning Pvt. Ltd. p. 9. ISBN 978-81-203-4024-4. Retrieved 15 March 2013.
- [2] A. Antoniou, Digital Filters: Analysis, Design, and Applications, New York, NY: McGraw-Hill, 1993.
- [3] The STM32F405xx and STM32F407xx family is based on the high-performance ARM®Cortex®-M4 32-bit RISC core operating at a frequency of up to 168 MHz. <http://www.st.com/web/catalog/mmc/FM141/SC1169/SS1577/LN11/PF252140>



- [4] ARM Inc. “Cortex-M4 Devices Generic User Guide”, 2010.
- [5] The Definitive Guide to the ARM Cortex-M3 and Cortex-M4 Processors; 3rd Edition; Joseph Yiu; Newnes; 600 pages; 2013; ISBN 978-0124080829.
- [6] Shyam Sadasivan, Developing optimized signal processing software on the Cortex-M4 processor, White Paper, ARM, Noviembre 2010.
- [7] Application note STM32F40x/STM32F41x in-application programming using the USART, 2011 STMicroelectronics.
- [8] Application Note - EMC design guide for ST microcontrollers, Cyril Troise, Microcontroller Division Applications, 2003 STMicroelectronics.
- [9] The ARM® Cortex® Microcontroller Software Interface Standard (CMSIS).  
<http://www.arm.com/products/processors/cortex-m/cortex-microcontroller-software-interface-standard.php>
- [10] The Keil IDE <http://www.keil.com/uvision/>
- [11] The Cocox IDE <http://coocox.org/>
- [12] J. Ye. (2011, Dec. 22) GNU tools for ARM embedded processors. [Online]. Available: <https://launchpad.net/gcc-arm-embedded>.
- [13] STM32CubeMX initialization code generator.  
[http://www.st.com/web/catalog/tools/FM147/CL1794/SC961/SS1743/PF259242?icmp=stm32cubemx\\_pron\\_pr-stm32cubef2\\_apr2014&sc=stm32cube-pr2](http://www.st.com/web/catalog/tools/FM147/CL1794/SC961/SS1743/PF259242?icmp=stm32cubemx_pron_pr-stm32cubef2_apr2014&sc=stm32cube-pr2).
- [14] The NuHz tool <http://www.nuHz.com/>
- [15] STMicroelectronics. (2010, Sep. 14) STM32 discovery kit. [Online]. Available: [http://www.st.com/internet/com/press\\_release/p3065.jsp](http://www.st.com/internet/com/press_release/p3065.jsp)

**Recibido:** 2017-12-17

**Aprobado:** 2017-12-26

**Datos de edición:** Vol. 2-Nro. 2-Art. 4

**Fecha de edición:** 2017-12-28

**URL:** <http://reddi.unlam.edu.ar/>